**WEB-BASED VOLUNTEER MANAGEMENT SYSTEM**

BY

CHANG WEI HANG

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology

(Kampar Campus)

JUNE 2025

# COPYRIGHT STATEMENT

# ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my supervisor, Ts. Tan Teik Boon, for his invaluable guidance, patience, and unwavering support throughout this research journey. His expertise, constructive feedback, and encouragement were instrumental in shaping this project and overcoming challenges. I am truly grateful for his dedication and mentorship, which have profoundly enriched my academic and professional growth.

My heartfelt thanks also go to my beloved family for their endless love, understanding, and encouragement during this journey. Their belief in me has been my anchor through every obstacle, and I owe my perseverance to their constant support.

Lastly, I extend my appreciation to my friends who offered advice and motivation, making this journey more meaningful.

# ABSTRACT

This project focuses on developing a robust web-based Volunteer Management System (VMS) to enhance organisational efficiency in volunteer coordination. The system addresses common pain points in traditional volunteer management, including inefficient skill matching, poor geographical allocation, and low engagement retention. Through comprehensive analysis of existing VMS platforms, critical limitations were identified in dynamic skill assessment and location-aware scheduling. The proposed solution utilises React.js for frontend development, supported by a Node.js backend and MySQL database, creating a responsive and scalable platform. Key innovations include a weighted skill-matching algorithm that evaluates both skills importance level required and volunteer proficiencies, a distance-based matching algorithm using Google Maps Distance Matrix API, and leaderboard gamification elements for volunteer contribution recognition.

Since the universe of possible volunteer skills is too broad, focusing on a single domain helps reduce the scope of skills to a manageable and realistic level. For demonstration purposes, this project applies the system to the animal welfare domain. However, the skill framework is designed to be domain-tailorable, allowing the same system to be adapted for other sectors such as education, healthcare, or disaster relief.

The system offers comprehensive functionalities such as event creation, event modification, event application, and volunteer leaderboard. By integrating these features, the VMS provides organisations with a unified platform for volunteer onboarding, event coordination, and long-term engagement, while offering volunteers a more personalised and accessible experience.

Area of Study (Minimum 1 and Maximum 2): Web Front End Development, Web Back End Development

Keywords (Minimum 5 and Maximum 10): Volunteer Management System, Event-Volunteer Matching Algorithm, Weighted Skill Matching Algorithm, Distance-based Matching Algorithm, Leaderboard Gamification

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF SYMBOLS

# LIST OF ABBREVIATIONS

| | |
|---|---|
| *VMS* | Volunteer Management System |
| *RSVP* | Respond, if you please |
| *ReDi* | Readiness Disaster intelligence |
| *UI* | User Interface |
| *HTML* | HyperText Markup Language |
| *DOM* | Document Object Model |
| *XML* | Extensible Markup Language |
| *IDE* | Integrated Development Environment |
| *HTTP* | Hypertext Transfer Protocol |
| *npm* | Node Packet Manager |

# Chapter 1
# Introduction

## 1.1 Project Background

Volunteers are the backbone of nonprofit organisations, providing essential support to help them fulfil their missions. Therefore, a proper way to manage volunteers effectively is crucial to maintain the long-term viability and success of programs and events that depend significantly on volunteer involvement [1]. However, many nonprofits face challenges in managing volunteers due to limited resources and outdated methods. Traditional approaches such as using spreadsheets or even pen and paper, are often inefficient and can hinder an organisation's ability to recruit, onboard, and manage volunteers effectively. To address these challenges, modern volunteer management systems have evolved to simplify these processes, improve volunteer experiences, and enhance overall operational efficiency [2].

Volunteer Management Systems (VMS) are essential tools that streamline the volunteer management lifecycle as shown in figure 1.1.1 below, start from planning, recruitment and screening, scheduling, training, engagement, to evaluation. Planning usually includes setting goals and creating role descriptions, followed by volunteer recruitment through various channels. Effective scheduling matches volunteers with the right tasks, while thorough training and onboarding prepare them to perform their roles perfectly. To enhance retention, volunteers must feel appreciated and supported, which can be achieved through ongoing engagement and personal growth opportunities. Finally, evaluating the program and gathering feedback help refine processes [1]. To achieve these, VMS provide a centralised platform to handle crucial functions like volunteer recruitment, scheduling, communication, and reporting. With common features such as volunteer registration, task matching, and integrated communication tools, VMS enhances the overall efficiency of managing volunteers. Additionally, advanced systems may offer background checks and integrate with donor management software, providing a holistic view of an organisation’s supporters [1, 3]. Ultimately, a well-implemented VMS improves the experience for both volunteers and coordinators, helping organisations better manage their valuable volunteer resources.

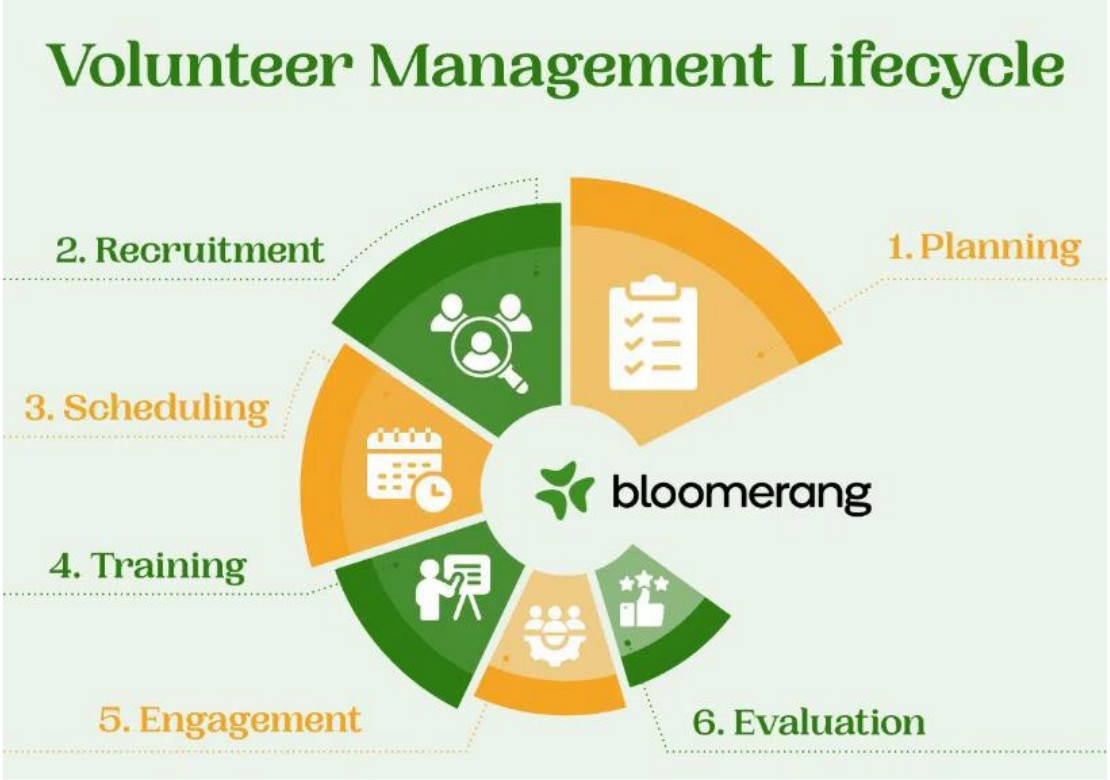



Figure 1-1: Volunteer management lifecycle [1].

Each Volunteer Management System (VMS) offers different benefits to different organisations, particularly for those who transitioning from manual, paper-based processes [2]. Some of the common advantages provided by VMS include efficiency, enhanced communication, improved engagement, data management, and reporting and analytics. By automating administrative tasks such as data entry, it reduces manual efforts and saves time for both organisers and volunteers, thus improving efficiency. Besides, VMS helps to promote clear and direct communication, ensuring that volunteers stay informed about opportunities and updates. Furthermore, volunteer engagement can be improved by allowing volunteers to easily view and sign up for their interested available roles. With centralised data management, tracking of volunteer hours, preferences, and contributions can be easily done. Additionally, advanced reporting and analytics feature enable organisations to analyse data, identify trends, and make informed decisions to continually optimise their volunteer programs [3].

## 1.2 Problem Statement and Motivation

Many existing volunteer management systems often face significant limitations in their skill-matching capabilities. These platforms primarily rely on simple keyword matching systems that fail to assess actual skill levels or verify competencies. The absence of robust proficiency assessment mechanisms means that these systems cannot distinguish between varying levels of skill mastery or validate claimed capabilities. As a result, this will lead to mismatches in volunteer assignments where individuals may be placed in roles that either underutilize their qualifications or exceed their actual capabilities. Such mismatches will create operational inefficiencies for organisations, as they require additional time and resources to manage the volunteer recruitment process. At the same time, it also frustrates volunteers with inappropriate opportunity suggestions.

Besides, geographical consideration is another major weakness in current systems. Most platforms either ignore location entirely or use imprecise regional categories rather than actual distance calculations. Volunteers often encounter distant opportunities before nearby ones, even when both match their interests. This creates unnecessary participation barriers, as most volunteers prefer conveniently located roles. Consequently, organisations always struggle with low attendance and last-minute volunteer shortages. The failure to implement intelligent proximity-based matching represents a significant missed optimisation opportunity, as prioritising nearby volunteers would naturally lead to more reliable attendance and greater participation consistency.

Lastly, maintaining volunteer engagement and motivation is a critical issue faced by many volunteer management systems. Inadequate contribution recognition can lead to volunteers feeling disconnected from the organisation. When volunteers feel disconnected or unappreciated, their enthusiasm and commitment will be decreased, resulting in high turnover rates. This low engagement environment makes it difficult for organisations to establish a stable and reliable volunteer base. Hence, there are persistent challenges for organisations who try to maintain an active and dedicated group of volunteers.

Therefore, I am motivated to develop a web-based Volunteer Management System (VMS) to overcome the critical challenges faced by organisations in managing their volunteer programs. Effective volunteer management is essential for the success of charities, schools, and

community organisations, yet traditional methods often fall short. Many existing systems struggle to manage and catalogue the diverse skills of volunteers efficiently, leading to underutilisation of their talents and difficulties in assigning tasks appropriately. This inefficiency can impact task performance and diminish the overall effectiveness of volunteer programmes. Additionally, current systems fail to properly account for geographical proximity when matching volunteers, often suggesting distant opportunities before nearby ones. This oversight discourages participation and leads to lower turnout, even when qualified volunteers live close by. Furthermore, inadequate recognition can cause volunteers to feel disconnected, which decreases their enthusiasm and commitment. This leads to high turnover rates and makes it difficult for organisations to build a stable and reliable volunteer base.

By developing a web-based VMS in this project, I aim to address these challenges directly and enhance the volunteer-event matching algorithms. A well-designed system can streamline the management of volunteer skills, improve event coordination, and enhance volunteer engagement. This will not only make volunteer programmes more effective but also help organisations better utilise their resources and foster a more dedicated volunteer community. My goal is to create a solution that simplifies and enhances volunteer management, ultimately contributing to the greater success and sustainability of the organisations that depend on their volunteers.

## 1.3 Research Objectives

The first objective is to develop a comprehensive and organised system for managing and indexing volunteer skills through an advanced weighted matching algorithm. Unlike traditional systems that rely on basic keyword matching, this system employs predetermined skill categories that can be tailored to the specific domain in which the volunteer management system (VMS) is deployed. Within these categories, volunteers will self-rate their skill proficiency, while organisers will rate the importance of each skill for a given event. These two ratings will be systematically matched to determine the most suitable volunteers for each opportunity. This dual-scale approach moves beyond basic keyword matching by properly accounting for both volunteer competency and event requirements, leading to better-qualified matches. Furthermore, the system allows volunteers to update their skill proficiency as they gain more experience, ensuring that the matching process remains accurate and responsive to both volunteer growth and organisational needs.

Next, the second objective is to enhance volunteer-event matching by developing a distance-based algorithm that prioritises nearby events while maintaining flexibility for farther ones. This algorithm will calculate proximity scores to recommend conveniently located events first, reducing commute-related dropouts. Besides, this algorithm also balances both suitability (skills) and accessibility (proximity) by integrating with the weighted skill-matching system. By addressing geographical barriers, the system improves participation rates and volunteer satisfaction, ensuring opportunities align not just with competencies but also practical logistics.

Finally, the third objective is to develop a system that recognises volunteer contributions in meaningful ways. By implementing a volunteer leaderboard, it can provide volunteers with a sense of accomplishment and recognition for their efforts. Volunteering hours and number of events participated will be logged upon attending events, completing tasks, and contributing to the organisation, which can be reflected on a public leaderboard. This gamification element encourages healthy competition, fosters motivation, and increases long-term volunteer commitment. By making recognition visible and tangible, the system aims to enhance volunteer satisfaction and retention.

## 1.4 Project Scope and Direction

The project aims to develop a comprehensive web-based Volunteer Management System (VMS). The VMS will be accessible via standard web browsers, designed to streamline volunteer recruitment, volunteer management, and event management. Key features of the system will include a weighted skill matching algorithm that evaluates both volunteer skill proficiencies and organisational skill requirements through weighted scoring, enabling precise matching of volunteers to suitable roles based on their capabilities and preferences. This targeted approach will optimise volunteer engagement while improving organisational efficiency in task allocation. Furthermore, the system will implement distance-based matching to prioritise conveniently located opportunities for volunteers, significantly reducing commute-related participation barriers. By calculating precise proximity scores, it ensures volunteers see nearby events first while retaining flexibility for those willing to travel farther. This geographical optimisation will then be combined with weighted skill-matching algorithm, creating a dual factor matching system that addresses both accessibility and qualification suitability. Additionally, the VMS will enhance the recognition system by incorporating gamification elements such as volunteer leaderboard to recognise and motivate volunteers. This

approach aims to motivate volunteers, foster a sense of accomplishment, and promote long-term commitment. Besides, common features such as events creation, events application, hours tracking, etc. are also provided. The project will be completed in phases—requirements gathering, design, development, testing, and deployment—ensuring a solution that effectively addresses existing gaps in volunteer management and meets the needs of diverse organisations.

## 1.5 Contributions

This project delivers meaningful impact through its innovative approach to two critical aspects of volunteer management: weighted skills-based volunteer-event matching and distance-based volunteer-event matching algorithm. The weighted skills-based matching system represents a major advancement over current methods by introducing weighted proficiency assessments, where volunteers self-rate their capabilities and organisers specify required skill levels, creating far more accurate pairings than simplistic keyword matching allows. This precision prevents the frustration of underqualified placements while ensuring organisations can properly utilise volunteers' full potential. Additionally, the distance-based matching algorithm is equally transformative where it calculates the precise distance metrics to prioritise conveniently located opportunities. Consequently, it effectively removes one of the most common barriers to consistent volunteer participation.

The combination of these two innovations creates a powerful synergy - the system does not just find skilled volunteers, but skilled volunteers who can realistically attend. For volunteer-reliant organisations like food banks, community shelters and youth programmes, this translates to substantially improved operational reliability. No longer will crucial roles go unfilled because qualified volunteers live too far away, nor will events suffer from last-minute cancellations due to impractical travel requirements. By solving both the "can do" and "can attend" aspects of volunteer placement, this project elevates volunteer management from an administrative task to a strategic function that maximises social good. What makes this project compelling is its immediate applicability - these are not theoretical improvements but practical, measurable enhancements that any organisation can implement to strengthen their volunteer programmes today.

## 1.6 Report Organisation

This report is organised into seven chapters: Chapter 1 Introduction, Chapter 2 Literature Review, Chapter 3 System Methodology/Approach, Chapter 4 System Design, Chapter 5 System Implementation, Chapter 6 System Evaluation and Discussion, and Chapter 7 Conclusion and Recommendation.

The first chapter introduces the project, covering the project background, problem statement and motivation, research objectives, project scope, contributions, and report organisation. Chapter 2 presents a literature review, analysing existing volunteer management systems and relevant development techniques, evaluating their strengths and weaknesses, and justifying the choice of technologies used in this project. Chapter 3 describes the system methodology and approach, including the incremental prototyping methodology, system flow, and use case diagrams. Chapter 4 details the system design, including module specifications, the novel event matching algorithm, and the design of weighted skill-based and distance-based matching mechanisms. Chapter 5 focuses on system implementation, discussing hardware and software setup, project configuration, database design, and implementation challenges encountered. Chapter 6 evaluates the system through use case testing, presents the testing outcomes, and assesses whether the research objectives were achieved. Finally, Chapter 7 provides the conclusion of the project, summarises the contributions, and provides recommendations for future improvements and enhancements.

# Chapter 2
# Literature Review

## 2.1 Existing Volunteer Management System Review

### 2.1.1 Review on VolunteerHub

VolunteerHub was originally developed in the mid-1990s to assist with volunteer registration at the University of Michigan's campus chapter of Habitat for Humanity. It has grown significantly since its inception. Now, it has become a leading volunteer management software for nonprofits and other organisations which produced by BetterGood. With a cloud-based approach, it provides clients with the flexibility and control needed for effective volunteer management. Over the past 20+ years, it has supported many organisations in managing billions of volunteer hours, millions of users, and events. Today, it helps thousands of organisations worldwide to streamline volunteer recruitment, registration, tracking, and reporting [4]. VolunteerHub integrates volunteer management, event management, and volunteer communication into a cohesive volunteer management system.

To manage volunteers efficiently, VolunteerHub has many features. Firstly, VolunteerHub's **volunteer recruitment features** optimize the recruitment process to allow clients to spend more time focusing on building genuine and lasting relationships with users. The recruitment features are aided by event-specific landing page and volunteer groups function [5]. Event-specific landing page can inspire more volunteers to sign up by offering a customisable engaging and compelling landing page. At the same time, Volunteer groups allow organisations to target specific recruitment group based on specific needs. Besides, VolunteerHub implements list, calendar, and map view to ease the **volunteer scheduling process** [6]. For instance, volunteer events displayed on a map view shown in figure 2-1 assist volunteers to visualise schedules, avoid conflicts, and plan their participation over time.

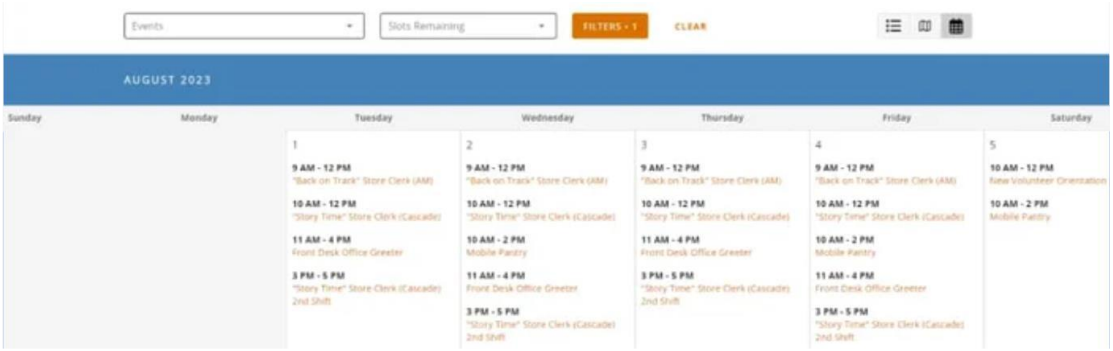



Figure 2-1 Volunteer events displayed on map view [6].

VolunteerHub also facilitates the **volunteer hour tracking process** by eliminating the manual hour tracking. There are various methods for tracking volunteer hours, including manual entry, automatic tracking upon registration, volunteer self-reporting, contactless tracking via QR codes, and through administrative interface onsite [7]. In order to store and keep track all the data and records, VolunteerHub use a **centralised cloud-based database** to increase the data accessibility. Due to the cloud-based solution, the database is up to date with the latest version without any software installation [8]. Furthermore, VolunteerHub's **reporting features** provides a range of standard reports, allowing for the analysis of data stored within the platform. Not only that, but every field in the database also serves as a data point that can be utilised to generate reports customized to specific needs. The custom reports can be easily created, and the template can be saved for future use [9].

Apart from the common features, VolunteerHub has some special features to manage volunteers as well. For example, VolunteerHub implements a **rewards and recognition feature** to boost volunteer engagement and retention. This is done by creating a points-based reward program and volunteer leaderboard. Volunteers can earn points by completing events and the points can be used to redeem branded merchandise, swag, and services. Besides, the volunteer leaderboard enables volunteers to see their standing relative to other volunteers to foster a sense of friendly competition that drives commitment within the organisation [10]. Moreover, **volunteer fundraising feature** in VolunteerHub helps to convert volunteers into donors. Clients can create numerous custom donation pages, each featuring unique request messages, thank you messages, and donation amount options. The donation page can be set to automatically prompt to volunteers whenever they sign up for specific events. The user-friendly page can raise attention of volunteers and subsequently increase the donation. If a donation is received, an automated thank you message will be sent to the donor [11]. Last but not least,

VolunteerHub's **liability waiver feature** enables clients to create, store, and manage multiple waivers for specific events, protecting organisations from legal claims and financial liability in the event of accidents. By using this feature, the waiver can be customised to cover specific risks associated with the tasks that volunteers will perform [12].

Next, VolunteerHub also includes several features to facilitate the event management process. First of all, VolunteerHub provides **multi-event editor** to allow modification on multiple volunteer events at once. For example, the administrator can update event information like starting date to several events at the same time without doing it redundantly. The multi-event editor comes in with a filter and find function which enable clients to use several filtering options to quickly identify the events that should be make changes. The filter options include date range, event contact, start and end time, event groups, etc [13]. Moreover, VolunteerHub's **group manager feature** allows organisers to manage multiple groups of volunteers for different events. Organisers can create different groups for volunteers based on their skills and interests. This is especially useful when it comes to volunteer recruitment. This is because the events that need special skills can be easily send to the targeted group that possess the skills and only the qualified members are available to register for the events [14]. Apart from that, **configurable forms feature** provided by VolunteerHub ease the process of data collection. This is due to the forms can be customised to reach different groups of volunteers. Therefore, the data collected from the right group of volunteers can be more accurate and thus driving better decision making [15].

In addition, volunteer communication is a crucial component in volunteer management system as it fosters strong relationships between the organisation and its volunteers. Therefore, VolunteerHub integrates **email messaging, text messaging, and social media** to enhance volunteer communication. With VolunteerHub, clients can effortlessly send both one-to-one and automated emails to individual volunteers, specific volunteer groups, or all volunteers within the organisation. Besides, there is no need for a separate messaging system because emails sent through the platform appear to come directly from the organisation's admin. This setup allows all email replies to go directly to the admin's inbox, ensuring that interactions between admin volunteers are direct and efficient [16]. However, some volunteers might miss some important emails since not everyone will check email regularly. Hence, VolunteerHub provides a text message feature to allow direct communication. It can be used to send text

reminders, notifications, and also automated thank you messages to targeted volunteers efficiently. Finally, VolunteerHub integrates social media in its system to enhance volunteer communication. For instance, an organisation can include social media sites to allow volunteers to know more about them. Volunteers can also share events on their social media accounts and thus increasing events exposure [17].

### 2.1.2 Review on POINT

POINT is a collaborative volunteer platform designed to connect communities with local causes for free, with a mission centred around empowering nonprofits, particularly those led by women. The platform provides powerful, time-saving technology to support these organisations, helping them maximize their impact in the community. Besides, they also offer pay version that include additional paid features and services with enhanced capabilities to support organisations that need more advanced tools. POINT Technologies operates the technical platform, while POINT Global, a nonprofit entity, manages donations and program-based support. Together, they aim to equip local leaders with the tools they need to uplift their communities [18]. POINT focuses on five features to support their volunteer platform which are recruitment, management, engagement, tracking, and integration.

Firstly, POINT's **website integration feature** simplifies volunteer recruitment by allowing organisations to create events once on the POINT dashboard, which then automatically updates the volunteer signup page on their own website. This means that organisers only have to create the events once and volunteers can use POINT to sign up for events while staying on the website without being redirected elsewhere [19]. This website integration significantly reduces administrative effort and save volunteers time in recruitment process. Furthermore, POINT allows **easy event sign up and check in** to ease the volunteer recruitment. This is because organisers can easily create and publish events while volunteers just have to tap on a button to register for the events they are interested in [20]. The sign up process are even enhanced by POINT’s cause-based local volunteering opportunities. This feature allows volunteers to select location and causes they are engaged in, and the system will suggest the related events in volunteers’ personalised feed [21]. For the check in feature, admins can easily view signed up volunteers and manually adjust their status whether they are checked in, no showed, or cancelled [20]. Not only that, POINT is also **available on both web platform and mobile application** to reach wider users, allowing volunteers to access it using preferred devices [21].

Finally, POINT include a **page feature** called Volunteer Hub to allow volunteers to know more about your organisation during the recruitment process. It is basically a customised profile page designed for newly registered nonprofit organisation in the system and all the details about the organisation can be found in the page. The profile page can be edited easily as well by just picking and dropping the essentials [22].

Next, POINT also have some features to aid the management process. One of the features is **volunteer activity profiles** which plays an important role in volunteers management. The profiles provide a comprehensive view of each volunteer's engagement in specific events. By personalising these profiles, organisers can track essential details such as contact information, attendance rates, total volunteer hours, donation history, and involvement in events and causes. Additionally, profiles enable organisers to engage volunteers meaningfully by analysing their participation patterns and preferences, allowing organisers to tailor events based on their interests and send personalized thank-you notes [23]. Besides, **volunteer programs feature** on POINT offers a streamlined approach to manage events according to different aspects of organisation's mission. For example, organiser can separate homeless services event into good bank program and free clinic program. Different programs can be assigned with different volunteers based on their skills set [24]. Furthermore, POINT's **opportunity, event, and shifts features** simplify event management and enhance volunteer engagement. This feature allows organisers to break long or multi-day events into different shifts, allowing volunteers to choose specific time that suit them. Shifts also enable organisers to manage different roles within an event effectively by creating different named shifts for specific tasks, allowing volunteers to sign up for roles in advance [25]. For instance, a shift 10am to 12pm will for meal giveaway role while another shift 12pm to 2pm will for cleanup role. This flexibility will make the events more appealing to volunteers. Not only that, **calendar view** on POINT is also a powerful tool for organisers to manage events. It offers admins a clear and organised visual of all upcoming events, providing quick access to event details, volunteer signups, and check-in capabilities, thus streamlining the event management [26]. Moreover, POINT's **Cohost features** provide powerful collaboration tools that enhance partnerships with other organizations, including local nonprofits, schools, and companies. This feature allows organiser to co-host events with other organisations on the platform. Adding a co-host enables shared management of volunteers, including inviting participants, communicating with them, editing event details, and checking volunteers in [27]. Finally, POINT's integration with Checkr streamlines the **background**

**check** process, making it fast and efficient. With this integration, organisers can order background checks for volunteers in seconds, either individually, for groups, or automatically as volunteers register for events [28]. It allows organisations to easily verify the history and credentials of an individual, thus ensuring that the individuals they bring on board are trustworthy and reliable for the events.

Apart from management, engagement is also focused by POINT. In order to allow organisers to easily engage to volunteers, POINT provides a comprehensive **email management system**. There are two sent box for emailing in POINT, which are self-sent box and team sent box. Self-sent box allows organisers to keep track and email entire contact database, event lists, groups, specific volunteers in a program, or individual volunteers directly from the platform. On the other hand, team sent box ensures that all admins within the organisation stay informed about volunteer communications. This shared inbox allows everyone on your team to see sent emails, eliminating confusion about whether a message has been sent. POINT also provides auto signup confirmations, reminders, and thank you emails to reduce the burden of organisers [29].

The fourth focus of POINT is about volunteer tracking. POINT simplifies the process of **tracking volunteer hours** by automatically recording them whenever a volunteer checks in to an event. These hours are immediately added to both the volunteer's impact profile and your organization's overall statistics. The system accurately tracks hours even if a volunteer works just one shift at a multi-day event, ensuring precise data without the need for manual calculations [30]. POINT provides **two ways to check in which are kiosk and QR code.** Volunteers can either use an organisation's computer via kiosk or scan a custom QR code directly from their smartphones to check in without needing an admin to present [31]. Not only that, POINT also consider volunteers who work outside events by offering **volunteer self-submission feature**. It allows volunteers manually submit contribution hours and admins can approve so that it is recorded [32]. Last but not least, **auto impact report feature** of POINT streamline data management for organisations by automatically tracking volunteer and donor activities, and generating comprehensive statistical reports. These reports are accessible on the organisation’s dashboard, showcasing the organisation's impact in granting applications and donor communications. Filters are also available in generating statistical report to provide flexibility in presenting insights [33].

The final feature that POINT includes to maintain its volunteer platform is integration. POINT's **donation button and form integrations** simplify fundraising by allowing organisations to connect their existing fundraising tool or choose from a list of integrated platforms. The setup is user-friendly to integrate with tools like PayPal, Venmo, Givebutter, Donorbox, DonorPerfect, and Donately. Once integrated, donations made through POINT will direct donors to pre-existing donation forms and all data will be streamlined with the system used. Additionally, POINT offers exclusive discounts on fundraising tools for organisations without an existing tool, making it easier to expand their donor base [34].

### 2.1.3 Review on GetConnected

GetConnected by Galaxy Digital is a powerful volunteer management software designed to strengthen communities by connecting people, organizations, and resources through simple and effective technology. It is rooted in a vision of contributing to a healthier and more positive society. GetConnected empowers nonprofits, corporations, and campuses to efficiently manage volunteerism, track impact, and communicate their successes. With a strong commitment to social change and community engagement, GetConnected has supported millions of hours of volunteer work, helping organizations make a lasting difference in their communities [35] It divides its volunteer system into five sectors working with each other, which are volunteer management, scheduling & events, communication & engagement, reporting & data, and integrations & support.

Firstly, GetConnected offers **volunteer hours tracking** ability for individual volunteers, teams, and groups to manage volunteer activities. The system's real-time tracking ensures that hours are accurately recorded from check-in, bulk assignment and approval functions [36]. The automated tracking and easy approval processes have facilitated the volunteer management process. Next, GetConnected provides **skill-based volunteerism** tool to allow organisers to collect and review volunteers' qualifications during registration, ensuring that only qualified volunteers participate in specific events [37]. Besides, GetConnected's **volunteer group engagement feature** allows organisers to easily create more specialised team for different groups to enhance the management process [38]. Moreover, GetConnected provides **volunteer liability waiver** and **eSign documents** to ensure the protection of organisations. By integrating electronic waivers into the volunteer signup process, volunteers are unable to sign up for the event until the waivers are electronically signed. The documents signed will be automatically

saved to each volunteer's profile, making information readily accessible and confidential [39, 40]. These features have facilitated the volunteer management process.

Next, scheduling & events play an important role in this volunteer management system as well. GetConnected's **event landing pages** allow admins to create compelling event pages, display RSVP (Respond, if you please) options, and include essential event details such as time, location, and contact information for volunteers. Integration with Google Maps also makes it easy to add maps to events, and the platform supports both public and private event access [41]. Besides, GetConnected incorporated **volunteer calendar feature** for volunteers to browse and sign up for events in a clean and interactive calendar format. Therefore, volunteers can easily access detailed events descriptions and additional information [42]. Furthermore, GetConnected also facilitates the **event management** and **volunteer scheduling** process by using an easy-to-use dashboard. Organisers can easily view open shift slots and send invitations to schedule the available volunteers [43]. In addition, GetConnected offers **volunteers check-in feature** to ensure that every check-in process is smooth and hassle-free. With features like check-in kiosk and mobile app, volunteers can easily sign in, track their hours, and receive reminders for their shifts [44]. Apart from that, GetConnected's **automated notification feature** provides over 50 customisable notifications to help organisations to manage their volunteers to ensure that they are keep informed and engaged. These automated notifications can be used in many conditions such as welcoming new users, reminding volunteers of upcoming events, or sending follow-up surveys [45]. Finally, GetConnected partnered with Stripe, an online payment processing technology to allow organisations to securely receive donations directly through their platform. This has assisted organisers in managing and keep tracking their donors and amounts in just single place [46].

Moreover, communication and engagement are also focused by GetConnected. It provides a **free mobile application** to maximise volunteer engagement by providing an efficient way to track hours, view schedules, check in events, etc [47]. Besides, GetConnected's **event promotion feature** allows customisation of landing pages to highlight key jobs and events. Then, the page will be advertised to the right volunteers with the necessary skills and interests, thus ensuring targeted volunteer engagement [48]. Not only that, GetConnected also offers **volunteer profile** and **volunteer rewards & recognition feature** to maintain volunteer engagement. Volunteer profile feature offers a personalised experience for volunteers, serving

as a portal for managing their involvement and contributions. Through this comprehensive profile, volunteers can easily track their impact over time, manage their causes, and receive tailored opportunity recommendations based on their interests and skills [49]. On the other hand, volunteer rewards and recognition feature offers an engaging way to acknowledge volunteer contributions. By setting participation benchmarks, organisations can automatically reward volunteers with custom badges, recognising their achievements and fostering a sense of accomplishment. This gamified approach not only shows appreciation but also motivates repeat volunteerism and increased participation [50]. Last but not least, GetConnected platform powerful communication tools enable organisations to maintain communication with their volunteer base through targeted **email and text messaging**. The built-in eBlast feature simplifies mass or targeted email messages by allowing personalised emails. For instance, organisers can send emails to hundreds of recipients by auto filling their first name and other specific information [51]. For urgent or time-sensitive communications, the integrated text messaging tool provides a direct line to volunteers, ensuring important announcements and reminders are received and acted upon quickly [52].

Additionally, provides robust volunteer reporting features to empower organisations to efficiently track, analyse, and present volunteer contributions. With automated hours logging, **volunteer hours report** can be easily generated to showcase volunteer contributions [53]. **Impact reports** offer a powerful way to highlight the positive changes your organisation is making in the community by providing a variety of reports that can help to demonstrate impacts [54]. Not only that, all the reports are **easily exportable** for internal evaluation and external communication [55]. Finally, the **volunteer resume feature** in GetConnected offers volunteers a visually appealing and downloadable resume that highlights their contributions and impact. This tool enables volunteers to showcase their volunteer experience effectively in the workplace, classroom, or community [56].

Last but not least, GetConnected focuses on integration and support. It simplifies social media strategy for organisations with its seamless **social media integration**. With these quick-share icons integrated throughout the platform, both organizations and volunteers can effortlessly share contents directly with their networks, boosting visibility and encouraging more volunteerism [57]. In terms of supporting, GetConnected have incorporated a **ReDi (Readiness Disaster intelligence) disaster management software** to support organisations across all

phases of disaster management. Before a disaster occurs, ReDi helps organisations register volunteers, track their training, and assign roles based on skills and qualifications. This ensures that when a disaster strikes, the organisation has a prepared and capable team ready to deploy. During a disaster, ReDi facilitates rapid mobilisation of volunteers through automated notifications via text and email. It helps in posting and promoting volunteer events in real-time, allowing for a swift response. After a disaster, ReDi aids in verifying volunteer hours and managing long-term recovery projects. It provides tools for tracking and reporting on multi-day or ongoing projects, ensuring that recovery efforts are sustained and well-documented [58].

### 2.1.4 Comparison between Existing Volunteer Management System

Table 2-1: Features comparison between Volunteer Management System.

| Feature | | VolunteerHub | POINT | GetConnected |
|---|---|---|---|---|
| **Volunteer Recruitment** | Event-specific landing pages | Yes | Yes | Yes |
| | Volunteer groups | Yes | Yes | Yes |
| | Volunteer skills | Yes | No | Yes |
| | Cause-based local opportunities | No | Yes | No |
| **Volunteer Management** | Volunteer profiles | Yes | Yes | Yes |
| | Rewards & recognition | Yes (points system, leaderboard) | No | Yes (badges) |
| | Volunteer fundraising | Yes | Yes | Yes |
| | Liability waivers | Yes | No | Yes (with eSign documents) |
| **Event Scheduling** | Multi-event editor | Yes | No | No |
| | Configurable forms | Yes | No | No |
| | Calendar view | Yes | Yes | Yes |
| | Shifts | No | Yes | Yes |
| | Event check-in | Yes | Yes | Yes |
| **Communication & Engagement** | Email messaging | Yes | Yes | Yes |
| | Text messaging | Yes | No | Yes |
| **Reporting** | Standard and Impact reports | Yes | Yes | Yes |
| | Volunteer Resume | No | No | Yes |
| **Integrations** | Website integration | No | Yes | No |
| | Background checks | No | Yes (Checkr) | No |
| | Social media integration | Yes | Yes | Yes |
| | Disaster management | No | No | Yes (ReDi) |

### 2.1.5 Critical Remark of Volunteer Management System

VolunteerHub, POINT, and GetConnected are reviewed and each of them offer useful features for volunteer coordination but suffer from significant limitations in their matching capabilities. All three systems provide basic volunteer recruitment tools like event-specific pages and group management, but their matching approaches remain underdeveloped. Although POINT's cause-based local opportunities allow volunteers to select preferred causes within their community, but its implementation falls short by relying on broad geographical categories rather than precise distance measurements. This approach fails to prioritise opportunities based on actual proximity, meaning volunteers may still see distant events before ones that are genuinely nearby. This oversight misses a key chance to reduce participation barriers by matching volunteers with conveniently located events.

The skill-matching implementations in VolunteerHub and GetConnected also demonstrate fundamental flaws in their technical design. They are relying on simple keyword matching rather than comprehensive proficiency assessment. This basic approach fails to account for critical dimensions of volunteer suitability, including skill competency levels and experience depth. As a result, the platforms' inability to implement weighted scoring mechanisms will generate matches that often poorly align with both volunteers' true qualifications and organizations' specific role requirements, leading to inefficient pairings that frustrate users on both sides.

Next, all platforms support volunteer profiles, but differ in rewards and recognition systems. VolunteerHub uses a points-based reward program and leaderboard to recognise volunteers' contribution while GetConnected rewards volunteers with badges. However, POINT does not have a formal rewards system. A rewards and recognition system are essential in volunteer management as it fosters motivation and long-term engagement. When volunteers feel appreciated for their efforts, they are more inclined to stay committed to the organisation. Acknowledging their contributions with points, badges, or public acknowledgement not only enhances morale, but also creates a sense of accomplishment and pride in their roles.

## 2.2 Techniques Review

### 2.2.1 React

React is a powerful JavaScript library developed by Facebook in 2013, designed specifically for building user interfaces [59]. It is particularly well-suited for single page applications (SPAs), a web app implementation that loads only a HTML document on the first request, then update the body content of the document using JavaScript. This approach eliminates the need for full page reload from server, thus providing a smooth experience for users [60]. React's rise in popularity can be attributed to its innovative approach to UI development which focuses on a **component-based architecture**, introduction of **virtual document object model (DOM)**, and **declarative syntax** [59, 61]. This combination has made React a dominant force in modern web development, providing developers with the tools to build dynamic and efficient web applications. The core features, programming language used, cross-platform capabilities, backend availabilities, and limitations of React will be focused in this review.

React's component-based architecture allows developers to break down complex UIs into reusable, self-contained pieces, known as components. These components can manage their own state and be composed together to form complex user interfaces. This architecture promotes reusability as each component can be reused across different parts of application and thus reducing code duplication. Moreover, separation of concerns also can be reached. This is because each component manages its state independently, so updating one component will not cause unintended changes in other components [62]. Next, virtual DOM is another standout feature of React. DOM is a programming interface for HTML and XML documents that represents the document's structure as a hierarchical tree of objects, where each object corresponds to a part of the document. The DOM provides a way for programs to access and manipulate the content and structure of a document dynamically [63]. However, DOM manipulation might be slow and inefficient because it can trigger reflows and repaints. A reflow recalculates the layout of the entire webpage, which can be time-consuming, while a repaint updates the visual appearance of elements [64]. Virtual DOM introduced by React solves the problem by creating a lightweight copy of actual DOM. When a component's state changes, React updates the virtual DOM first and then determines the minimal set of changes needed to update the real DOM. This results in significant performance improvements, especially in applications with frequent updates [63]. Additionally, React's declarative syntax simplifies the process of creating user interfaces by allowing developers to describe what the UI should look

like based on different states, then React will automatically updates the DOM when the state changes. This declarative approach contrasts with the imperative programming model, where developers need to specify step-by-step how to achieve the desired application state [65].

React is built on JavaScript library and JavaScript is one of the most widely used programming languages used to create dynamic and responsive web applications. Its ubiquity across web development environments makes React a natural choice for developers looking to build modern web applications. A JavaScript library is a collection of pre-written code that simplifies app development by providing ready-to-use patterns, components, and functions for various tasks on a webpage. Therefore, the use of JavaScript makes React to benefit from the vast ecosystem of JavaScript libraries and tools. Furthermore, React extends JavaScript with JSX, a syntax extension that allows developers to write HTML-like code within their JavaScript files [66]. JSX improves code readability and makes it easier to visualise the structure of the UI components directly in the code. Instead of separating HTML and JavaScript files, JSX combines them, making it clearer and more intuitive to see how the interface is built and how the components fit together.

While React is primarily a library for building web applications, React Native extends its capabilities to offer a powerful solution for cross-platform mobile app development. This is done by allowing developers to use the same React principles and JavaScript codebase across multiple platforms. This approach maximises code reuse, reduces duplication of effort, and ensures consistency in the user experience across different platforms. To conclude, React Native bridges the gap between web and mobile development by sharing a significant portion of code between platforms. This cross-platform approach reduces development time and effort, allowing for a more unified codebase and consistent behaviour across different devices [67].
Since React is a front-end JavaScript library, it does not inherently include backend functionality, but it can easily integrate with various backend technologies such as Node.js, Express.js, or any other RESTful API-based service [68]. Moreover, React works well with both relational databases and NoSQL databases such as MySQL, PostgreSQL, MongoDB, Firebase, etc. This flexibility enables developers to choose a backend technology that best fits their project's needs. Besides, React ecosystem also supports GraphQL, a query language and runtime designed for APIs that enables clients to request precisely the data they need. It is

compatible with a range of databases and backend technologies, facilitating efficient data retrieval and manipulation [69].

One of the key limitations of React is the high pace of development. Although high pace of development fosters rapid innovation, it can also become a significant drawback. The frequent updates and changes in the environment can make it difficult for developers to keep their skills current, requiring continuous learning and adaptation. This fast-paced evolution also contributes to poor documentation, as the technology often advances more quickly than official resources can be updated. As a result, developers frequently find themselves relying on self-written guides and community-generated content to stay informed about the latest tools and practices, which can be time-consuming and inconsistent. Another limitation is that React is primarily a UI library, meaning it focuses on the view layer of the application. Developers must rely on other libraries or frameworks to handle routing, state management, and other aspects of application development [70].

### 2.2.2 Flutter

Flutter is a powerful open-source framework developed by Google in 2017, designed to enable developers to create visually appealing, natively compiled applications across multiple platforms, all from a single codebase. By enabling developers to build, test, and deploy apps for mobile, web, desktop, and embedded devices, Flutter significantly enhances the development process. It also enhances developer experience with automated testing and developer tooling, ensuring production-quality apps. Besides, Flutter is trusted by numerous global brands and deeply integrated with Google's services such as Firebase, Google Pay, Google Maps, etc [71]. Since its release, Flutter has attracted a large and active community of developers due to its **widget-based framework** and **hot reload feature** [72]. In this review, the core features, programming language used, cross-platform capabilities, backend availability, and limitations of Flutter will be focused.

Widget-based framework of Flutter states that every component in Flutter is a widget, they are the building blocks of Flutter application. The screen's appearance is entirely determined by the selection and arrangement of widgets used in app development. Flutter widgets are primarily categorized into 14 groups based on the different functionalities such as accessibility, basics, animation, input, etc [73]. Flutter widgets are highly aligning to customisable and

reusable principles. It allows developers to adjust the widget appearance and behaviour, enabling developers to craft unique and visually engaging interfaces. Furthermore, developers can build custom widgets by combining existing one and reuse these custom widgets throughout the app or even in different projects [74]. However, widgets are immutable once they are created, meaning that their properties cannot be changed. A new widget with updated properties must be created if modifications on a widget's appearance is needed. This immutable property of widgets helps to reduce the chances of bugs caused by unintended changes [75]. Next, Flutter's Hot Reload feature allows developers to instantly see changes made to the code without the needs of restarting the entire application [76]. Hot Reload works by updating the running Dart Virtual Machine (VM) with new source code files, then rebuilds the widget tree to reflect changes immediately [77]. This feature accelerates the development process by enabling rapid experimentation and iteration on UI designs and functionality.

Next, Flutter uses Dart as its primary programming language. Dart is an open source and object-oriented language with a syntax similar to C, Java and JavaScript [78]. There are two major program compilation methods supported by Dart, which are Just-In-Time (JIT) and Ahead-of-Time (AOT). JIT compiles the code at runtime, offering flexibility and speeding up development cycles by allowing developers to test changes on the fly. It is useful for performance improvement during runtime, though it has a slower startup. In contrast, AOT compiles the code before deployment, ensuring faster startup times and more consistent performance without runtime delays [79]. AOT is typically used for production, where quick startup and stability are key, while JIT is beneficial during development for its real-time feedback capabilities. Moreover, Dart's asynchronous programming capabilities with 'async' and 'await' keywords enhances performance by allowing tasks to run concurrently without pausing the execution of the entire program. When a function is marked as async, it can include await expressions that pause the function's execution only until the awaited task completes, without halting the whole program. This means that while waiting for a network request or file operation, the app can continue to process other tasks and update the UI smoothly [80]. This approach prevents the app from becoming unresponsive and helps maintain a seamless user experience, making Flutter applications both efficient and user-friendly.

Moving on to cross-platform capabilities, this is why Flutter stands out from many others software development kit (SDK). One of the features that enable cross-platform development

in Flutter is single codebase. By maintaining a single code base and deploy it to multiple platforms, Flutter allows developers to build applications for iOS, Android, web, desktop, and embedded devices [81]. Furthermore, Flutter implements Skia as its graphics engine to display the visual components. Skia is a robust open source 2D graphics library created by Google. It offers standardised APIs that function across multiple hardware and software platforms, allowing developers to produce high-quality graphics in their applications. By using Skia, Flutter directly converts widgets into pixels that can be displayed on screen. This ensures that the UI looks and behaves the same on iOS, Android, Windows, macOS, Linux, and web, providing a seamless cross-platform experience [82].

While Flutter is primarily used for front-end mobile and web development, it also offers backend integration capabilities through various libraries and tools. Flutter typically interacts with backends through RESTful APIs, GraphQL, or Firebase, with Firebase being a particularly common choice due to its close integration with Google's ecosystem. Flutter supports both real-time databases and cloud functions via Firebase, which allows for the creation of serverless applications. Furthermore, Flutter developers can use third-party packages such as Dio or HTTP for API requests, enabling smooth communication between Flutter apps and backend services [83]. While Flutter doesn't directly handle backend processing, its ability to seamlessly integrate with a wide range of backend technologies makes it versatile for full-stack application development.

The first disadvantage of Flutter is large file size. Flutter tends to have a larger file size compared to native apps due to the inclusion of the Flutter engine in the app package, which may affect distribution and installation [84]. Flutter includes its own rendering engine, Skia and Dart runtime to ensure a consistent performance and appearance across platforms, which adds to the app's initial size. In contrast, native apps rely on the platform's existing libraries, resulting in a smaller base file size. Besides, the learning curve of Dart language is also one of the limitations. Despite the many advantages that Dart offers to Flutter developers, its relatively low adoption remains a challenge. Unlike more widely used languages such as JavaScript, Java, or C#, Dart is less commonly selected, especially by beginners entering the programming field [85]. As a result, developers must invest additional time and effort to become proficient in Dart before they can achieve full productivity with Flutter.

### 2.2.3 Technique chosen to develop Volunteer Management System

It is crucial to select a right framework to develop a volunteer management system (VMS), as it directly impacts the system's performance, scalability, and user experience. React and Flutter are two famous techniques for building robust applications. To decide which techniques to be used, comparisons such as architecture and component structure, rendering and performance, programming language used, cross-platform support, and backend integration will be taken into considerations. In this project, I will choose React to build my volunteer management system instead of Flutter.

For architecture and component structure, React uses a component-based architecture while Flutter uses widget-based framework. React's UI is composed of reusable components and these components can manage their own state and be nested together to create complex UIs. This modular approach ensures code reusability and separation of concerns. On the other hand, every component in Flutter is a widget, and the entire app interface is built by combining these widgets. These widgets are customizable and reusable, which makes them highly flexible for building user interfaces. Both component-based and widget-based approaches enable the creation of visually stunning and complex UIs while promoting reusability. However, Flutter's widgets do not inherently manage state separately, while React's components provide built-in state management. Therefore, React allows developers to manage the state directly within individual components, facilitating the creation of self-contained UI elements that can independently handle their own state.

Moving on to rendering and performance perspective, React employs a virtual DOM to optimise rendering efficiency. When a component's state changes, React first updates the virtual DOM and then calculates the minimal set of changes needed to update the real DOM. This algorithm helps reduce overhead associated with direct manipulations to the real DOM, indirectly enhance the system performance. Flutter, on the other hand, uses the Skia graphics engine to render its UI directly. Unlike React, Flutter does not rely on a Virtual DOM. Instead, it constructs a widget tree that is converted directly into pixels. This direct rendering approach allows Flutter to maintain consistent performance across different platforms. Besides, Flutter's hot reload feature enhances the development process by enabling instant feedback on changes without restarting the application. In terms of rendering and performance, both React and Flutter are performing well.

Next, React is built on JavaScript library, one of the most ubiquitous languages in web development. Hence, JavaScript's wide adoption means that there is a vast ecosystem of libraries, frameworks, and tools, making it easier to find resources and support. Not only that, React also utilises JSX, a syntax extension that allows HTML-like code to be written within JavaScript, which can enhance the visualisation of component structure and its layout. In contrast, Flutter uses Dart, a modern object-oriented programming language developed by Google. Dart offers both JIT and AOT compilation, providing flexibility during development with real-time feedback through JIT and optimised performance in production with AOT. Although both JavaScript and Dart have their own strengths, I prefer React over Flutter. This is because I already have a background in JavaScript and would rather leverage the extensive ecosystem and resources provided by JavaScript than invest time in learning a new language which I am not familiar with.

In terms of cross-platform support, React is primarily designed for web applications but it extends its capabilities to mobile development through React Native. React Native allows developers to use the same React principles and a shared codebase for building mobile apps for iOS and Android. However, desktop applications and embedded devices are not supported by React Native itself but require third-party integrations. On the other hand, Flutter provides a comprehensive cross-platform development environment by allowing developers to use a single codebase to build applications for iOS, Android, web, desktop, and embedded devices. This is achieved through its widget-based architecture and the Skia rendering engine, which ensures consistent UI and performance across all platforms. Flutter's single codebase simplifies development process, making it particularly appealing for projects that require extensive cross-platform compatibility. However, since I am planning to develop a web-based volunteer management system for this project, React is well-suited and adequate for my needs.

Finally, both React and Flutter excel in backend integration, although they cater to slightly different needs and ecosystems. React, with its extensive JavaScript ecosystem, integrates seamlessly with a wide variety of backend technologies, including Node.js, Express.js, and databases like MySQL, MongoDB, and PostgreSQL. It also has strong support for GraphQL, which enhances data handling flexibility. Conversely, Flutter offers robust backend integration through RESTful APIs, Firebase, and third-party packages like Dio. Firebase is particularly popular in the Flutter ecosystem due to its serverless and real-time capabilities, aligning with

mobile-first development. In summary, React provides broader backend options due to its vast ecosystem and compatibility with various technologies, while Flutter offers tighter integration with Firebase and simpler setups for mobile-first projects. Hence, React is a better option for developing a web-based volunteer management system.

To conclude, React is chosen to build my web-based volunteer management system instead of Flutter because of independent state management capabilities, experienced in JavaScript, web-based focusing, and broader backend options.

# Chapter 3
# System Methodology/Approach

This chapter describes the methodology applied in the development of the Volunteer Management System (VMS). This project adopts the incremental prototyping methodology, a variant of the Prototyping Methodology to ensure systematic development and early validation of system components. This approach is particularly suitable for VMS, as the system consists of multiple interconnected modules, such as organiser event creation, volunteer event participation, attendance, leaderboard, etc.

### 3.1 Incremental Prototyping Methodology

Incremental Prototyping involves breaking the final product into smaller, manageable prototypes that are developed, tested, and refined individually [86]. Each prototype represents a functional module of the system, such as authentication, event creation, event participation, etc. Instead of attempting to deliver the entire system at once, development progresses in increments where each prototype adds a usable component to the overall solution.

Once validated, these prototypes are incrementally integrated to form the complete system. This modular approach provides several benefits. Firstly, it allows for faster delivery of partial but functional system components, enabling the ability to visualise and interact with parts of the system early in the development cycle. Secondly, it reduces development complexity because the system is divided into smaller and focused parts, where each of them can be designed, implemented, and tested independently. Thirdly, it ensures early detection of issues, since problems identified in one prototype can be addressed before integration with others, reducing the likelihood of large-scale rework. In addition, the approach offers flexibility, as modifications or improvements can be incorporated into subsequent prototypes without disrupting the entire system. Ultimately, this iterative and modular process leads to a refined and user-centric final product, where feedback and improvements are continuously integrated, and risks are managed throughout the development cycle.

The methodology consists of the following phases:

1. **Requirement Analysis and System Partitioning**
    - System requirements were first gathered through literature review of existing VMS platforms and identification of organisational challenges.
    - The VMS was divided into key modules:
        - User Authentication (volunteer and organiser login/registration)
        - Event Creation (organiser event creation)
        - Event Modification (delete events, modify events, manage applications)
        - Event Listing (list events with matching feature)
        - Event Participation (volunteer participated events management)
        - Attendance (view past created events by organiser and mark volunteer attendance)
        - History (view participated events history by volunteer)
        - Profile (volunteer and organiser profile)
        - Leaderboard (volunteer leaderboard)
2. **Quick Design for Individual Modules**
    - Each module was initially designed through low-fidelity sketches and interface mockups.
    - The design focused on user interaction and data flow between modules rather than technical details.
3. **Prototype Development (Incremental)**
    - Prototypes for each module were implemented incrementally using React.js, Node.js, and MySQL.
4. **Testing and User Feedback**
    - Each prototype was tested individually for functionality and usability.
    - Feedback was collected from supervisor at each stage, ensuring alignment with requirements.
5. **Integration of Prototypes**
    - Once individual modules were refined, they were progressively integrated to form a unified system.
    - Careful planning of system architecture and database schema ensured that the modules fit together seamlessly.
6. **Final System Delivery**

- The integrated system was tested as a whole to verify data consistency, module communication, and overall usability.
- The final product represents the collective outcome of all incremental prototypes.

## 3.2 System Design Diagram

### 3.2.1 System Flow Diagram

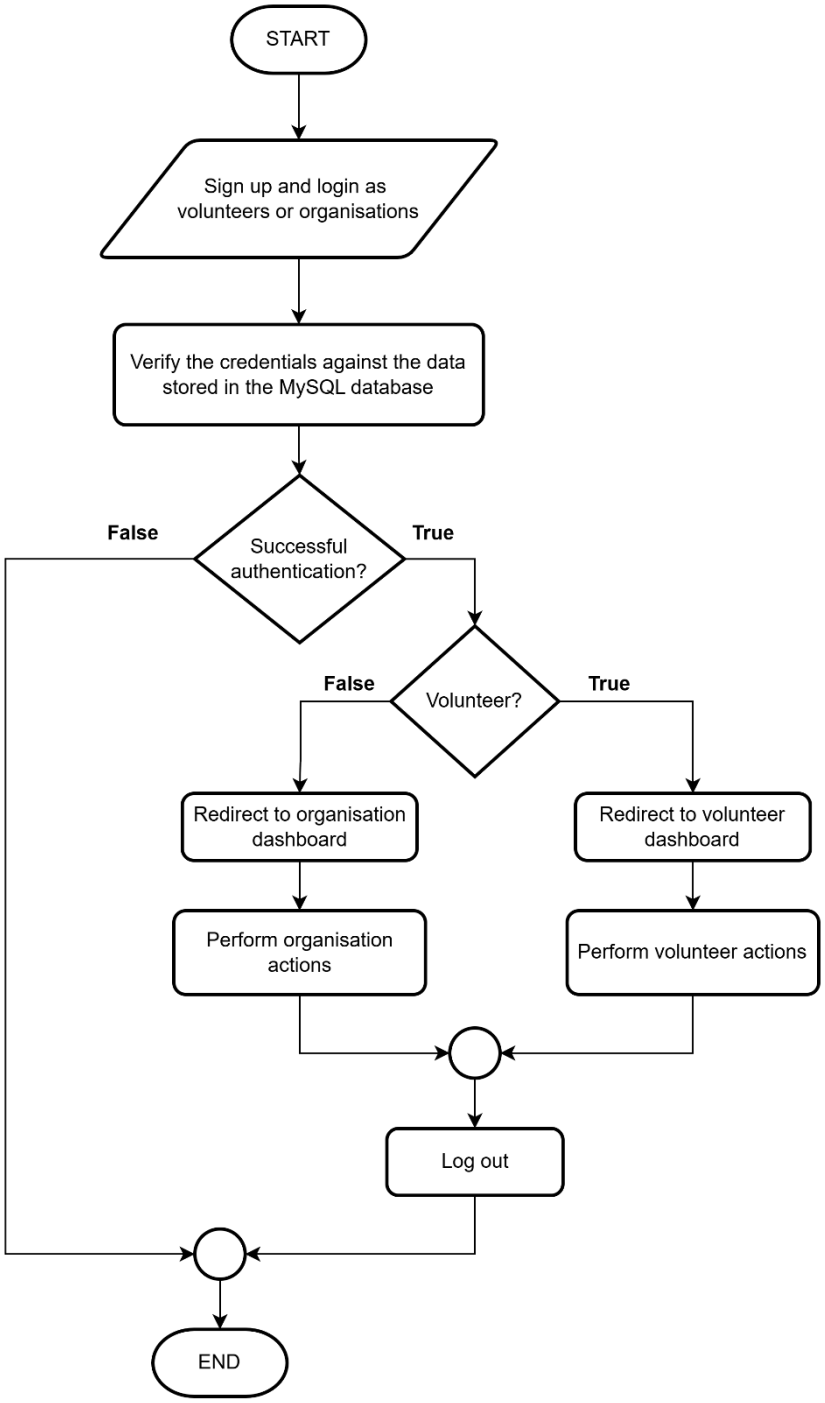



Figure 3-1: System flow diagram of proposed system.

The system flow of the web-based Volunteer Management System (VMS) begins when users access the application through a browser. Users are presented with the option to sign up or log in either as a volunteer or an organisation. Upon entering their credentials, the information is sent to the backend server, which is built using Node.js and Express.js. The server then verifies the credentials against the data stored in the MySQL database. If authentication is successful, users are redirected to their respective dashboards based on their roles. Volunteers can perform volunteer actions such as view available event lists, apply or leave events, view their profile, and check the volunteer leaderboard, all of which are powered by data retrieved through API calls to the backend. On the other hand, organisations have access to functionality that allows them to create new events, edit or cancel existing ones, monitor volunteer participation and manage volunteer attendance. All the activities that can be done by volunteers and organisations will be further discussed in the next section – Use Case Diagram. All interactions between the frontend and the backend are handled through HTTP requests, ensuring smooth communication and data updates in real time. This system flow ensures that each user has tailored experience while maintaining efficient data management and interaction throughout the platform.

### 3.2.2 Use Case Diagram

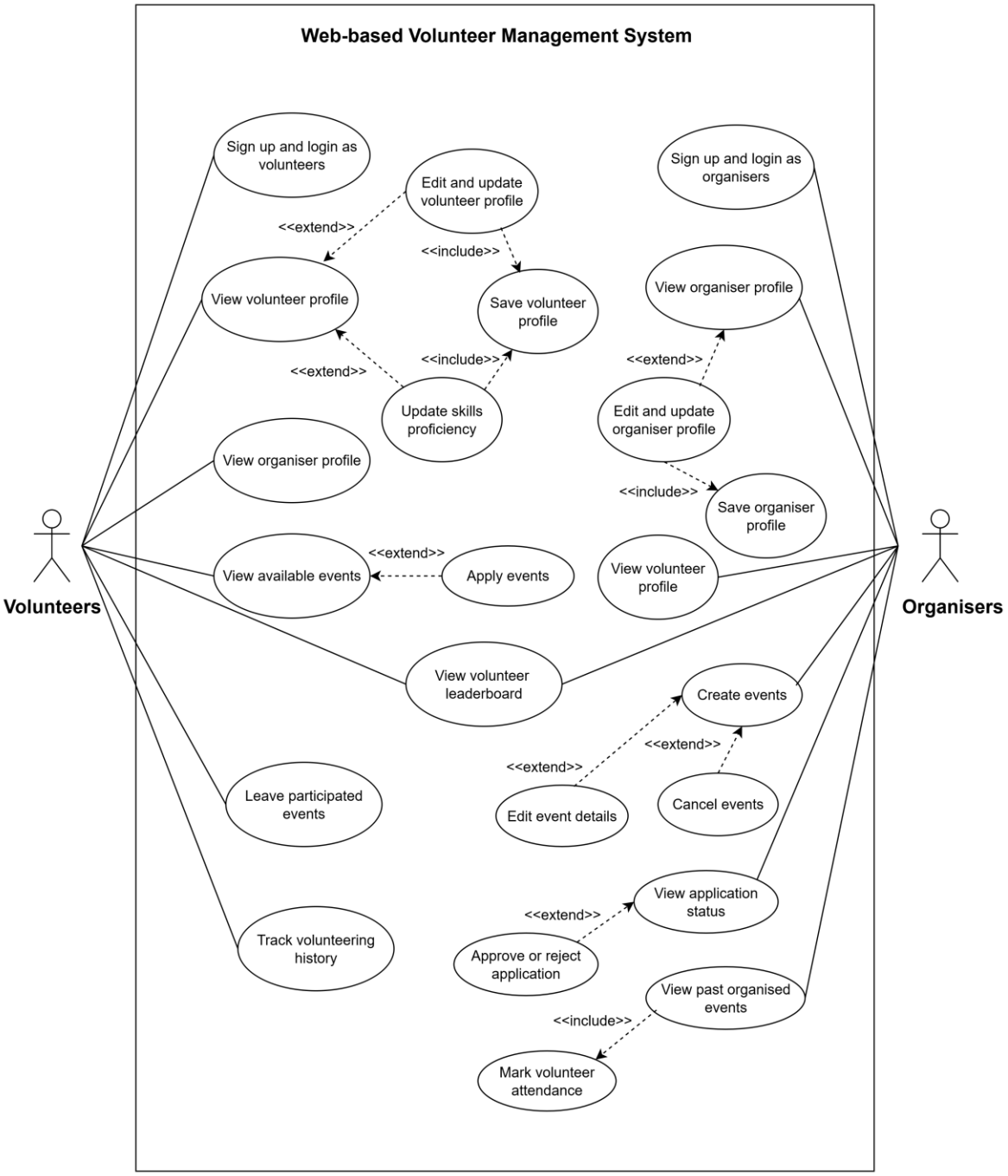



Figure 3-2: Use Case Diagram of proposed system.

The proposed web-based volunteer management system involves two actors, which are volunteers and organisers. For both volunteers and organisations, they can sign up or login their respective user accounts, view each other profiles, edit and update profiles, view volunteer leaderboard, and track events history. For first time log in volunteers, they are recommended to perform a self-evaluation on skills proficiency and update address in their

profile so that they can enjoy the skill matching feature. Furthermore, volunteers can view and search available events and apply for interested events. For events that have been approved, volunteers can choose to leave the events if any uncertainty arises.

On the other hand, organisers can create event details and specify skills importance level required for the events to optimise the event-volunteer matching performance. Created events can also be cancelled if any uncertainty happens. Moreover, organisers can view the application status for each event and choose to approve or reject the applicants. After the events have completed, organisers have to mark the attendance of volunteers so that all the records are logged and volunteer leaderboard can be updated to recognise the contribution of volunteers.

# Chapter 4
# System Design

## 4.1 System Block Diagram

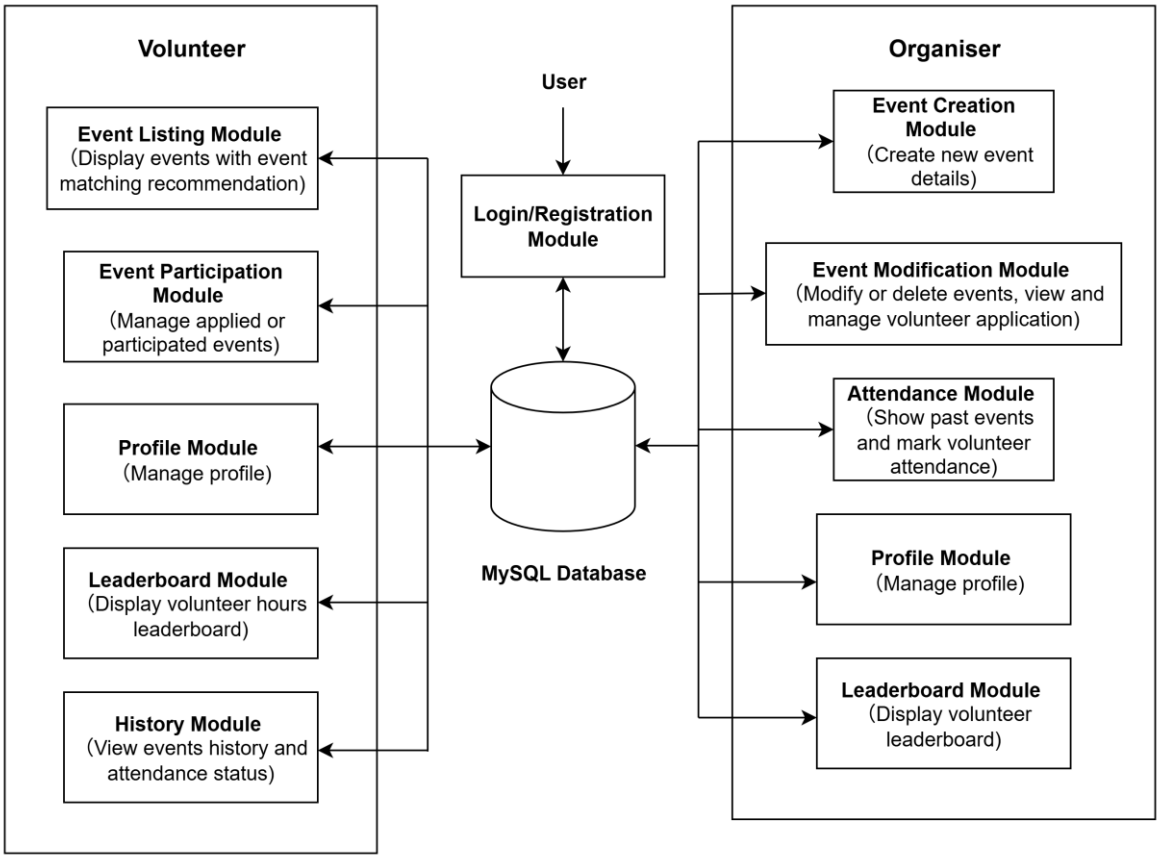



Figure 4-1: System Block Diagram of proposed system.

The system block diagram above illustrates the modular structure of the proposed Volunteer Management System. It is divided into two main user roles: volunteer and organiser, with a shared Login/Registration Module and centralised MySQL Database. The Login/Registration Module acts as the entry point, authenticating users and granting access to role-specific modules. For volunteers, the system provides an Event Listing Module to display available events with matching recommendations, an Event Participation Module to manage applied or participated events, a Profile Module for managing personal details and skills, a Leaderboard Module to display volunteer rankings based on volunteer hours or events participated, and a History Module to view past participation records. On the organiser side, the system includes an Event Creation Module to register new events, an Event Modification Module to edit or

delete events and manage volunteer applications, an Attendance Module to view past evets and record volunteer attendance, a Profile Module for organiser information, and a Leaderboard Module to track volunteer contributions. At the core, the Database stores and manages all relevant data, ensuring both volunteer and organiser modules are consistently updated and synchronised.

### 4.1.1 Login/Registration Module

The Registration Module enables new users to sign up for the Volunteer Management System by entering their username, email, mobile number, password, and selecting a role as either a volunteer or an organiser. The form includes validation to ensure that all fields are completed correctly, such as checking the email format and the length of the username. Once submitted, the system saves the user's information into the database and provides immediate feedback to confirm whether the registration was successful or if corrections are needed. For convenience, the form also provides a redirect link for users who already have an account, ensuring smooth navigation to the login page. This module ensures secure and structured account creation while enhancing the user experience through clear validation and feedback mechanisms.

The Login Module allows registered users to securely access the system by entering their email, password, and role. The system verifies the information against records stored in the database, ensuring that only valid users can gain access. Upon successful login, users are directed to their respective dashboards—volunteers are taken to the volunteer page, while organisers are directed to the organiser page. Clear error messages and feedback are provided when login details are incorrect, helping users understand what needs to be corrected. This module works hand-in-hand with the Registration Module, completing the core user authentication flow that ensures secure and role-specific access to the system.

### 4.1.2 Event Creation Module

The Event Creation Module allows organisers to create and publish volunteer events in the system. Organisers are required to provide key details such as the event title, description, date, time, location, city, and the number of volunteers needed. In addition, organisers can assign skills to each event and set their importance levels using a proficiency scale from novice (1) to expert (5). This ensures that events are clearly defined with both general information and skill requirements. Once the organiser submits the form, the event data is validated and sent to the

backend for storage. A confirmation is shown upon successful creation, and the new event is added to the organiser's list. This module ensures that organisers can efficiently define opportunities while helping volunteers find events that match their skills and interests.

### 4.1.3 Event Modification Module

The Event Modification Module is designed to allow organisers to efficiently manage the events they have created. One of its primary features is the ability to view detailed event information, including the event title, description, location, date, time, required and current number of volunteers, and associated skill categories. Organisers can also track volunteer application statistics for each event, such as the number of pending, approved, and rejected applications, helping them monitor participation and engagement.

Another key feature is the ability to manage volunteer applications directly from the module. Organisers can view individual applications in a modal interface, which displays the volunteer's profile, contact information, matching score, motivation message, and application status. From here, organisers can approve or reject applications, with changes immediately reflected in the event's statistics and volunteer counts, ensuring accurate real-time tracking.

The module also supports inline event editing, allowing organisers to update event details without leaving the page. Editable fields include event title, location, city, date, time, description, required volunteers, and skills importance level. Besides, the module also enforces validation rules such as ensuring the required number of volunteers is not lower than the current number, maintaining data integrity.

Finally, the module provides event deletion functionality, enabling organisers to remove events with a confirmation prompt. Events are also sorted based on pending and total applications, helping organisers quickly identify events that need attention. Together, these features give organisers a centralized and intuitive interface to manage events, track volunteer participation, and oversee applications efficiently.

### 4.1.4 Attendance Module

The Attendance Module is designed to display past organised events and help organisers to efficiently track and manage volunteer attendance for events. It displays a list of past events

with key details such as the event name, date, time, location, and the total number of volunteers. Organisers can select an event to view a list of volunteers registered for it, including their profile information, email, phone number, and current attendance status.

Furthermore, the module allows organisers to mark individual volunteers as present or absent, as well as apply attendance changes to all volunteers at once. Volunteers whose attendance has not yet been recorded are clearly indicated, helping organisers quickly identify unmarked participants. Attendance statistics are also provided, summarising the number of volunteers present, absent, or unmarked, along with the total volunteer count for the event. To improve usability, the module also includes search and filtering features that allow organisers to find specific events or volunteers based on name, date range, or attendance status. Volunteers can be filtered to show only present, absent, or unmarked participants, enabling more efficient management during busy events.

Overall, the module provides a streamlined and intuitive interface for organisers to accurately track and manage volunteer attendance, make bulk updates, and maintain up-to-date records, ensuring smooth event operations and effective volunteer management.

### 4.1.5 Profile Module

The Profile Module allows both volunteers and organisers to view and manage their personal information. For volunteers, the module displays details such as their name, email, phone number, profile picture, and their self-rated skills. Additionally, volunteer statistics including events joined, attendance rate, and hours volunteered are also displayed to provide an overview of volunteer engagement. Volunteers can view a summary of their skills to keep track of their skill proficiencies. Volunteers can also update their personal information and skill proficiency level, with all changes saved to the MySQL database to ensure their profile remains current.

For organisers, the profile shows personal details including name, email, phone number, and organisation name. Organisers can edit and update their contact information and view organiser statistics, such as the total number of events created, number of volunteers managed, and events hours organised. All the updates are stored in the MySQL database as well, keeping organiser information accurate and up to date.

Overall, the Profile Module provides a clear and intuitive interface for both volunteers and organisers to maintain accurate personal data, track skills level and performance metrics, and support better volunteer management and engagement.

### 4.1.6 Leaderboard Module

The Leaderboard Module provides a way to track and showcase volunteer engagement within the system. It displays a ranked list of volunteers, showing details such as profile picture or initials, total hours volunteered, number of events attended, and recently participated activity. For the top 3 volunteers, it also highlights their weekly hours and weekly events participated with a small green indicator, showing recent contributions in addition to overall performance. Furthermore, filters are also applied to the leaderboard, allowing users to view rankings by period—week, month, year, or all time—and sort by either hours volunteered or events attended, giving insight into their progress and achievements. Volunteers can see their own ranking highlighted and compare their contributions with others.

For organisers, the module offers a comprehensive overview of volunteer activity across events. They can quickly identify top-performing volunteers, monitor hours and event participation, and view recent activity updates. The module helps organisers make informed decisions for volunteer recognition, engagement, and event planning.

Overall, the Leaderboard Module is designed to create a gamification system that motivates volunteers by showcasing their contributions, highlighting top performers, and providing insights into individual progress.

### 4.1.7 Event Listing Module

The Event Listing Module allows volunteers to view a card list of all available events. Each event card displays key information such as the event title, organisation, description, date, time, location, number of volunteers required, and required skill tags with importance levels. If volunteers have rated their skills proficiency and updated their address in their profile, an event matching recommendation feature is provided. The events will display a matching score and be sorted by the score, indicating how well the volunteer's skills and location availability align with the event. The event matching algorithms will be further discussed in 4.2 Event Matching Algorithm Design. Furthermore, a tooltip is provided to explain the score as follows:

- **80 and above:** Excellent Match
- 7**0–79:** Good Match
- **50–69:** Fair Match
- **Below 50:** Not Recommended

After reviewing the event details and matching scores, volunteers can choose to apply for events that interest them. During the application process, they are encouraged to submit a motivation message, explaining their interest or suitability for the event. This message helps organisers better understand the volunteers' intent and commitment, improving the selection and approval process. Once applied, events are removed from the list to prevent duplicate applications. The module also filters out events that are full or already applied to, ensuring volunteers only see relevant opportunities.

Overall, the Event Listing Module streamlines event discovery and application, helping volunteers to find suitable events efficiently while providing insights into their compatibility with each opportunity.

### 4.1.8 Event Participation Module

The Event Participation Module allows volunteers to view and manage their current applications and approved participations in events. It displays each event's key details, including the title, description, organiser name, date, time, location, required and current number of volunteers, and the event's skills involved.

Each application is clearly labelled with the current application status such as Pending, Approved, or Rejected, supported by color-coded visual cues and status icons for quick recognition. Approved applications display a green status, pending applications are marked in yellow, and rejected applications are indicated in red. Action options are also provided based on the status:

- Pending applications: Volunteers can cancel their application using the Cancel Application button.
- Approved applications: Volunteers can leave the event using the Leave Event button.
- Rejected applications: The module displays the rejection outcome but disables further actions.

Moreover, a confirmation modal will appear before leaving or cancelling, clearly outlining the consequences of the action and showing the volunteer's original application message for context. This ensures volunteers make informed decisions before proceeding.

Overall, the Event Participation Module offers a simple, structured interface for volunteers to keep track of their applications, manage their commitments, and stay informed about their engagement with upcoming events.

### 4.1.9 History Module

The volunteer History Module provides volunteers with a detailed record of their past participation in events. It displays each event's title, description, organiser name, date, time, location, skills involved, and the volunteer's attendance status. Attendance is clearly indicated as Present, Absent, or Not Marked, with color-coded visual cues for easy recognition.

Moreover, filters for attendance status and time period are provided. Volunteers can filter their history by selecting attendance status such as "All", "Present", "Absent", and "Not Marked" or filter by time period such as the last 3 days, last week, last month, or last 6 months. This feature makes volunteers easy to quickly find and review specific events of interest.

Overall, the volunteer History Module offers an intuitive interface for tracking personal volunteer engagement, reviewing past contributions, and reflecting on experience gained across different types of events.

## 4.2 Novelty Event Matching Algorithm Design

### 4.2.1 Weighted Skill-Based Volunteer-Event Matching Algorithm

#### 4.2.1.1 Skill Level Identification and Definition

Skill levels provide a structured way to measure and differentiate the expertise of individuals, which is crucial for effective planning, team composition, and skill development. According to iCombine [87] which is inspired by the Dreyfus model of skill acquisition, these levels can be categorised as follows:

1. **Novice** – Has only minimal or textbook knowledge and relies heavily on supervision, often struggling to deal with complex situations.
2. **Advanced Beginner** – Possesses a foundational understanding of key aspects and can perform simple, straightforward tasks independently, though they still require guidance when handling broader or more complex responsibilities.
3. **Competent** – Demonstrates solid working knowledge and can complete most tasks using their own judgment, showing the ability to cope with complexity through deliberate analysis and planning, although refinement may still be lacking.
4. **Proficient** – Develops a deep understanding of their discipline, consistently meets acceptable standards, and takes full responsibility for their work. They are capable of addressing complex challenges holistically and making confident decisions, while also seeing how individual actions align with broader goals.
5. **Expert** – Represents the highest level of mastery, combining authoritative knowledge with tacit understanding across practice areas. Experts achieve excellence with ease, go beyond established standards, and can approach complex situations both intuitively and analytically, envisioning innovative solutions and possibilities.

In the context of the Volunteer Management System (VMS), the five-level skill scale serves two important purposes. For organisers, it provides a way to define the required level of skill importance for each event, ensuring that the right expertise is allocated where it is most needed. For volunteers, the scale allows them to rate their own proficiency in different skills, creating a clear profile of their strengths and areas for growth. This dual approach enables more accurate matching between event requirements and volunteer capabilities, ensuring tasks are handled effectively while also giving volunteers opportunities to apply and develop their skills. Ultimately, it enhances both the efficiency of event management and the overall volunteer experience.

#### 4.2.1.2 Skill Selection and Proficiency Levels

Since the scope of a volunteer management system can be extremely broad, it is not practical to include every possible skill that volunteers may possess. To narrow the focus and provide a concrete demonstration example for this project, the system is designed around the context of an animal welfare volunteer management system (VMS). The animal welfare sector is chosen because it represents a common area where many nonprofit organisations operate. However, it

is important to note that the skills defined in this project are not fixed; they can be customised depending on the domain in which the VMS is deployed. For instance, an animal welfare VMS may include skills such as empathy and animal handling, while a disaster relief VMS may focus on skills like first aid, logistics, and crisis communication. This flexibility ensures that the system can be adapted for use by different types of organisations according to their operational needs.

According to CharityJob [88], several core skills are essential for animal welfare volunteers, as the nature of this work often involves dealing with vulnerable animals and sensitive environments. The first skill is **empathy**, which refers to understanding and responding to the emotional state of animals that may have suffered abuse, neglect, or trauma. Empathy helps volunteers approach animals with compassion, ensuring they feel safe and supported. Even for roles not involving direct contact with animals, such as fundraising or campaigning, empathy remains critical as it strengthens appeals for donations and public support. The second skill is **patience and sensitivity**, which are required when interacting with animals that may initially react with fear or aggression. Building trust often takes time, and volunteers must remain calm, gentle, and consistent to allow animals to adapt and feel comfortable in their care. The third skill is **animal handling knowledge**, which involves the ability to interpret animal body language and apply safe handling techniques tailored to different species and situations. Some of these skills may be formally taught, while others are gained through practice and experience in shelters. The fourth skill is **safeguarding**, which is vital because many animals in welfare organisations have been mistreated. Volunteers must be able to notice when an animal shows signs of mistreatment and respond in a way that protects the animal's wellbeing. At the same time, they should be emotionally prepared to handle these difficult situations, as working with vulnerable animals can often be upsetting. Finally, **communication** is also considered a key skill, as volunteers must work effectively with colleagues, shelter staff, and the public to achieve organisational goals. This often involves sharing important information about animals, such as their behaviour, needs, or progress, so that everyone can provide the best care. Good communication includes active listening, teamwork, and maintaining professionalism, even under pressure.

After reviewing the definition of skill levels, five levels of proficiency were identified, ranging from Novice (1) to Expert (5). Each level represents a different degree of expertise, clarifying

not only what stage of learning an individual is at, but also how they progress in terms of knowledge, judgment, and responsibility. Within the context of this project, these levels framework are applied to the selected animal welfare skills to show how volunteers may develop over time. This mapping allows organisers to better understand the varying capabilities of volunteers, while also enabling volunteers to reflect on their own growth and proficiency. The following outlines how each of the five selected skills can be defined across different levels of expertise:

**1. Empathy**

- **Novice (1):** Recognises that animals have feelings but relies on rules or guidance from others.
- **Advanced Beginner (2):** Notices signs of distress or fear but responds step-by-step without seeing the bigger picture.
- **Competent (3):** Reads emotional states with reasonable accuracy, adapts approach, and balances short- and long-term animal welfare goals.
- **Proficient (4):** Consistently empathetic across diverse situations, anticipates responses, and guides others in empathetic care.
- **Expert (5):** Develops a deep intuitive connection with animals, senses subtle behavioural cues, and creates new approaches grounded in empathy.

**2. Patience and Sensitivity**

- **Novice (1):** Wants immediate compliance, struggles with waiting, and may become frustrated.
- **Advanced Beginner (2):** Understands patience is needed but only manages it in straightforward cases.
- **Competent (3):** Manages frustration, applies calming techniques, and steadily builds trust.
- **Proficient (4):** Adapts smoothly to individual animal needs, rarely unsettled by setbacks.
- **Expert (5):** Models patience and sensitivity for others, designs stress-free care protocols.

**3. Animal Handling Knowledge**

- **Novice (1):** Minimal knowledge, follows instructions rigidly (e.g., lifting a cat/dog).
- **Advanced Beginner (2):** Handles animals safely in simple, low-stress situations.
- **Competent (3):** Reads body language and adjusts approach across different conditions.

- **Proficient (4):** Confident in complex or high-stress environments.
- **Expert (5):** Authoritative across species, develops safe handling techniques, trains staff/volunteers.

**4. Safeguarding**

- **Novice (1):** Aware safeguarding matters but unsure what signs to look for; relies on explicit guidance.
- **Advanced Beginner (2):** Recognises some indicators of mistreatment but misses subtle ones.
- **Competent (3):** Identifies common safeguarding issues and responds appropriately.
- **Proficient (4):** Anticipates risks, makes confident protective decisions, trains others.
- **Expert (5):** Contributes to safeguarding policy, develops frameworks, influences sector standards.

**5. Communication**

- **Novice (1):** Basic ability to share information, struggles with active listening or teamwork.
- **Advanced Beginner (2):** Communicates well in simple tasks but falters under stress.
- **Competent (3):** Clear communicator in most situations, adapts messages for team or public, resolves misunderstandings.
- **Proficient (4):** Skilled communicator even in crises, coordinates teams, mentors others.
- **Expert (5):** Inspires through communication, advocates publicly, develops strategies for engagement.

By focusing on these five skills, the animal welfare VMS demonstrates how the system can categorise, manage, and match volunteer expertise with organisational needs. The inclusion of proficiency levels further strengthens this process by allowing organisers to assign roles based not only on the presence of a skill but also on the depth of expertise a volunteer has attained. For example, if an event involves handling animals in a moderately stressful environment, organisers may define competent as the minimum required level for animal handling. This ensures that volunteers selected for the task have at least the ability to read animal behaviour, adjust their approach, and maintain safety. By setting clear expectations in this way, the system ensures that events are staffed with volunteers who meet the necessary skill levels, while also giving volunteers a clear understanding of their own progress and areas for further development.

#### 4.2.1.3 Calculation of Weighted Skill Matching Score

Volunteer-event matching in many existing systems often relies on simple keyword matching which do not account for the varying proficiency levels of volunteers. This can result in mismatches, such as assigning a beginner to tasks requiring advanced expertise, or overlooking volunteers who could contribute more effectively in roles that align with their skill level. To address this problem, a weighted skill matching algorithm is used in this project, as it provides a more accurate and fair method of evaluating how well a volunteer's skills align with the requirements of an event.

The process begins with the organiser defining the skills required for an event and specifying the minimum proficiency level for each skill, based on the definitions outlined earlier. On the other hand, volunteers have to self-assess and rate their proficiency levels against the same scale (Novice to Expert). This dual input creates the foundation for calculating a matching score.

The algorithm works by first calculating the highest possible skill score for the event. For each required skill, the maximum possible score is obtained by multiplying the maximum proficiency level (5) by the required level defined for that skill. These values are then summed across all required skills to give the maximum possible skill score. Next, the actual skill score for a volunteer is calculated in the same way, by summing the multiplication of the volunteer's self-rated proficiency level and the required level for each matching skill. Finally, the skill matching score is then expressed as a percentage by dividing the actual skill score by the maximum possible skill score:

$$Skill\ Matching\ Score\ (\%) = (Actual\ Skill\ Score)/(Maximum\ Possible\ Skill\ Score) \times 100\%$$

**Example Scenario:**

Suppose an animal adoption event requires three skills:

Table 4-1: Skills level required for animal adoption event.

| Skill | Required Skill Level | Description |
|---|---|---|
| Empathy | 5 (Expert) | Develops a deep intuitive connection with animals, senses subtle behavioural cues, and creates new approaches grounded in empathy. |
| Animal Handling Knowledge | 4 (Proficient) | Confident in complex or high-stress environments. |
| Communication | 2 (Advanced Beginner) | Communicates well in simple tasks but falters under stress. |

**Maximum Possible Skill Score** = (5×5) + (5×4) + (5×2) = 55.

Suppose a volunteer has skill proficiency as below:

Table 4-2: Skill proficiency of volunteer.

| Skill | Skill Proficiency | Description |
|---|---|---|
| Empathy | 5 (Expert) | Develops a deep intuitive connection with animals, senses subtle behavioural cues, and creates new approaches grounded in empathy. |
| Animal Handling Knowledge | 3 (Competent) | Reads body language and adjusts approach across different conditions. |
| Safeguarding | 2 (Advanced Beginner) | Recognises some indicators of mistreatment but misses subtle ones. |
| Communication | 3 (Competent) | Clear communicator in most situations, adapts messages for team or public, resolves misunderstandings. |

**Actual Skill Score** = (5×5) + (3×4) + (3×2) = 43.

**Skill Matching Score** = 43/55 × 100% = 78.18%.

This percentage provides organisers with a clear indication of how well a volunteer's skills align with the event's requirements.

This formula works effectively because it balances both breadth (covering all required skills) and depth (reflecting the volunteer's proficiency level within each skill). By setting required skill levels, organisers can define the baseline expectations for participation, while volunteers receive a fair and transparent evaluation of how closely their skills proficiency match event needs. As a result, the algorithm not only supports better event outcomes by identifying the most suitable volunteers but also motivates individuals to develop their skills further over time.

### 4.2.2 Distance-Based Volunteer-Event Matching Algorithm

To enhance volunteer participation and reduce dropouts caused by long commutes, the system incorporates geographical proximity as a key factor in matching volunteers with events. The algorithm calculates the road distance between the volunteer's address and the event location using the Google Maps Distance Matrix API. The request specifies the volunteer's address as the origin, the event location as the destination, and the default travel mode as driving. The API returns the distance in metres, which is then converted to kilometres for further calculations. Based on this distance, a proximity score is computed using an exponential decay function:

$$Proximity\ Score\ (\%) = e\text{^}(-0.025\ \times distance\ in\ km) \times 100\%$$

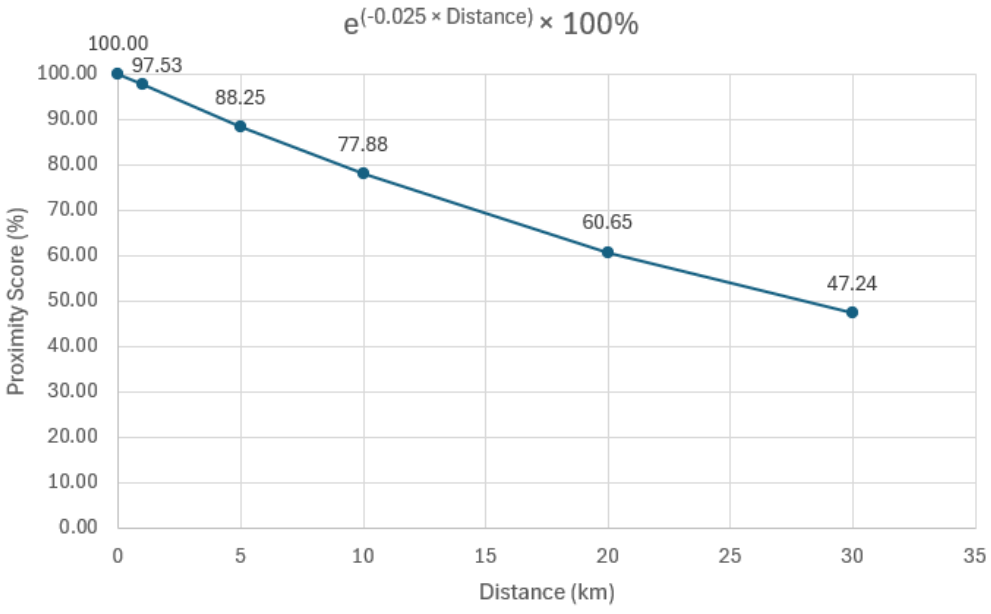



Figure 4-2: Graph of proximity score function.

This exponential decay function assigns higher scores to nearby events while gradually reducing the score for events located further away. For instance, an event 1 km from a volunteer receives a substantially higher score than one 10 km away, yet the decay remains smooth enough to still consider moderately distant events. The plotted graph of the proximity scores above clearly illustrates this behaviour: it starts at 100% for 0 km, drops sharply for the first few kilometres, and then gradually flattens as distance increases. For example, the proximity score decreases from 97.53% at 1 km to 77.88% at 10 km, and further to 47.24% at 30 km.

This function works effectively because the exponential decay naturally prioritises closer events while still allowing more distant events to retain some score. The rapid drop at short distances ensures that nearby events are strongly favoured, while the gradual flattening at longer distances prevents distant events from being completely excluded, providing a balanced and intuitive approach to volunteer-event matching. Moreover, the system considers actual driving distances rather than straight-line distances by using the Google Maps Distance Matrix API, further improving the accuracy of the proximity score. Overall, the distance-based matching algorithm not only prioritises events that are conveniently located for volunteers, reduce travel-related dropouts, but also ensures a fair and balanced recommendation of opportunities across varying distances.

### 4.2.3 Combination of both algorithms

To provide a comprehensive and balanced event recommendation system, the Volunteer Management System combines both the skill matching score and the proximity score into a single final matching score. This approach ensures that volunteers are recommended to events not only based on how well their skills align with the event requirements but also on how conveniently the event is located.

The final matching score is calculated using a weighted average of skill matching score and proximity score:

$$Final\ Matching\ Score\ (\%) = (Skill\ Matching\ Score\ \times 0.5) + (Proximity\ Score\ \times 0.5)$$

For instance, a volunteer has a skill matching score of 78.18% for animal adoption event from previous chapter and assume he is located 10km away from the event with a proximity score of 77.88%. Therefore, the final matching score will be (78.18 × 0.5) + (77.88 × 0.5) = 78.03%. This score reflects that the volunteer has a good match for the event in terms of skills and is reasonably close in distance, making them a strong candidate for participation.

By assigning equal weights to skill compatibility and geographical proximity, the system treats both factors as equally important in the matching process. A higher final score indicates that the volunteer is both highly suitable for the event in terms of skills and located close enough to reduce travel-related barriers. As a result, this combined approach helps organisers identify the most appropriate volunteers efficiently while also increasing the likelihood of volunteer participation and engagement.

# Chapter 5
# System Implementation

## 5.1 Hardware Setup

The primary hardware used in this project is a personal laptop computer. The laptop serves as the central development environment where all coding, testing, and database management tasks are performed. It is used to run the necessary software such as Visual Studio Code, Node.js, and MySQL Workbench for developing and managing the web-based Volunteer Management System (VMS). The laptop also functions as both the frontend and backend development platform, allowing the system to be developed, tested, and debugged locally. As this project is primarily web-based, no additional physical hardware is required.

Table 5-1 Specifications of laptop.

| Description | Specifications |
|---|---|
| Model | MSI Sword 16 HX B14VEKG |
| Processor | Intel(R) Core (TM) i5-14450HX |
| Operating System | Windows 11 |
| Graphic | NVIDIA GeForce RTX 4050 Laptop GPU |
| Memory | 16GB DDR4 RAM |
| Storage | 512GB SSD |

## 5.2 Software Setup

There are three software needed to be installed and downloaded in my laptop in order to start developing the web-based Volunteer Management System (VMS). They are Visual Studio Code, Node.js, and MySQL Workbench.

Visual Studio Code (VS Code) is a lightweight yet powerful source-code editor developed by Microsoft. It supports multiple programming languages and is widely used for web development. In this project, VS Code serves as the main Integrated Development Environment (IDE) for writing and maintaining both frontend and backend code [89]. It provides helpful features such as syntax highlighting, code suggestions, debugging tools, and an integrated terminal.

Node.js is an open-source, cross-platform runtime environment that allows JavaScript to be run on the server side. It enables developers to use JavaScript for backend development, making it easier to build scalable and fast web applications. Besides, Node.js also comes with npm (Node Package Manager), which is used to manage the dependencies and packages required for the project [90].

MySQL Workbench is a graphical user interface software developed by Oracle that provides an integrated environment for working with MySQL databases. It enables developers to design, model, manage, and interact with databases visually [91]. It streamlines tasks like writing and executing SQL queries, designing database schemas, and monitoring server performance. By offering a user-friendly interface, MySQL Workbench plays a vital role in ensuring efficient data storage, retrieval, and overall database management throughout the development process.

## 5.3 Setting and Configuration

The development of the web-based Volunteer Management System (VMS) requires a well-organised project setup, proper configuration, and efficient database design to ensure smooth development, maintenance, and deployment. The following subchapters explain the details of packages and extensions used, project structure, and database configuration.

### 5.3.1 Project Setup and Development Packages

The development of the Volunteer Management System (VMS) involves setting up a React project as the core frontend framework while incorporating several packages and extensions to streamline development and enhance functionality. The React project was initialised using the command *npx create-react-app fyp1*, which generated the default project structure including the src folder for all frontend code, a public folder for static assets, and configuration files such as package.json. To enhance development productivity and add functionality, several packages and extensions were installed. For example, packages like Tailwind CSS, lucide-react, axios, react-router-dom, and react-toastify are installed within the React project via the terminal using npm, providing styling, icons, HTTP request handling, routing, and notification features respectively. Dotenv is also included to manage environment variables securely. In addition, Prettier, a code formatting extension is installed in Visual Studio Code, ensuring consistent and

readable code throughout the project. The combined use of these packages and extensions enables modular, maintainable, and efficient frontend development.

### 5.3.2 Project Structure and Code Organisation

The project follows a structured and modular approach to code organisation. On the frontend, all code resides within the src folder, which is divided into two main subfolders: components and pages. The components folder primarily contains modules that implement specific functionalities, such as event creation, event modification, event listing, event participation, attendance management, volunteer history, leaderboard, and profile management. It also includes general UI components like the navigation bar and footer, which are shared across login and signup pages. On the other hand, the pages folder contains full-page views corresponding to specific routes, including the home page, login page, signup page, organiser dashboard page, and volunteer dashboard page.

On the backend, a minimal architecture is adopted. The server.js file serves as the entry point for the Node.js server, responsible for initialising the server, defining API routes, handling client requests, and managing database connections. For handling file uploads, such as volunteer and organiser profile pictures, an uploads folder is included in the project. Files are stored using a configured storage system, where uploaded images are saved to /uploads/profile_pictures with unique filenames generated based on the user type and timestamp. In order to ensure secure and flexible configuration across different environments, sensitive values such as the database host, username, password, and database name are stored in an .env file and accessed using the dotenv package. This project structure promotes maintainability, modularity, and easier debugging throughout development.

### 5.3.3 Database Configuration

The project uses MySQL as its database management system to efficiently store and manage system data. The database consists of several key tables, each serving a specific purpose to support the functionality of the system. These tables include Volunteer, Organiser, Events, Application, Participation, Skill Category, Volunteer Skills, Event Skills, and Score.

The Volunteer and Organiser tables store personal information, login credentials, and profile details for volunteers and organisers respectively. The Events table contains event-related

information such as the title, date, time, location, description, and the maximum number of volunteers allowed. The Application table records volunteer applications for events, including their application status, motivation messages, and the timestamp of application submission. The Participation table tracks volunteer participation for each event, including attendance status and hours volunteered. To manage skills, the Skill Category table defines different skill types and proficiency levels, while the Volunteer Skills table links volunteers to their skills and proficiency level. Similarly, the Event Skills table specifies which skills are required for each event and their importance, enabling effective skill matching between volunteers and events. The Score table records the calculated skill matching score for volunteers in relation to specific events.

In addition, relationships between tables are enforced through foreign key constraints. For example, events are linked to organisers, applications are linked to volunteers and events, and participation records are associated with both volunteers and events. This structure ensures data integrity, supports efficient queries, and allows the system to effectively manage volunteer histories, event participation, skill matching, and attendance tracking, thereby enhancing the overall reliability and functionality of the VMS.

### 5.4 Implementation Issues and Challenges

During the development of the web-based Volunteer Management System (VMS), several implementation issues and challenges were encountered that required careful consideration and problem-solving. One of the primary challenges was ensuring proper integration between the frontend and backend components. Since the frontend was developed using React and the backend relied on Node.js with a MySQL database, establishing seamless communication through API endpoints demanded careful planning, testing, and debugging. Common issues included data fetching errors, inconsistent or unexpected data formats returned from the database, and asynchronous request handling problems. Resolving these issues required careful design of API responses, validation of data formats, and proper error handling to ensure smooth and reliable data transfer between the client and server.

Another significant challenge is the process of designing and implementing the weighted skill-based algorithm and the distance-based algorithm to match volunteers to events effectively. The design process began with analysing the requirements for accurate volunteer-event

matching, including the types of skills, proficiency levels, and geographical constraints. For the weighted skill-based algorithm, careful consideration was given to how volunteer proficiency levels would be mapped to the skills required by each event, and how each skill's importance would influence the overall matching score. On the other hand, the distance-based algorithm required careful consideration in designing a formula to calculate the proximity score between volunteers and event locations, ensuring that both near and far events were appropriately weighted. Throughout the process, iterative testing and refinement were necessary to ensure that the algorithms produced correct, real-time results whenever volunteer skills or event requirements were updated. Balancing accuracy, efficiency, and maintainability in these calculations demanded iterative development and testing.

Lastly, time management, testing, and debugging were ongoing challenges throughout the project. Since the system comprised multiple modules with different features, careful planning was required to coordinate their development and ensure they worked together seamlessly. Thus, significant time and effort were devoted to testing and resolving issues that arose during development. This included addressing problems with API integration, state management in React, asynchronous operations, and database queries. Despite these challenges, the implementation process provided valuable learning opportunities and resulted in a robust, functional, and maintainable Volunteer Management System

# Chapter 6

# System Evaluation and Discussion

### 6.1 System Testing and Use Case Validation

The system testing of the Volunteer Management System (VMS) focuses on evaluating whether each module functions as intended and ensuring the overall reliability of the system. The testing approach follows a use case-based methodology, where each module is tested according to its core functionality, expected inputs, and anticipated outputs. This ensures that both functional and non-functional aspects of the system are examined. The use case testing for each module is summarised as follows:

Table 6-1: Use case testing for each module.

| Module | Use Case (Input) | Expected Output |
|---|---|---|
| Login/Registration | Volunteer and organiser registration with correct input format | Registration successful and save user data into database |
| | Volunteer and organiser registration with wrong input format | Registration failed |
| | Volunteer and organiser login with correct credentials | Login successful and redirect to respective user dashboard |
| | Volunteer and organiser login with wrong credentials | Login failed |
| Event Creation | Organiser navigates to event creation module | Display event creation form |
| | Event creation with correct input format | Event creation successful and store event details in database |
| | Event creation with wrong input format | Event creation failed |
| Event Listing | Volunteer navigates to event listing module | Display available events detail from database |
| | Volunteer click apply event button | Display application confirmation modal |
| | Volunteer confirm application | Save application details in database |
| Event Modification | Organiser navigates to event modification module | Display created events detail from database |
| | Organiser click edit event button | Display event editing form |
| | Event editing with correct input validation | Event editing successful |
| | Event editing with wrong input validation | Event editing failed |
| | Organiser click delete event button | Display event delete confirmation modal |
| | Organiser confirm delete event | Delete selected event from database |

| | | |
|---|---|---|
| | Organiser click view applications button | Display correct application details from database |
| | Organiser click approve application button | Update approved status in database |
| | Organiser click reject application button | Update rejected status in database |
| Event Participation | Volunteer navigates to event participation module | Display correct application details from database and corresponding action button |
| | Volunteer click cancel application button for pending application | Display cancel application confirmation modal |
| | Volunteer confirm cancel application | Delete selected application from database |
| | Volunteer click leave event button for approved application | Display leave event confirmation modal |
| | Volunteer confirm leave event | Event successfully leaved and delete application and participation record in database |
| Attendance | Organiser navigates to attendance module | Display past created events from database for marking attendance |
| | Organiser search event by using event name filter | Display events that match the event name filter |
| | Organiser filter events by using time period filter | Display events that match the selected time period |
| | Organiser click mark attendance button | Display volunteer attendance status from database |
| | Organiser click mark all present button | Automatically mark all volunteer present |
| | Organiser click mark all absent button | Automatically mark all volunteer absent |
| | Organiser click mark reset all button | Automatically reset all volunteer attendance status |
| | Organiser click tick button for particular volunteer | Mark the volunteer present |
| | Organiser click cross button for particular volunteer | Mark the volunteer absent |
| | Organiser click save attendance button | Save attendance status in database and log volunteered hours |
| History | Volunteer navigates to history module | Display participated event histories and attendance status from database |
| | Volunteer filter event histories by using attendance status filter | Display event histories that match the attentance status |
| | Volunteer filter event histories by using time period filter | Display event histories that match the time period |
| Profile | Volunteer or organiser navigate to profile module | Display profile statistics and information from database |
| | Volunteer or organiser click edit profile button | Profile information becomes editable |
| | Volunteer or organiser click upload profile picture button | Open file explorer for selecting profile picture |

| | | |
|---|---|---|
| | Volunteer or organiser edit their profiles and click save button | Update the profile information in database |
| Leaderboard | Volunteer or organiser navigate to leaderboard module | Display volunteer leaderboard from database |
| | Volunteer or organiser filter leaderboard by using time period filter | Display leaderboard that match the time period |
| | Volunteer or organiser use sort filter to sort leaderboard by hours or events | Display leaderboard that match the sort filter |
| Event Matching Algorithm | Organiser defines skills importance level and volunteer self-rate their skill proficiency | Calculate skill matching score accurately using weighted-based skill matching algorithm |
| | Organiser defines event location and volunteer update their address | Calculate proximity score accurately using distance-based skill matching algorithm |
| | Combination of both algorithms | Calculate final matching score accurately |

## 6.2 Testing Setup and Result

### 6.2.1 Outcome of Login/Registration Module Use Case

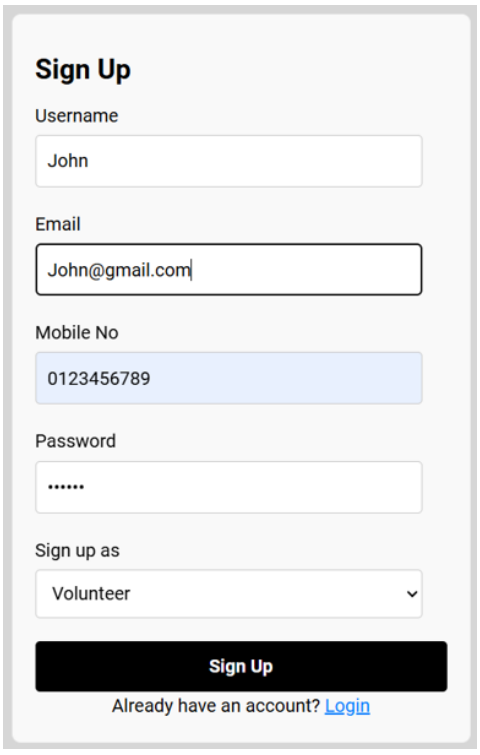



Figure 6-1: Volunteer registration with correct input format.

| volunteer_id | username | email | mobile | password | bio | address | profile_picture |
|---|---|---|---|---|---|---|---|
| 1 | John | John@gmail.com | 0123456789 | $2b$05$zTowBWEnqNA3ua9U9Aoy.uZRuwuOj... | This volunteer hasn't added a bio. | - | NULL |
| 2 | David | David@gmail.com | 0123456788 | $2b$05$8s2RJFJRWTMQj.VdcPgMdOYbcquFjcV... | This volunteer hasn't added a bio. | - | NULL |
| 3 | Emily | Emily@gmail.com | 0123456786 | $2b$05$j2eIW.Axel5/MKtWtSgHFeNcMn72uI28... | This volunteer hasn't added a bio. | - | NULL |

Figure 6-2: Actual output showing volunteer successfully registered and stored in database.

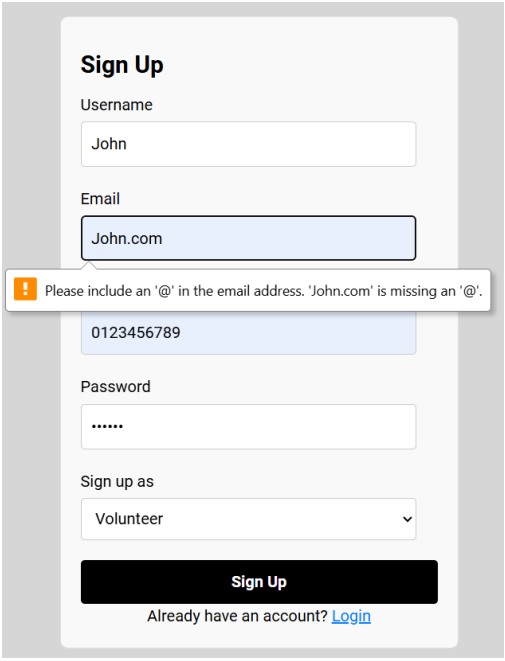



Figure 6-3: Wrong email format results in registration failed.

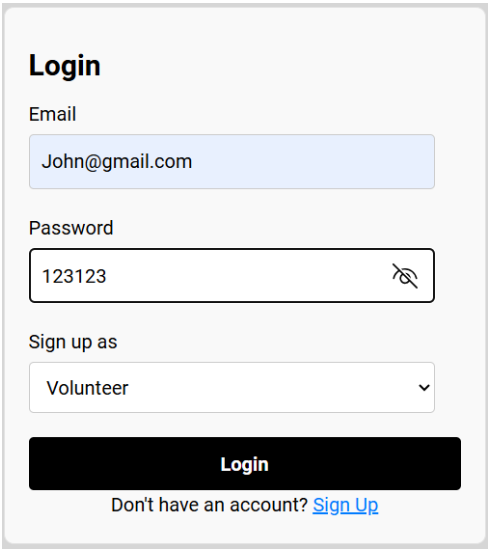



Figure 6-4: Volunteer login with correct credentials.

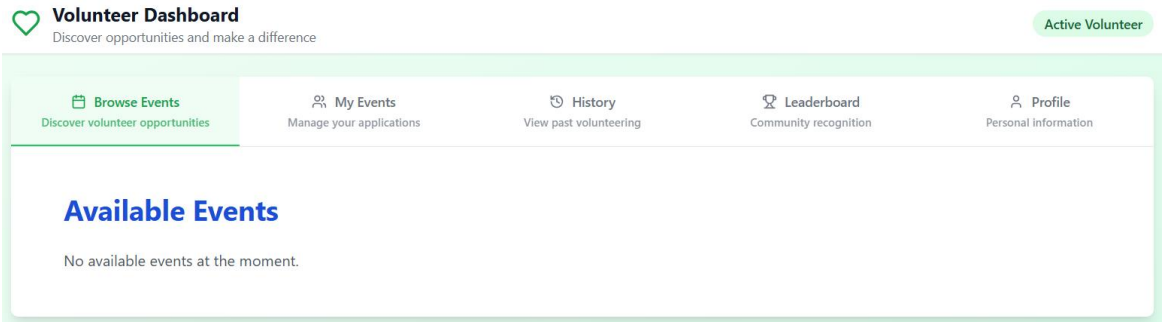



Figure 6-5: Actual output showing volunteer successfully login and redirected to volunteer dashboard.

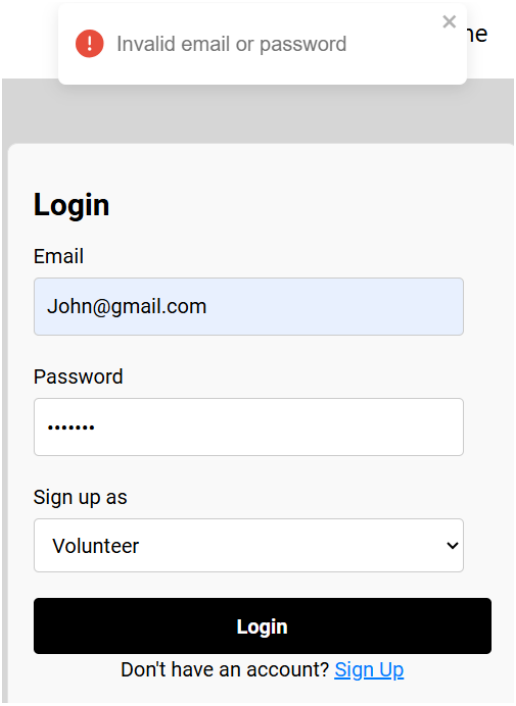



Figure 6-6: Volunteer login failed due to wrong credentials.

The outcome of the four use cases of Login/Registration Module have been shown in the figures above. We can see that when users input correct format of registration information, the registration process is successful and the information will be stored into either volunteer or organiser table based on the role selected. When incorrect format is entered, a reminder message will appear to indicate what is wrong and results in registration unsuccessful. On the other hand, when user login with correct credentials that align with data stored in database, they will be redirected to respective user dashboard based on their role. If wrong credentials are entered during login, a pop out message will appear showing login failed. Therefore, all the use cases of Login/Registration Module are valid as shown below:

Table 6-2: Outcome of Login/Registration Module use case.

| Module | Use Case (Input) | Expected Output | Outcome |
|---|---|---|---|
| Login/Registration | Volunteer and organiser registration with correct input format | Registration successful and save user data into database | ✓ |
| | Volunteer and organiser registration with wrong input format | Registration failed | ✓ |
| | Volunteer and organiser login with correct credentials | Login successful and redirect to volunteer dashboard | ✓ |
| | Volunteer and organiser login with wrong credentials | Login failed | ✓ |

### 6.2.2 Outcome of Event Creation Module Use Case

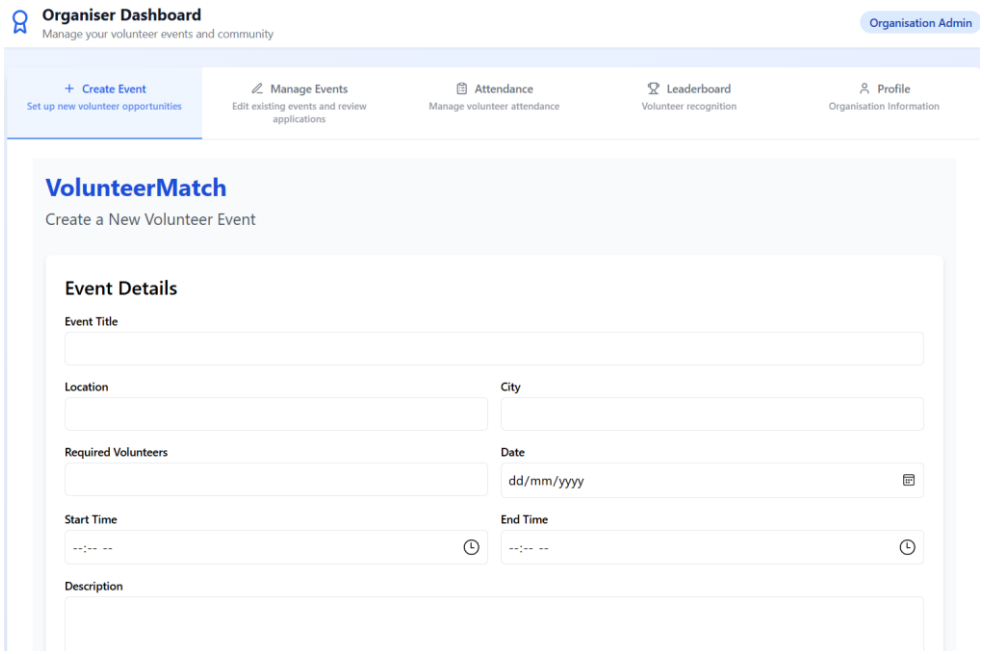



Figure 6-7: Actual output of event creation form displayed when organiser navigates to Event Creation Module.

| event_id | event_name | description | location | city | event_date | start_time | end_time | max_volunteers | current_volunteers | cre |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Stray Dog Vaccination Drive | A community event focused on vaccinating stra... | Ipoh Garden Animal Clinic | Ipoh | 2025-09-20 | 10:00:00 | 14:00:00 | 10 | 0 | 202 |
| 2 | Cat Adoption Day | An adoption fair showcasing rescued cats in ne... | Taman Boon Bak | Ipoh | 2025-09-26 | 16:00:00 | 19:00:00 | 10 | 0 | 202 |
| 3 | Community Pet Care Fair | A public event offering free basic health checks... | Klinik Haiwan Kampar | Kampar | 2025-09-28 | 08:00:00 | 12:00:00 | 5 | 0 | 202 |
| 4 | Wildlife Rescue Awareness Workshop | An educational session on how to safely report ... | YMCA Ipoh | Ipoh | 2025-09-15 | 18:00:00 | 20:00:00 | 6 | 0 | 202 |
| 5 | Shelter Clean-Up and Enrichment Day | Volunteers will join staff in cleaning animal shelt... | Ipoh Society for the Prevention of Cruelty to A... | Ipoh | 2025-10-02 | 10:00:00 | 13:00:00 | 6 | 0 | 202 |

Figure 6-8: Event creation successfully and event details stored in database if correct input format is entered during event creation.

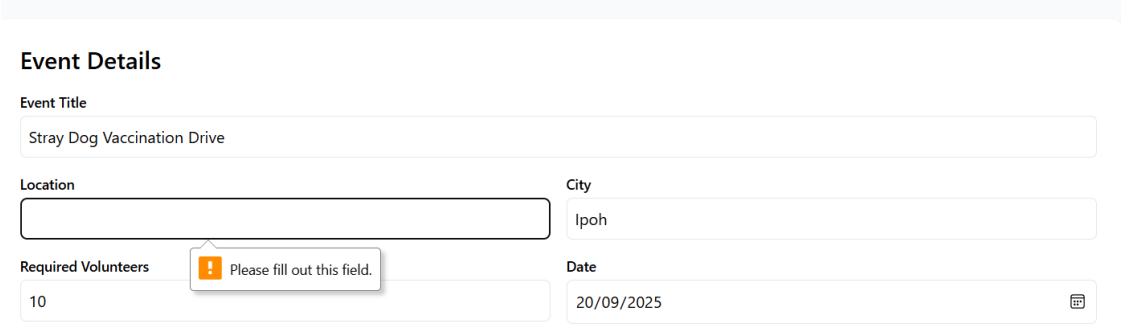



Figure 6-9: Event creation failed due to wrong input format such as empty field.

The outcome of the three use cases of Event Creation Module have been shown in the figures above. When organiser navigates to Event Creation Module, the event creation form will be displayed. If there is no wrong input format during event creation, the event will be successfully created and the information will be stored in the database. However, event creation will fail with an error message if there is wrong input format. Therefore, all the use cases of Event Creation Module are valid as shown below:

Table 6-3: Outcome of Event Creation Module use case.

| Module | Use Case (Input) | Expected Output | Outcome |
|---|---|---|---|
| Event Creation | Organiser navigates to event creation module | Display event creation form | ✓ |
| | Event creation with correct input format | Event creation successful and store event details in database | ✓ |
| | Event creation with wrong input format | Event creation failed | ✓ |

### 6.2.3 Outcome of Event Listing Module Use Case

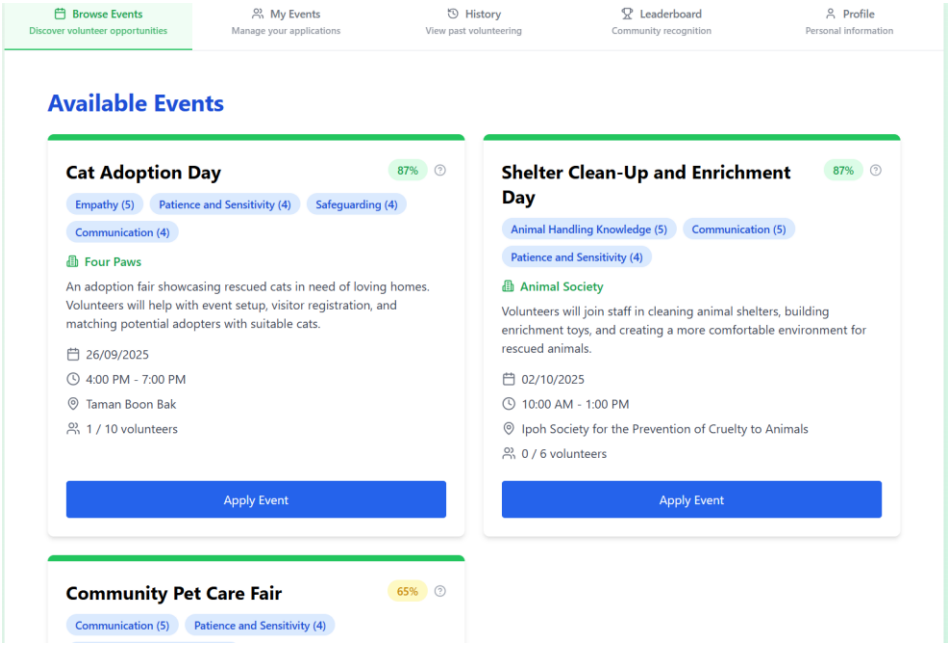



Figure 6-10: Available events displayed when volunteer navigates to Event Listing Module.

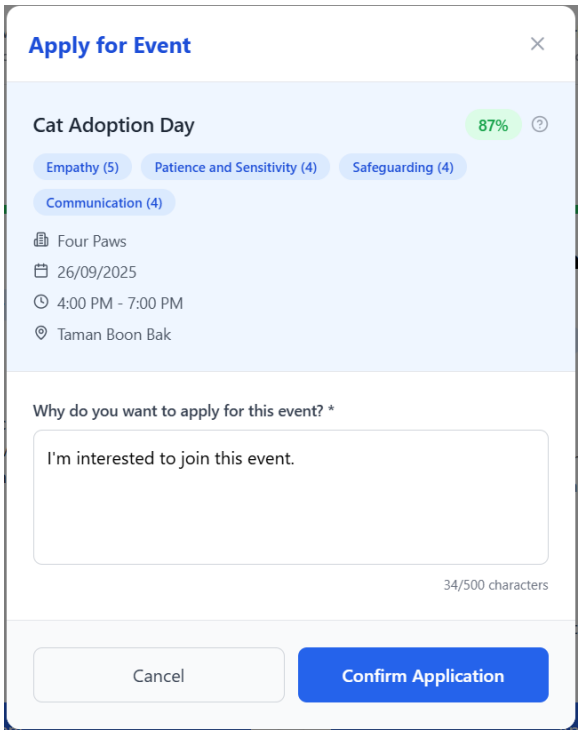



Figure 6-11: Application confirmation modal appears when volunteer click apply event button.

| application_id | volunteer_id | event_id | message | status | applied_at |
|---|---|---|---|---|---|
| 1 | 1 | 2 | I'm interested to join this event. | Pending | 2025-09-21 04:20:02 |

Figure 6-12: Application details successfully stored in database after confirmation.

The figures above show the outcome of the three use cases of Event Listing Module. When volunteer navigates to Event Listing Module, all the available events will be displayed along with the matching score. When volunteer click apply event button, the application confirmation modal has appeared. After confirming the application, the application details also successfully inserted into the database. Therefore, all the use cases of Event Listing Module are valid as shown below:

Table 6-4: Outcome of Event Listing Module use case.

| Module | Use Case (Input) | Expected Output | Outcome |
|---|---|---|---|
| Event Listing | Volunteer navigates to event listing module | Display available events detail from database | ✓ |
| | Volunteer click apply event button | Display application confirmation modal | ✓ |
| | Volunteer confirm application | Save application details in database | ✓ |

### 6.2.4 Outcome of Event Modification Module Use Case

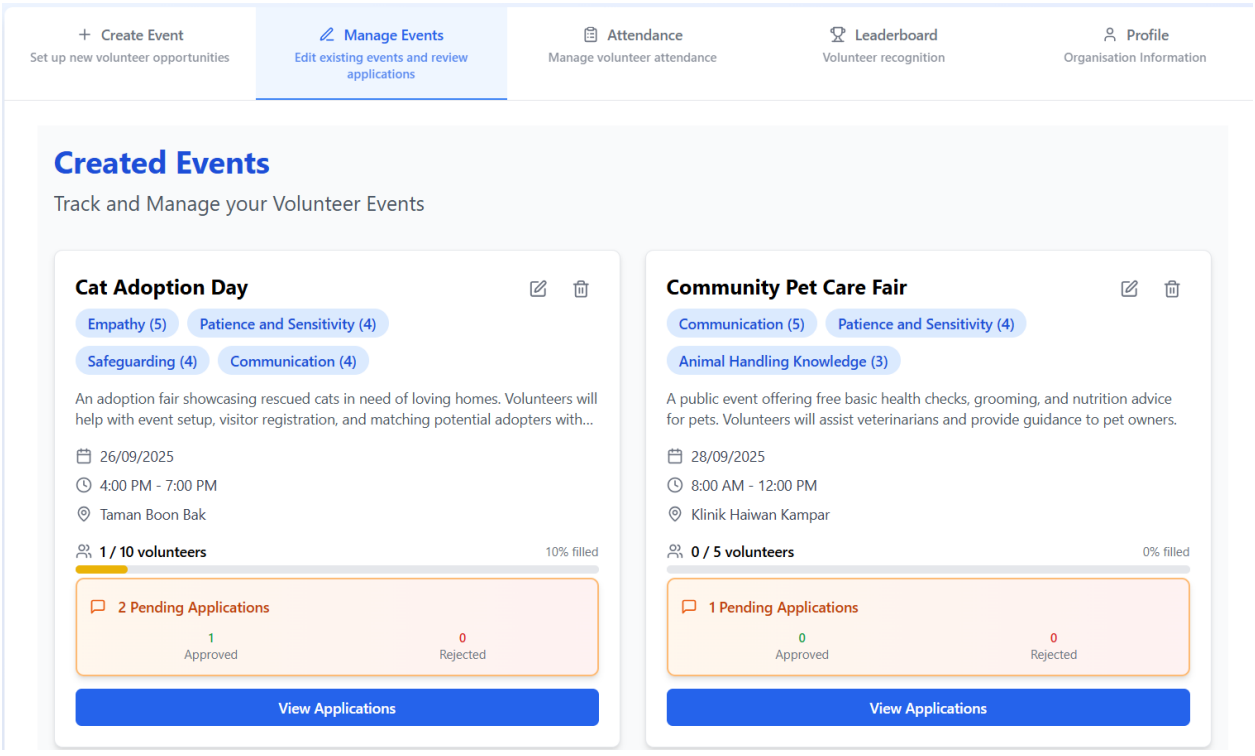



Figure 6-13: Created events displayed when organiser navigates to Event Modification Module.

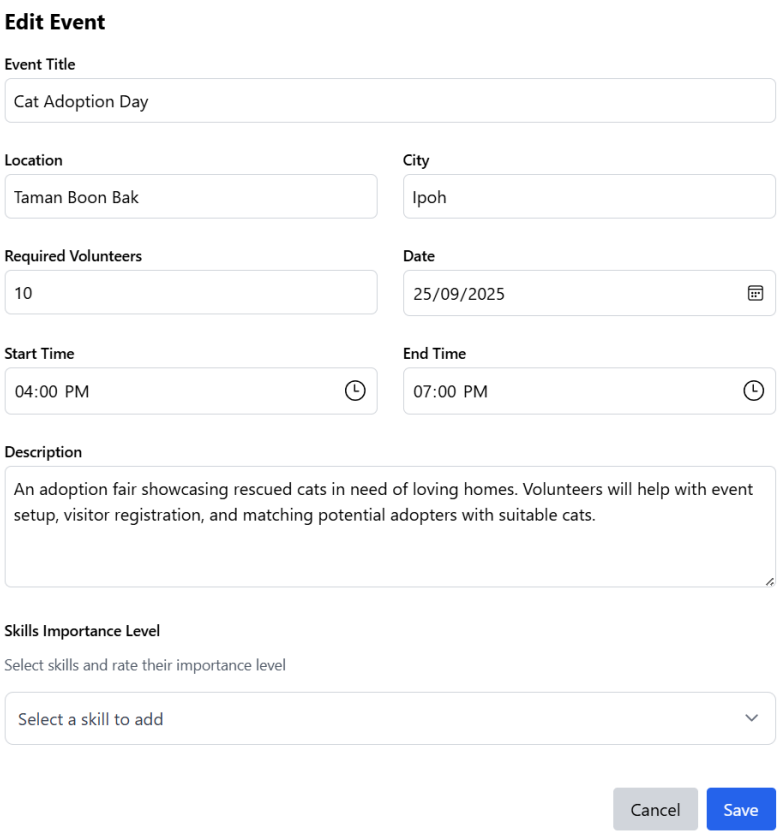



Figure 6-14: Event editing form displayed when organiser clicks edit event button.

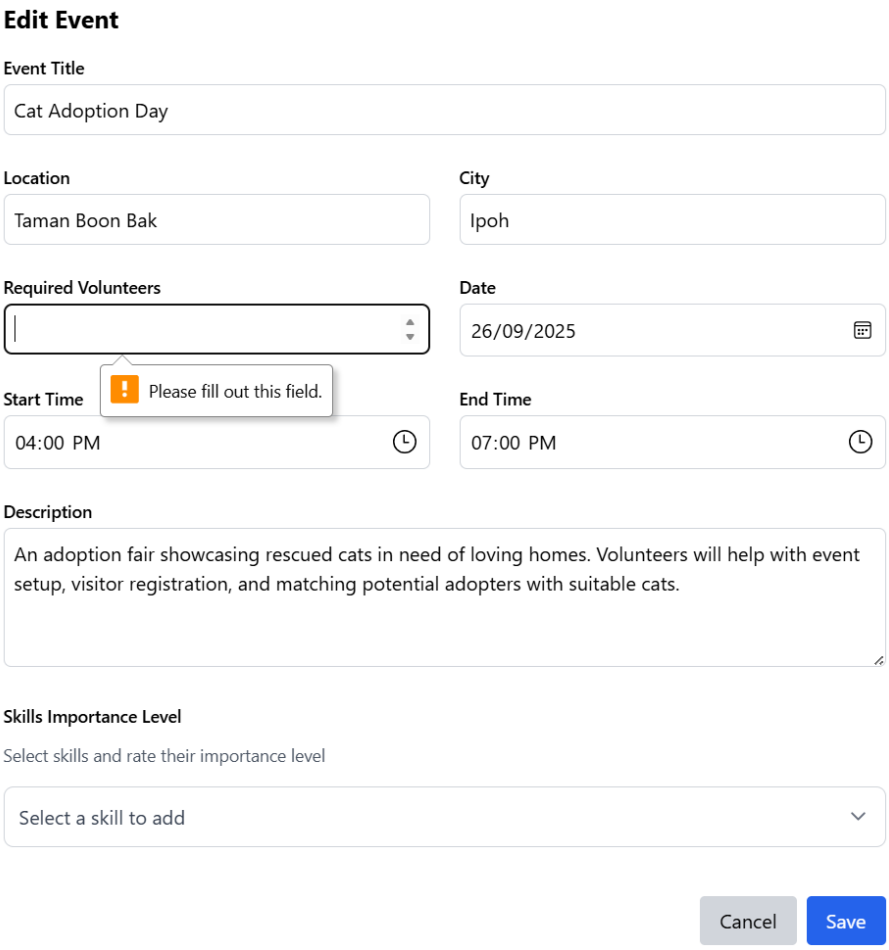



Figure 6-15: Event editing failed when wrong input format such as empty field.

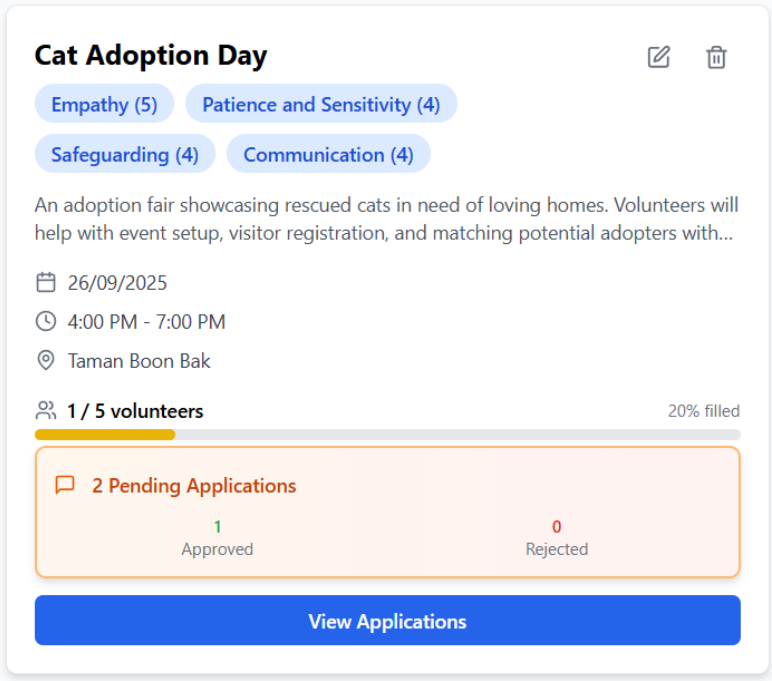



Figure 6-16: Event successfully edited (required volunteers and date) and updated in database.

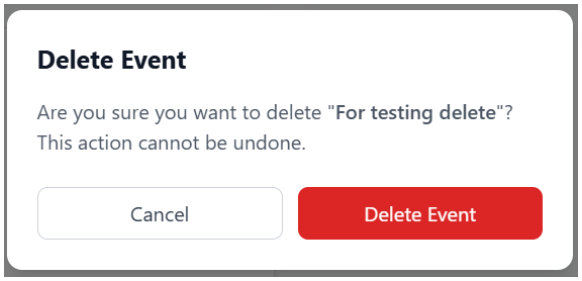



Figure 6-17: Event delete confirmation modal appeared when organiser clicks delete event button.

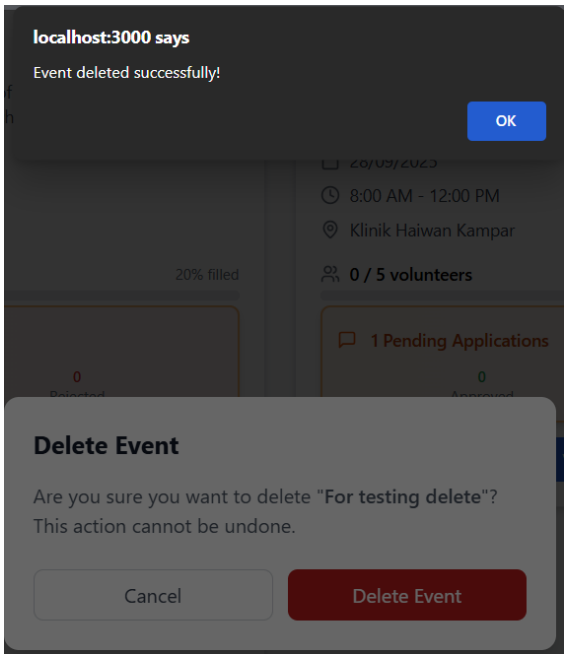



Figure 6-18: Event successfully deleted and remove from database once confirmed.

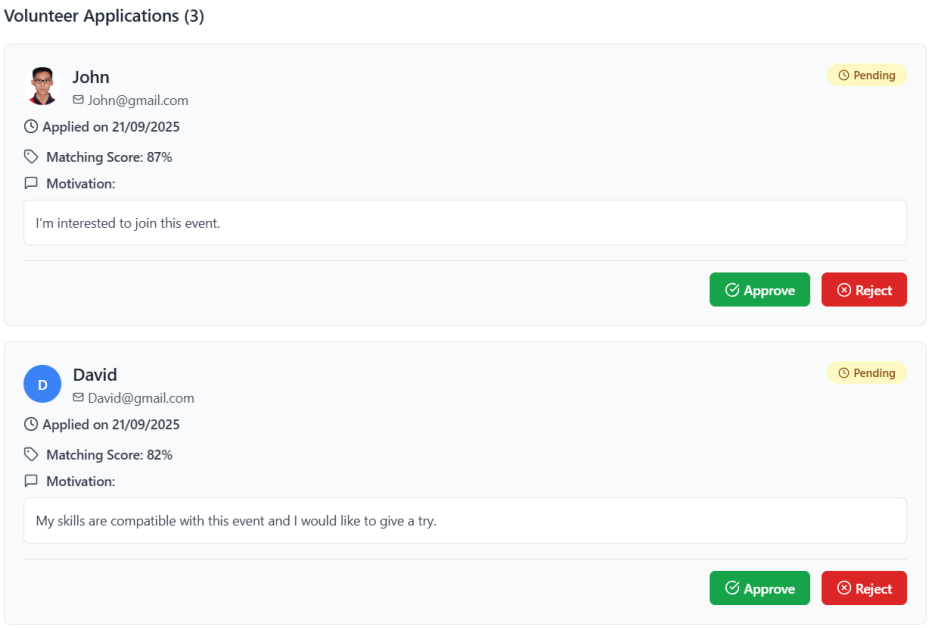



Figure 6-19: Volunteer applications displayed when organiser clicks view application.

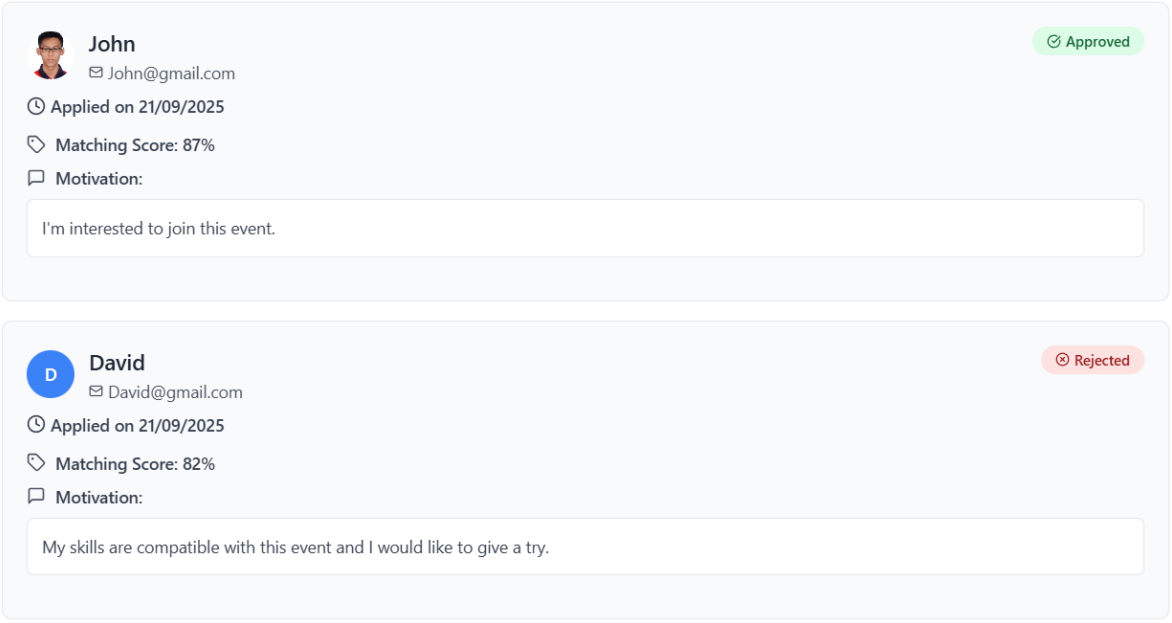



Figure 6-20: Application status successfully updated and stored in database.

The figures above show the outcome of the nine use cases of Event Modification Module. When volunteer navigates to Event Modification Module, created events will be displayed for event and application management. Organisers can update event details or delete event if any uncertainty arises. Furthermore, volunteer applications can be displayed accurately when view application button is clicked. Application approvement or rejection can also be done and updated in database without error. Therefore, all the use cases of Event Modification Module are valid as shown below:

Table 6-5: Outcome of Event Modification Module use case.

| Module | Use Case (Input) | Expected Output | Outcome |
|---|---|---|---|
| Event Modification | Organiser navigates to event modification module | Display created events detail from database | ✓ |
| | Organiser click edit event button | Display event editing form | ✓ |
| | Event editing with correct input validation | Event editing successful | ✓ |
| | Event editing with wrong input validation | Event editing failed | ✓ |
| | Organiser click delete event button | Display event delete confirmation modal | ✓ |
| | Organiser confirm delete event | Delete selected event from database | ✓ |
| | Organiser click view applications button | Display correct application details from database | ✓ |
| | Organiser click approve application button | Update approved status in database | ✓ |
| | Organiser click reject application button | Update rejected status in database | ✓ |

### 6.2.5 Outcome of Event Participation Module Use Case

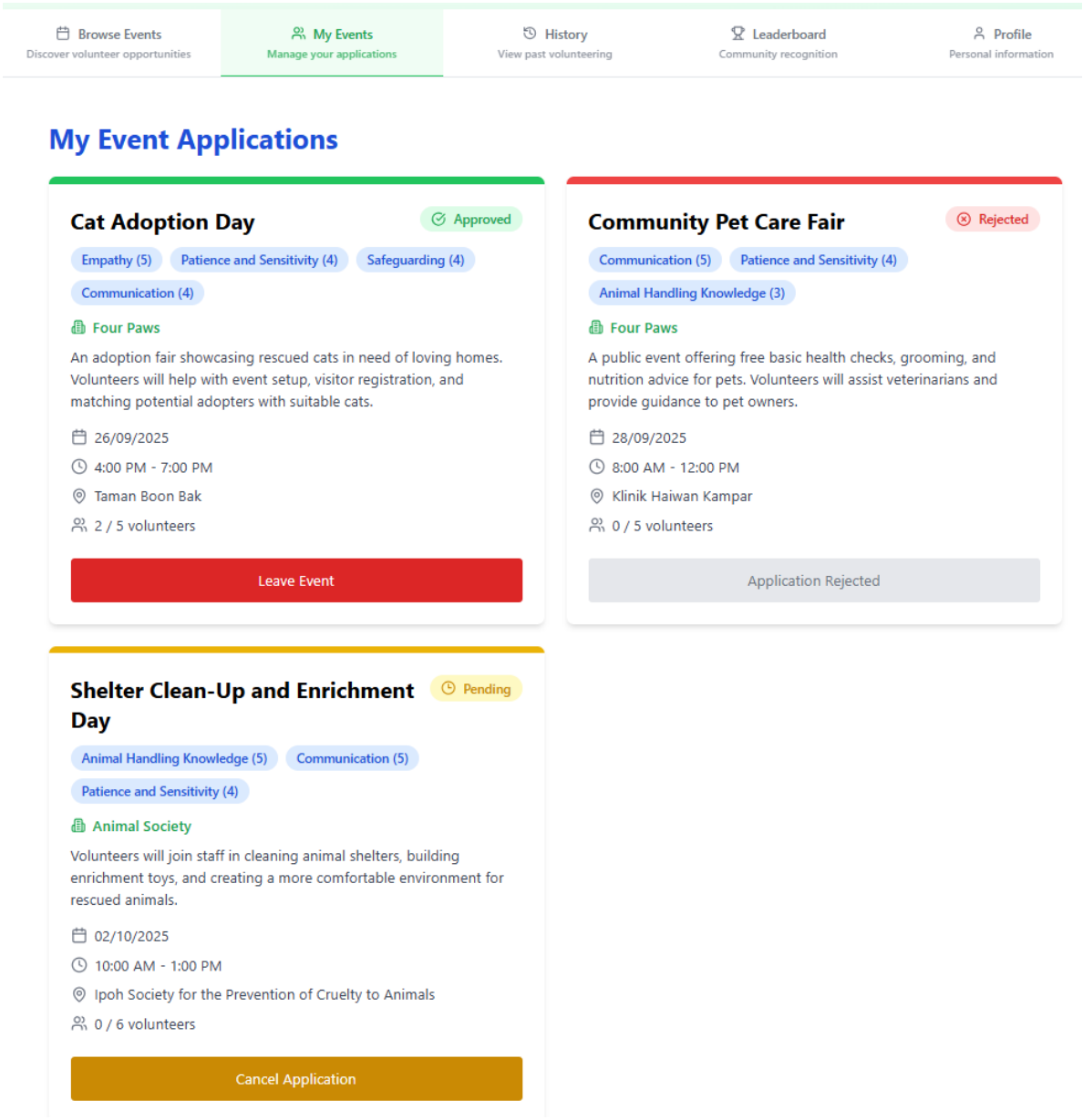



Figure 6-21: Application status and corresponding actions successfully displayed when volunteer navigates to Event Participation Module.

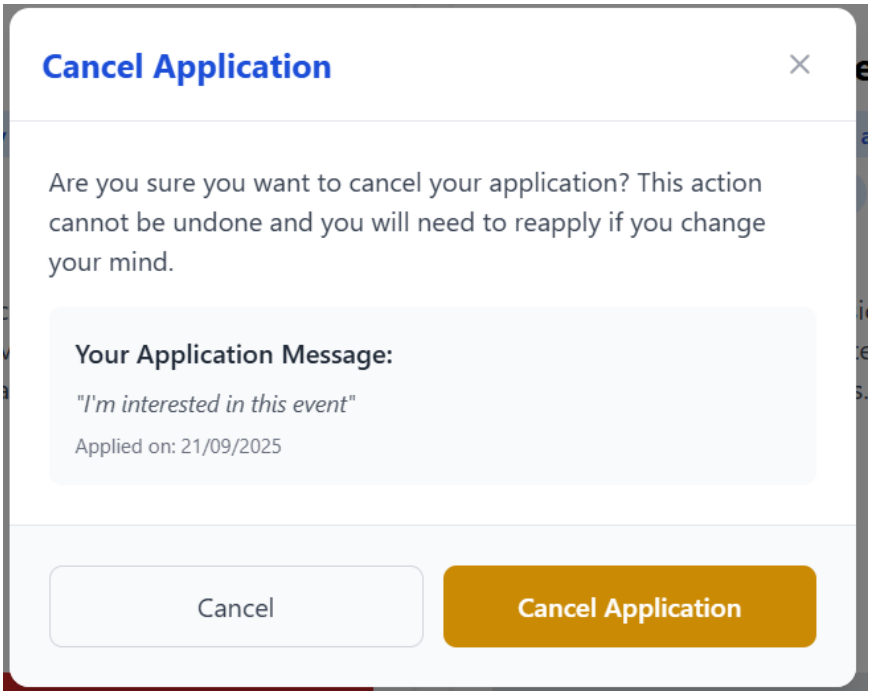



Figure 6-22: Cancel application confirmation modal displayed when volunteer click cancel application.

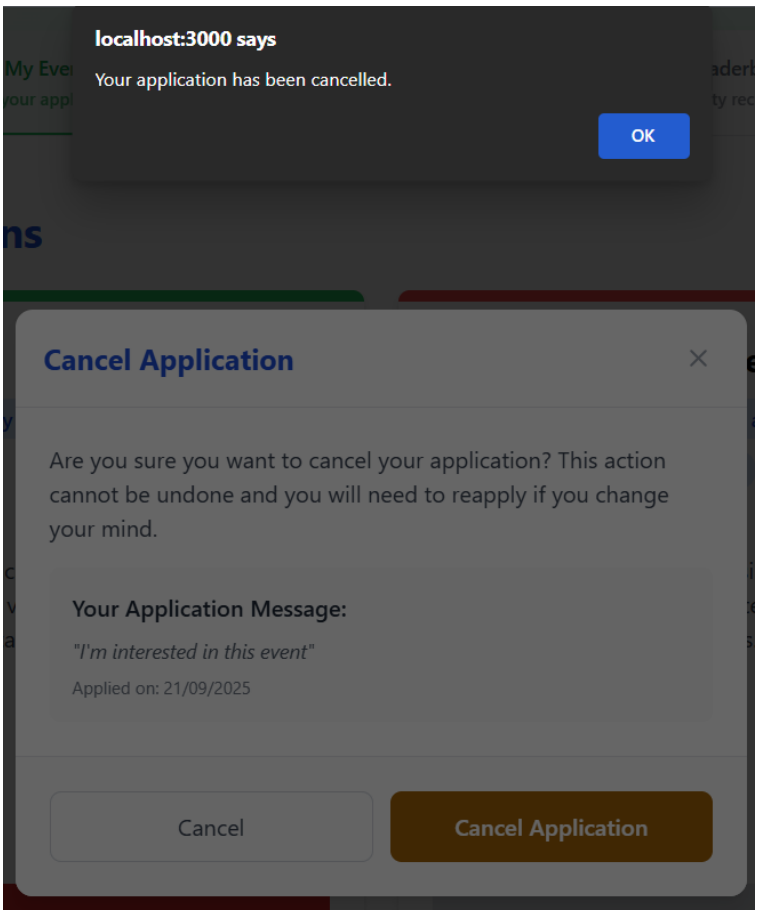



Figure 6-23: Application successfully cancelled and deleted from database.

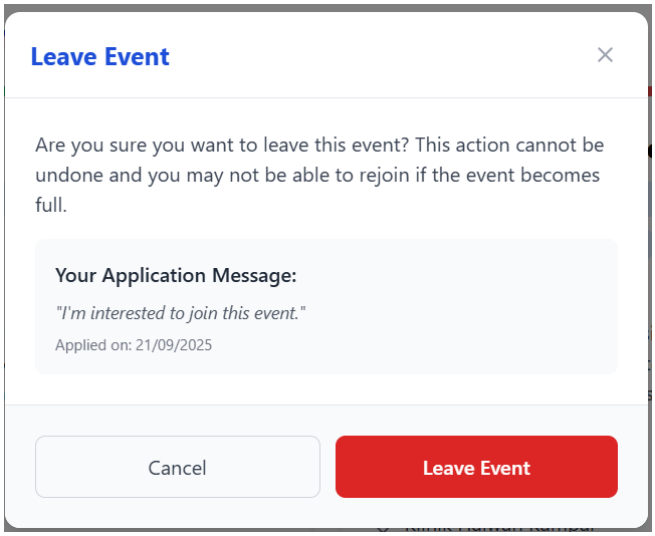



Figure 6-24: Leave event confirmation modal displayed when volunteer click leave event.

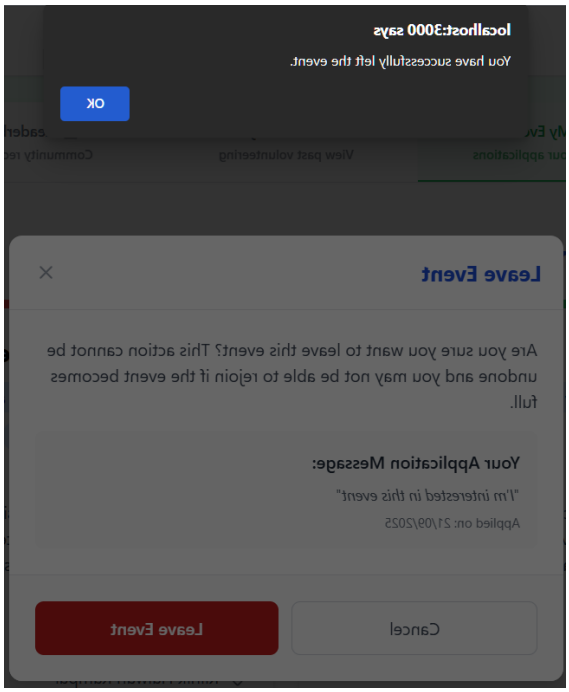


Figure 6-25: Event successfully leaved and deleted application and participation record in database.

The figures above show the outcome of the five use cases of Event Participation Module. When volunteer navigates to Event Participation Module, applied events with corresponding application status will be displayed. Volunteer can perform different actions based on the application status. For pending events, volunteer can choose to cancel the application while leave the event if the application status is approved. All the actions can be successfully updated to database. Therefore, all the use cases of Event Participation Module are valid as shown below:

Table 6-6: Outcome of Event Participation Module use case.

| Module | Use Case (Input) | Expected Output | Outcome |
|---|---|---|---|
| Event Participation | Volunteer navigates to event participation module | Display correct application details from database and corresponding action button | ✓ |
| | Volunteer click cancel application button for pendir | Display cancel application confirmation modal | ✓ |
| | Volunteer confirm cancel application | Application successfully cancelled and deleted from database | ✓ |
| | Volunteer click leave event button for approved app | Display leave event confirmation modal | ✓ |
| | Volunteer confirm leave event | Event successfully leaved and deleted application and participation record in database | ✓ |

### 6.2.6 Outcome of Attendance Module Use Case

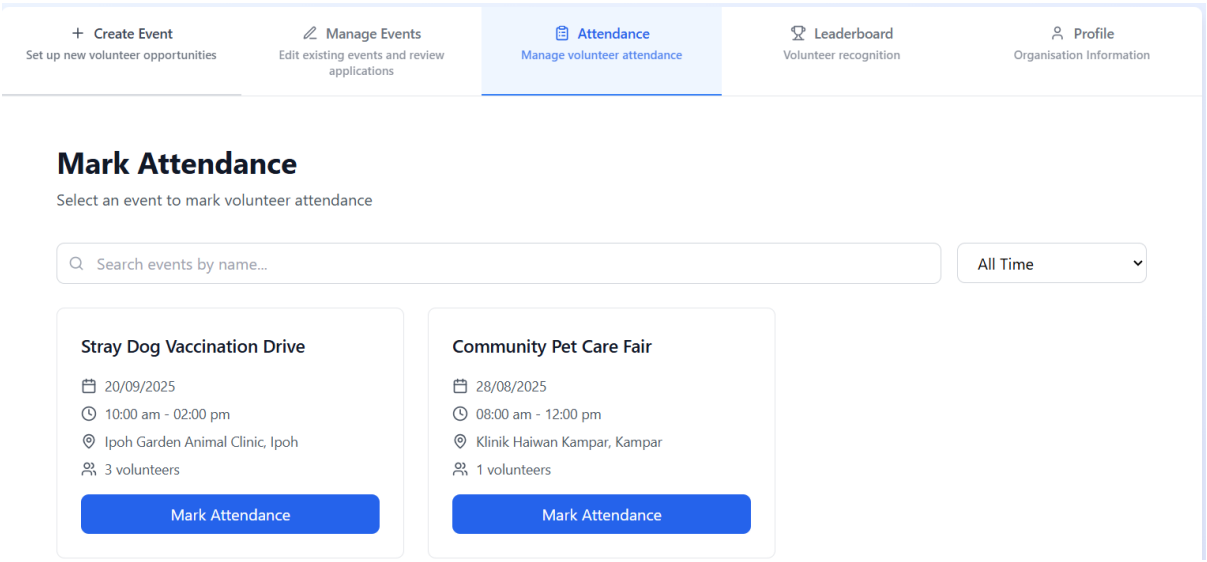



Figure 6-26: Past created events displayed for marking volunteer attendance when organiser navigates to Attendance Module.

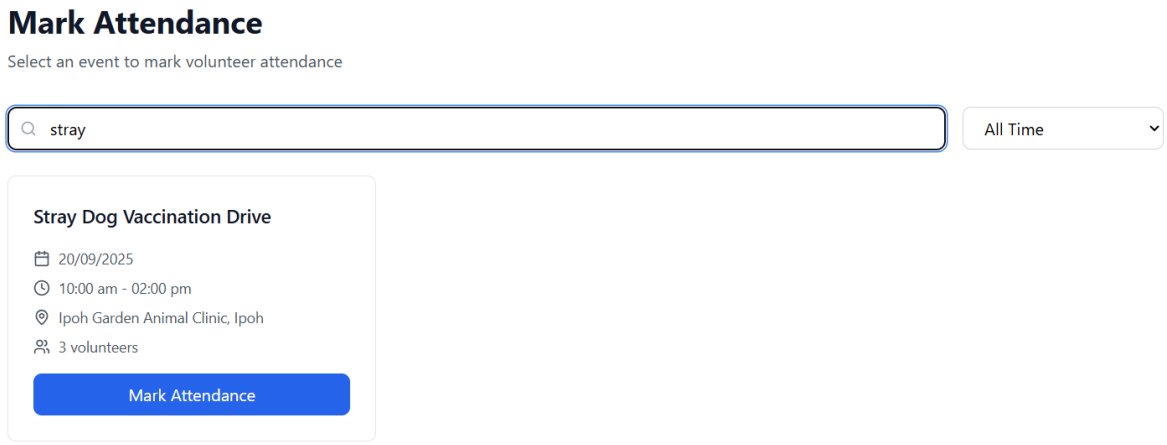



Figure 6-27: Event name filter works effectively.

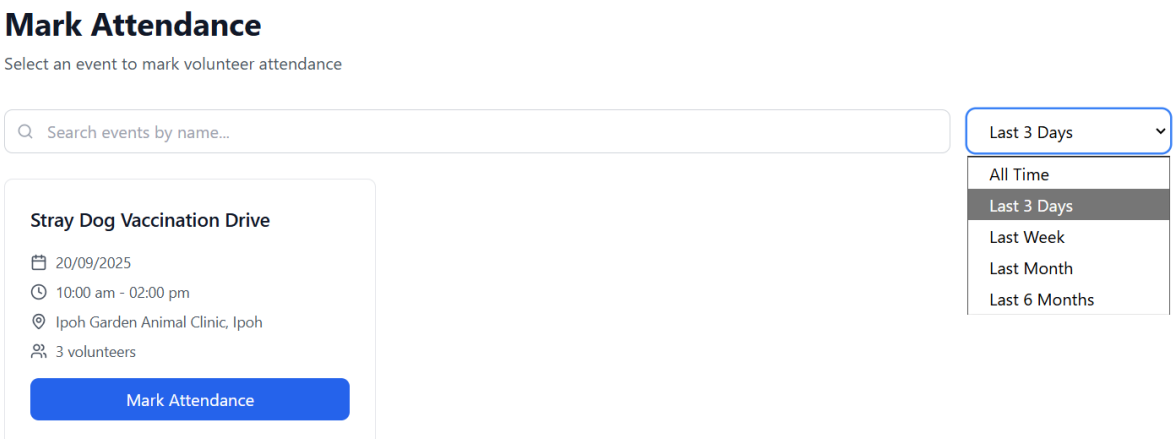



Figure 6-28: Time period filter works effectively.

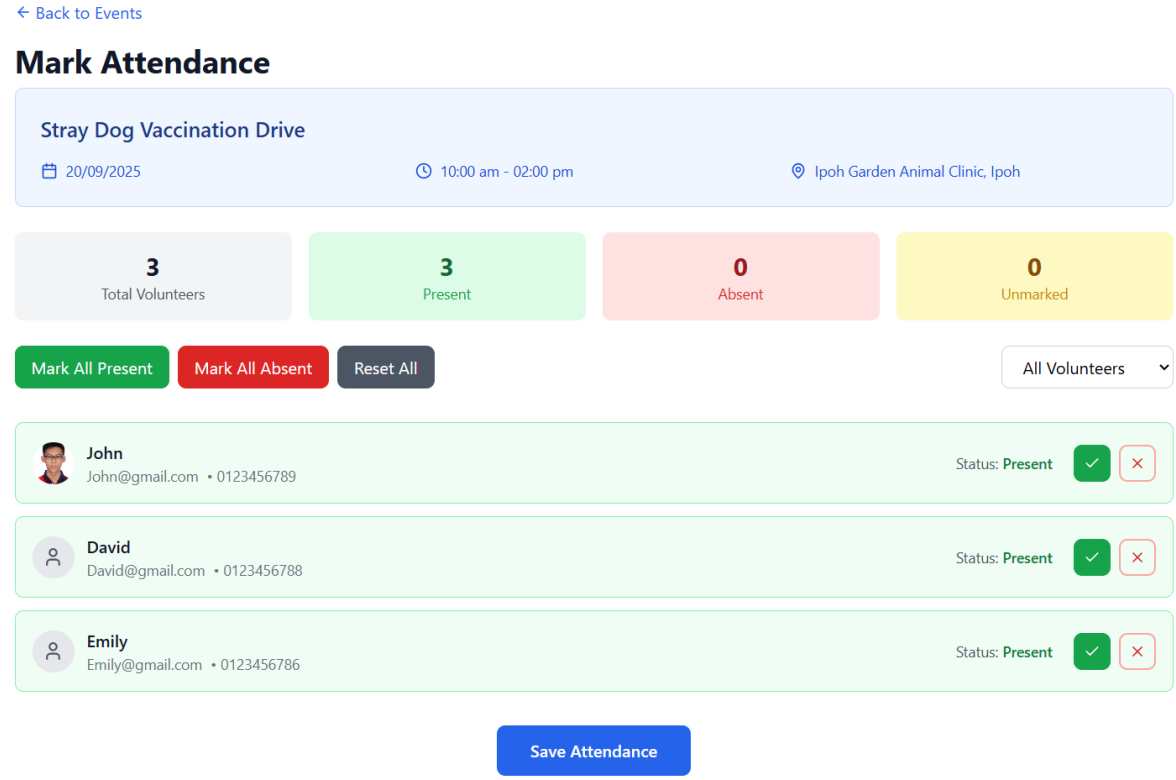



Figure 6-29: Volunteer attendance status displayed successfully when organiser clicks mark attendance, all filters work and attendance can be updated in database when save attendance button is clicked.

The figures above show the outcome of the ten use cases of Attendance Module. When organiser navigates to Attendance Module, past created events successfully displayed for mark attendance purpose. Search event filter and time period filter work normally to search specific events efficiently. When organiser click mark attendance button, volunteer attendance status also displayed successfully. Moreover, all the filters such as mark all present, mark all absent,

reset all can function to quickly mark attendance, and attendance status filter can also work to quickly filter volunteer by their attendance status. After marking attendance, all the records can be updated in database. Therefore, all the use cases of Attendance Module are valid as shown below:

Table 6-7: Outcome of Attendance Module use case.

| Module | Use Case (Input) | Expected Output | Outcome |
|---|---|---|---|
| Attendance | Organiser navigates to attendance module | Display past created events from database for marking attendance | ✓ |
| | Organiser search event by using event name filter | Display events that match the event name filter | ✓ |
| | Organiser filter events by using time period filter | Display events that match the selected time period | ✓ |
| | Organiser click mark attendance button | Display volunteer attendance status from database | ✓ |
| | Organiser click mark all present button | Automatically mark all volunteer present | ✓ |
| | Organiser click mark all absent button | Automatically mark all volunteer absent | ✓ |
| | Organiser click mark reset all button | Automatically reset all volunteer attendance status | ✓ |
| | Organiser click tick button for particular volunteer | Mark the volunteer present | ✓ |
| | Organiser click cross button for particular volunteer | Mark the volunteer absent | ✓ |
| | Organiser click save attendance button | Save attendance status in database and log volunteered hours | ✓ |

### 6.2.7 Outcome of History Module Use Case

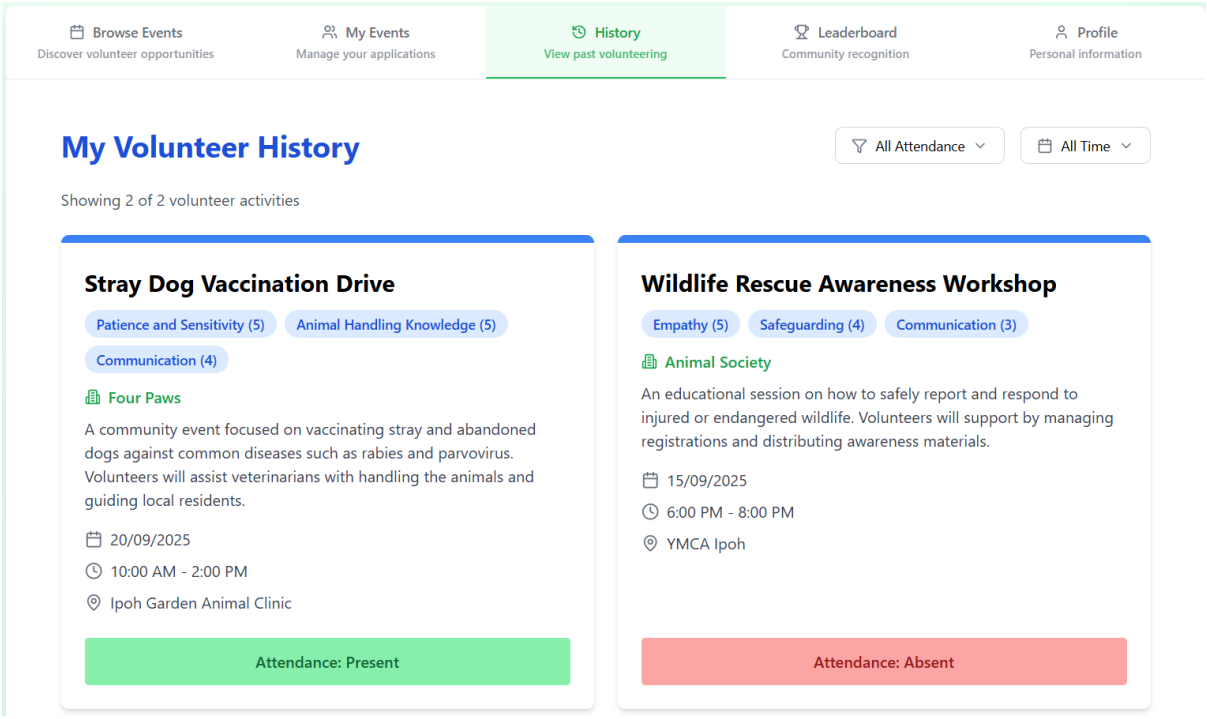



Figure 6-30: Participated events history displayed successfully when volunteer navigates to History Module.

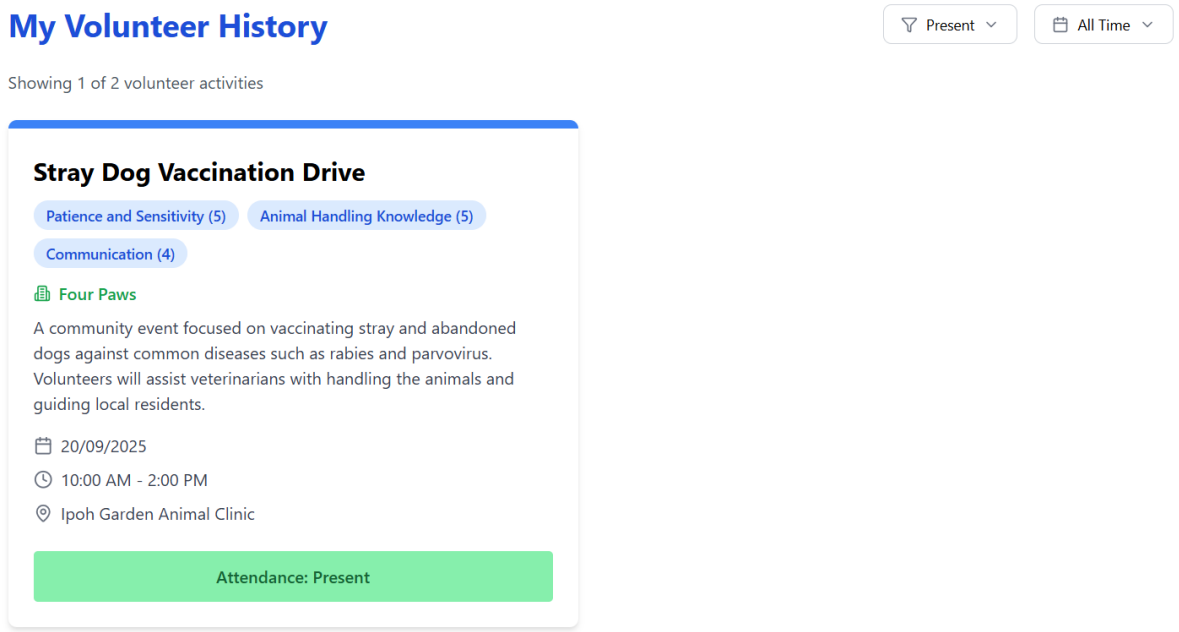



Figure 6-31: Attendance status filter functions normally.

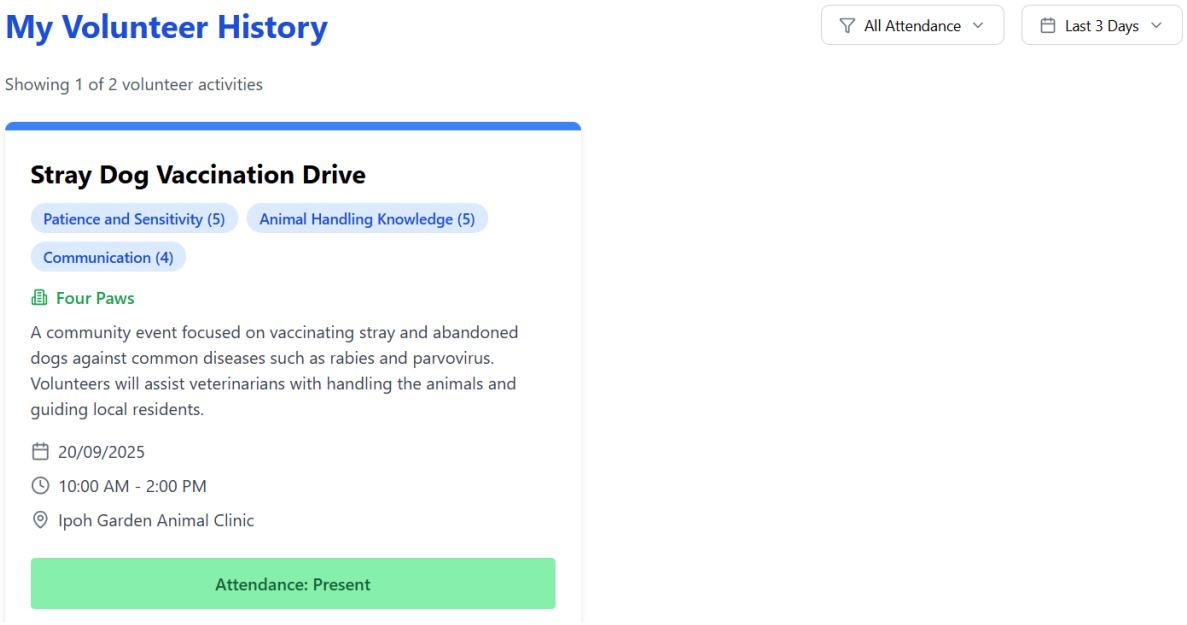



Figure 6-32: Time period filter functions normally.

The figures above show the outcome of the ten use cases of History Module. When volunteer navigates to History Module, participated events history along with attendance status successfully displayed. Furthermore, the attendance status filter and time period filter functions normally to help volunteers quickly search for specific event history. Therefore, all the use cases of History Module are valid as shown below:

Table 6-8: Outcome of History Module use case.

| Module | Use Case (Input) | Expected Output | Outcome |
|---|---|---|---|
| History | Volunteer navigates to history module | Display participated event histories and attendance status from database | ✓ |
| | Volunteer filter event histories by using attendance status filter | Display event histories that match the attentance status | ✓ |
| | Volunteer filter event histories by using time period filter | Display event histories that match the time period | ✓ |

### 6.2.8 Outcome of Profile Module Use Case

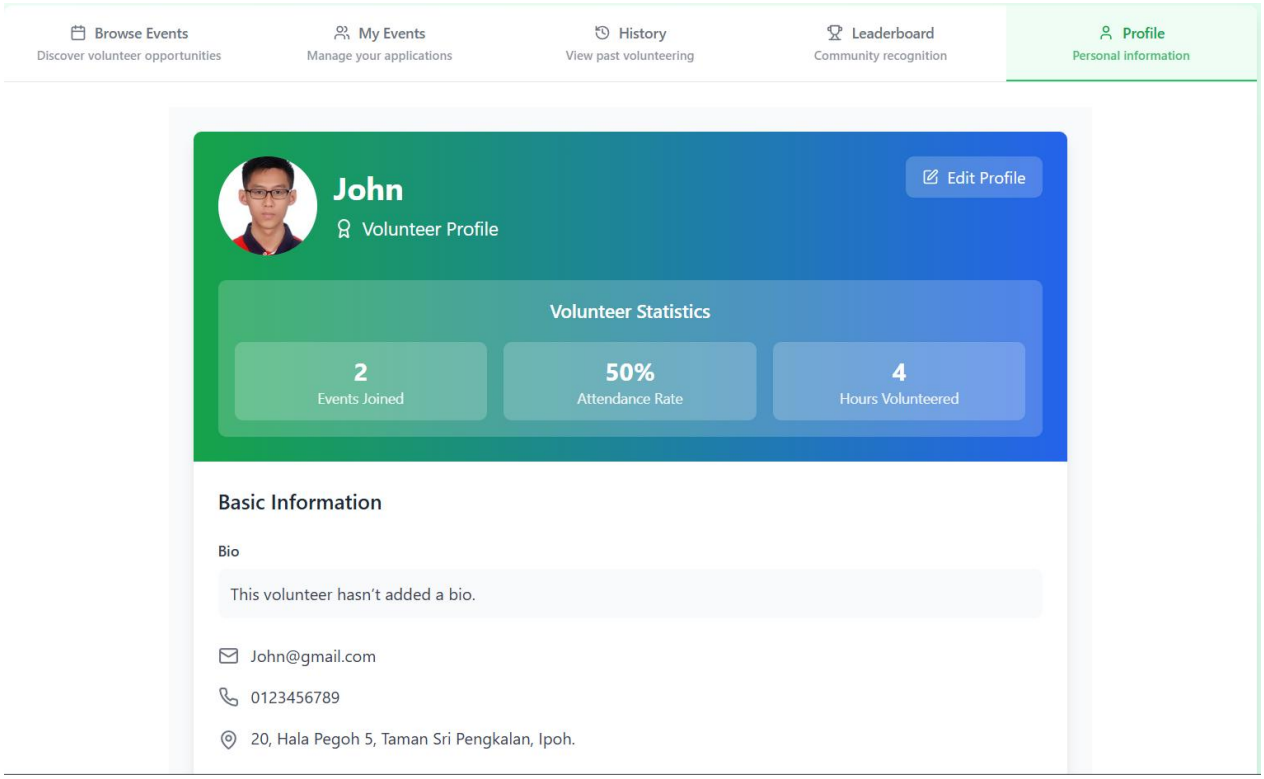



Figure 6-33: Profile information successfully displayed when user navigates to Profile Module.

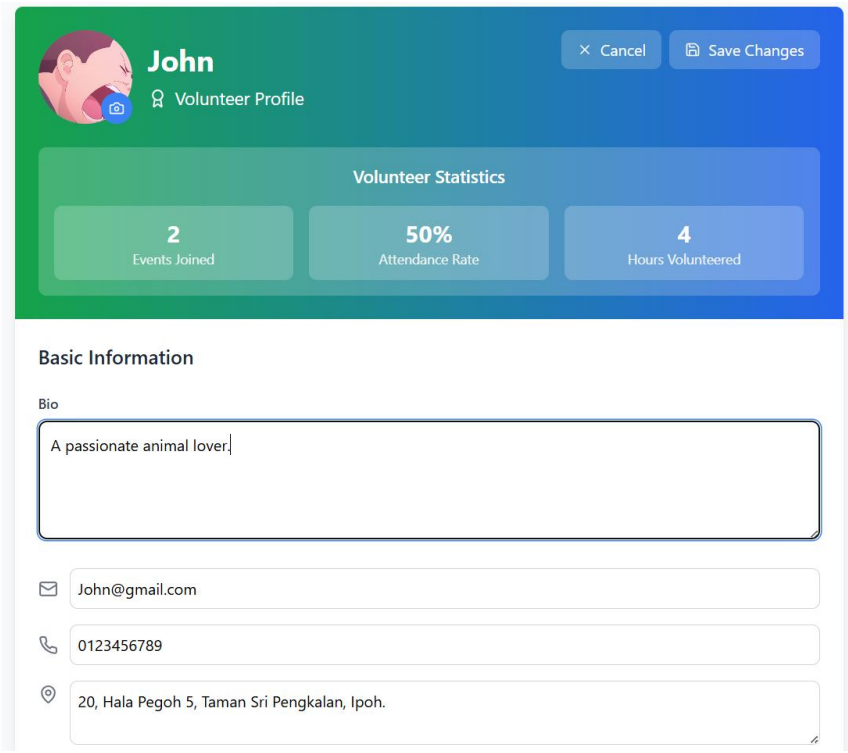



Figure 6-34: Profile information becomes editable when edit profile button is clicked.

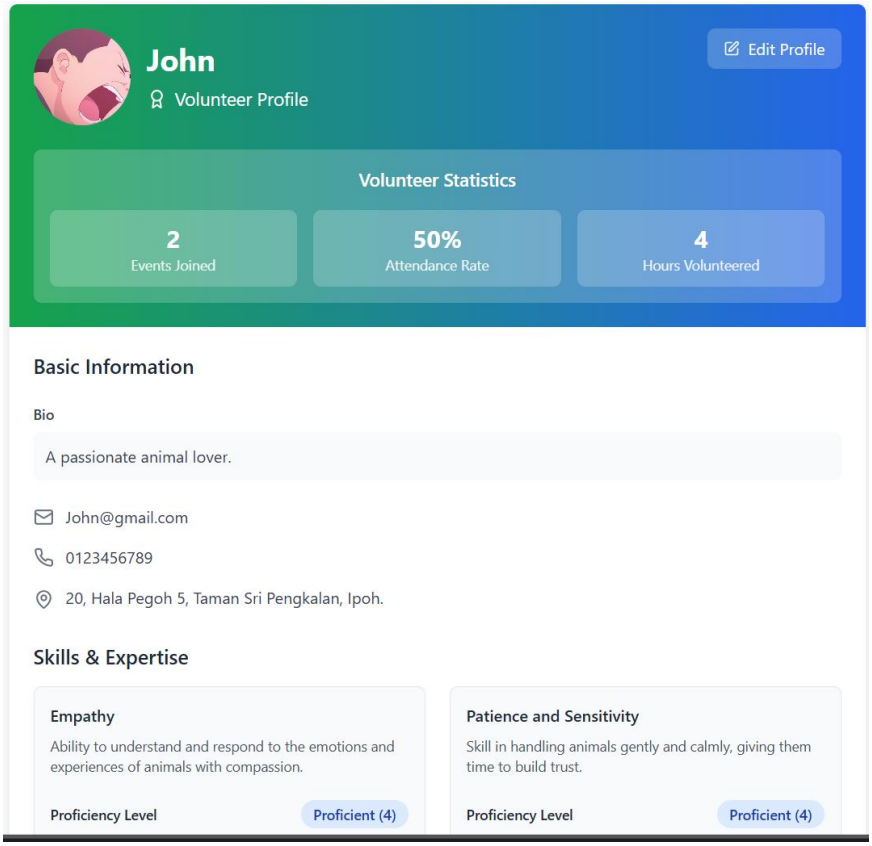



Figure 6-35: Profile information updated in database once save changes button is clicked.

The figures above show the outcome of the four use cases of Profile Module. When users navigate to Profile Module, they can view their profile statistics and personal information. Besides, they can also edit profile such as upload profile picture, edit personal information, and update skill proficiency level. Therefore, all the use cases of Profile Module are valid as shown below:

Table 6-9: Outcome of Profile Module use case.

| Module | Use Case (Input) | Expected Output | Outcome |
|---|---|---|---|
| Profile | Volunteer or organiser navigate to profile module | Display profile statistics and information from database | ✓ |
| | Volunteer or organiser click edit profile button | Profile information become editable | ✓ |
| | Volunteer or organiser click upload profile picture button | Open file explorer for selecting profile picture | ✓ |
| | Volunteer or organiser edit their profiles and click save button | Update the profile information in database | ✓ |

### 6.2.9 Outcome of Leaderboard Module Use Case

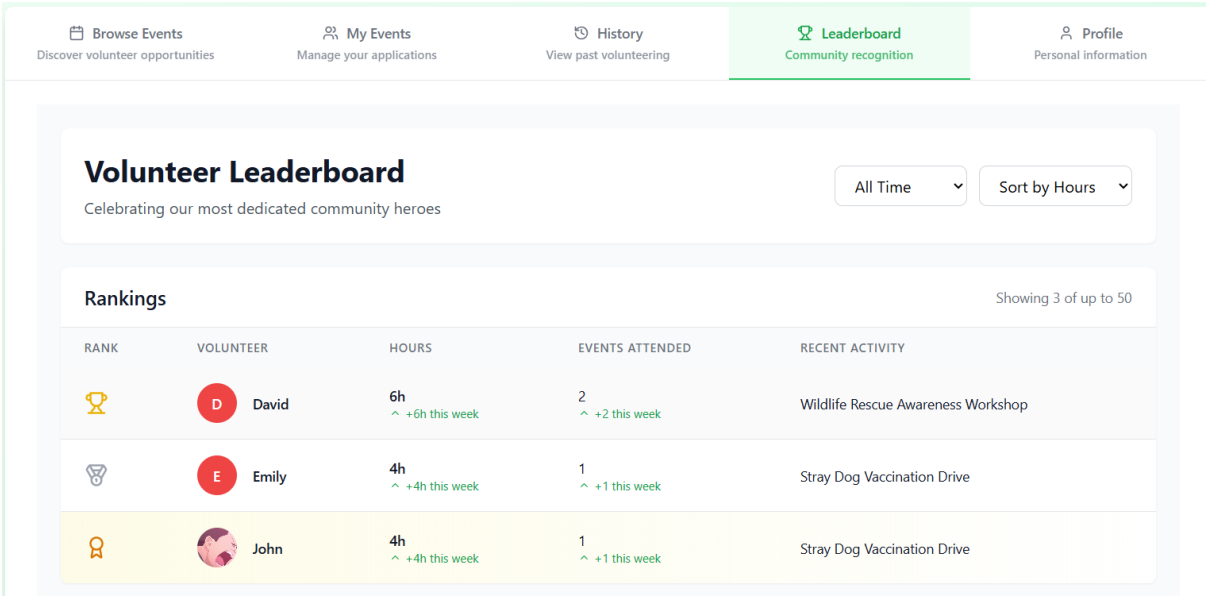



Figure 6-36: Volunteer leaderboard displayed when users navigate to Leaderboard Module, time period filter and sort by hours/events filter function normally.

The figure above show the outcome of the three use cases of Leaderboard Module. When users navigate to Leaderboard Module, they can view volunteer leaderboard which serves as a gamification purpose to recognise volunteer contribution. Furthermore, users can filter the leaderboard by time period or sort the leaderboard by hours volunteered or number of events participated. Therefore, all the use cases of Leaderboard Module are valid as shown below:

Table 6-10: Outcome of Leaderboard Module use case.

| Module | Use Case (Input) | Expected Output | Outcome |
|---|---|---|---|
| Leaderboard | Volunteer or organiser navigate to leaderboard module | Display volunteer leaderboard from database | ✓ |
| | Volunteer or organiser filter leaderboard by using time period filter | Display leaderboard that match the time period | ✓ |
| | Volunteer or organiser use sort filter to sort leaderboard by hours or events | Display leaderboard that match the sort filter | ✓ |

### 6.2.10 Outcome of Event Matching Algorithm Use Case

The event matching algorithm was tested to ensure that both the weighted-based skill matching algorithm and distance-based matching algorithm produced accurate results, and that their combination yielded a correct final matching score for each event. All matching scores generated by the system were verified to be accurate, and for demonstration purposes, David's volunteer profile was used as an example in the testing.

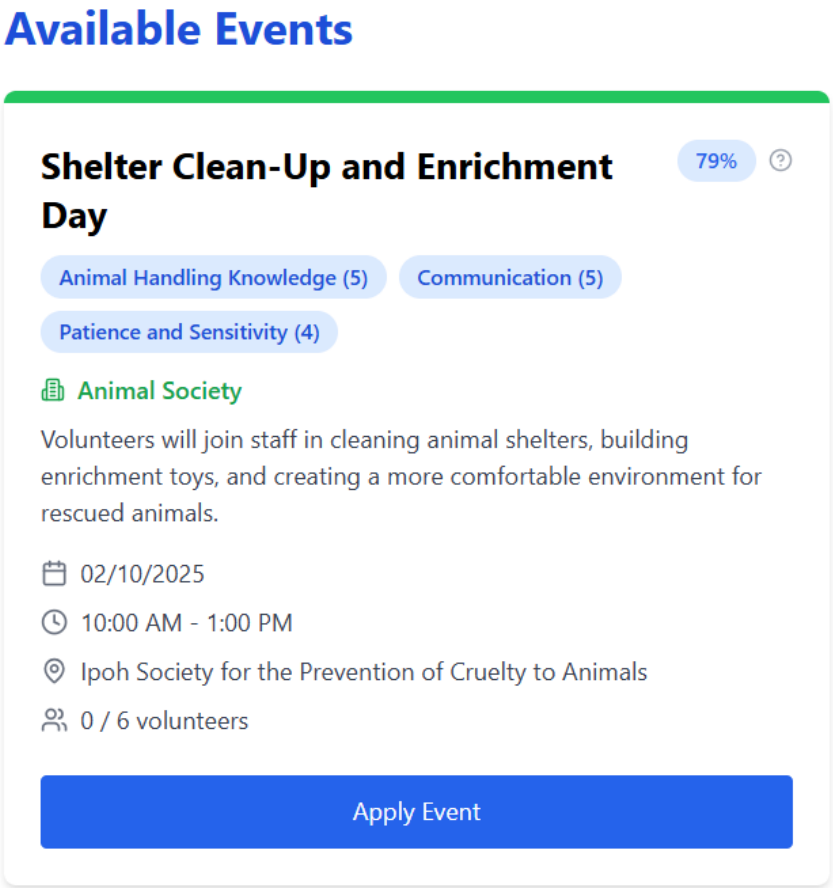



Figure 6-37: Available event example.

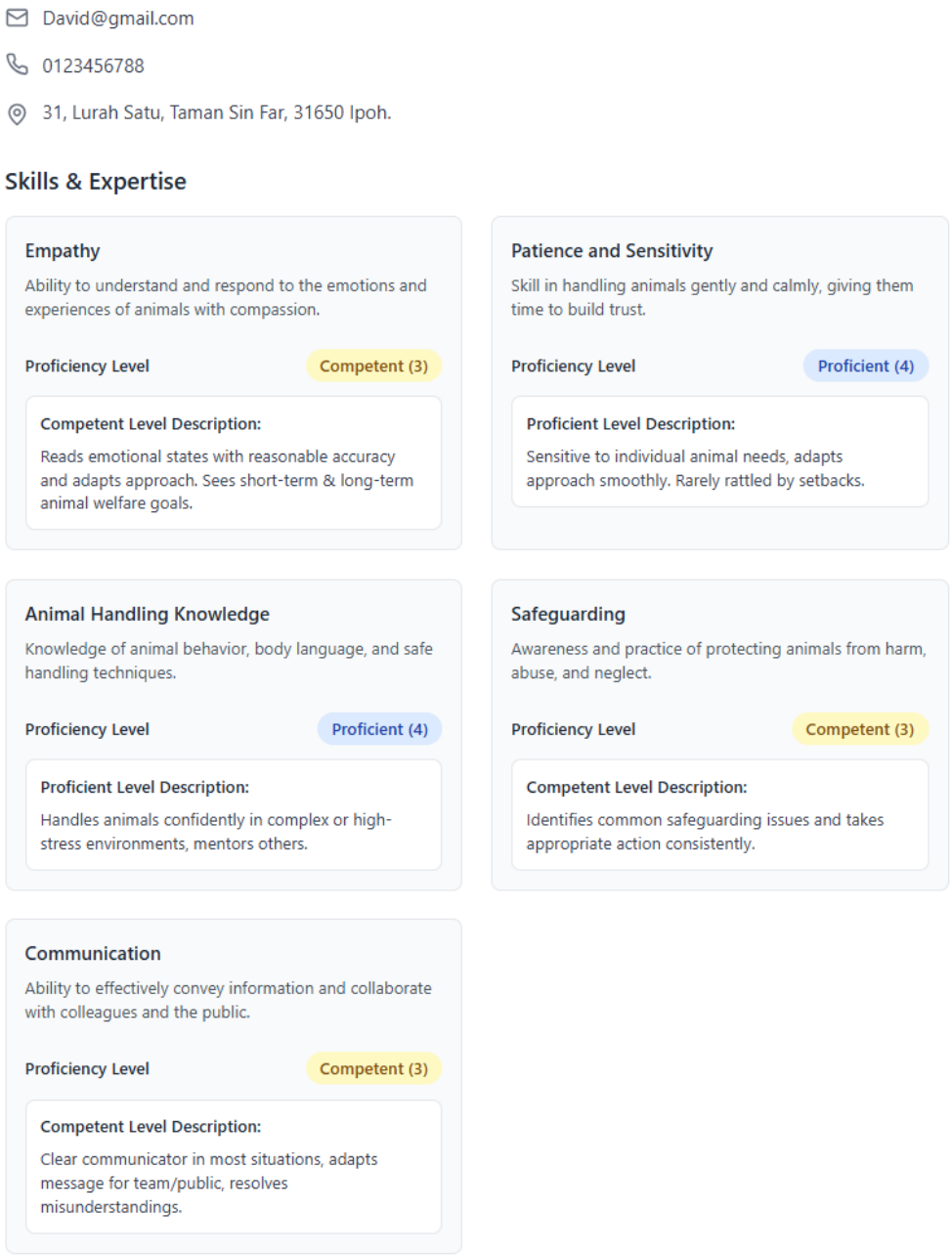



Figure 6-38: David's volunteer profile.

Based on the figures above, the event skills importance requirement are Animal Handling Knowledge (5), Communication (5), and Patience and Sensitivity (4). On the other hand, David's skill proficiency are Empathy (3), Patience and Sensitivity (4), Animal Handling Knowledge (4), Safeguarding (3), and Communication (3). Therefore, the skill matching score is $(((5 \times 4) + (5 \times 3) + (4 \times 4)) / ((5 \times 5) + (5 \times 5) + (4 \times 5))) \times 100\% = 72.86\%$.

Next, we can note that the event location is at Ipoh Society for the Prevention of Cruelty to Animals while David's address is at 31, Lurah Satu, Taman Sin Far, 31650 Ipoh. The figure below shows the distance between event location and David's address:

```
Event: Shelter Clean-Up and Enrichment Day | Location: Ipoh Society for the Prevention of Cruelty to Animals | Distance: 6.078 km
```

Figure 6-39: Distance between event location and David's address (6.078km).

Therefore, the proximity score obtained is $e^{-0.025(6.078)} \times 100\% = 85.90\%$ and the computed final matching score is $(72.86\% \times 0.5) + (85.90\% \times 0.5) = 79.38\%$. This matches the matching score displayed in the Event Listing Module as shown below:

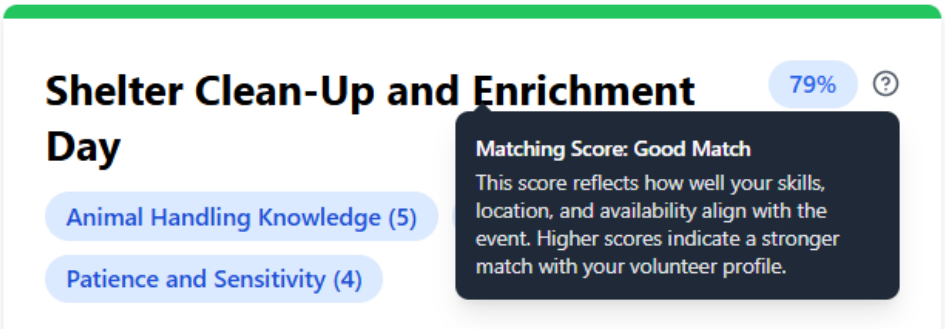



Figure 6-40: Matching score 79% displayed in Event Listing Module.

Therefore, the use cases of Event Matching Algorithm are valid as shown below:

Table 6-11: Outcome of Event Matching Algorithm use case.

| Module | Use Case (Input) | Expected Output | Outcome |
|---|---|---|---|
| Event Matching Algorithm | Organiser defines skills importance level and volunter self-rated their skill proficiency | Calculate skill matching score accurately using weighted-based skill matching algorithm | ✓ |
| | Organiser defines event location and volunter update their address | Calculate proximity score accurately using distance-based skill matching algorithm | ✓ |
| | Combination of both algorithms | Calculate final matching score accurately | ✓ |

## 6.3 Objectives Evaluation

The evaluation of the system is guided by the research objectives, each of which was designed to address specific challenges in existing volunteer management systems. The first objective was to develop a comprehensive and organised system for managing and indexing volunteer skills through a weighted matching algorithm. This has been accomplished by implementing a dual-scale approach where organisers define the importance of skills required for an event and volunteers self-rate their proficiency levels. The system then applies a weighted-based skill matching algorithm to calculate the skill matching scores. In chapter 6.2.10, the testing demonstrated that the algorithm produced correct results, as shown in the case of David's

dashboard, where the computed score aligned with the value displayed in the Event Listing Module. This confirms that the system effectively overcomes the limitations of traditional keyword-based matching by taking into account both competency and event requirements, thereby reducing mismatches and improving role assignments.

The second objective aimed to enhance volunteer-event matching by incorporating a distance-based proximity algorithm. This objective was achieved, as the system successfully retrieved distances between volunteer addresses and event locations using the Google Maps Distance Matrix API and translated them into proximity scores. Validation was performed using the distance-based matching algorithm: Proximity Score (%) = $e^{-0.025(\text{distance})} \times 100\%$, ensuring that the computed scores were accurate. When combined with the skill matching algorithm, the final matching score accurately reflected both suitability and accessibility. For example, testing with David's address produced a proximity score and final matching score that were consistent with the system's event listing results. This demonstrates that the system not only improves the recommendation of events but also addresses the geographical barriers that often discourage volunteer participation.

The third objective was to develop a system that recognises volunteer contributions in meaningful ways. This was accomplished through the implementation of the Leaderboard Module, which ranks volunteers based on hours volunteered or number of events attended. Testing results showed that the leaderboard functions correctly with filtering and sorting options, allowing recognition to be tailored to different time frames and metrics. By making volunteer efforts visible, the leaderboard encourages motivation, fosters healthy competition, and promotes long-term engagement. This directly addresses the problem of low volunteer retention caused by inadequate recognition in traditional systems.

In summary, the evaluation shows that all three objectives have been met. The system delivers accurate skill-based matching, integrates proximity considerations into event recommendations, and incorporates gamification to recognise volunteer contributions. Together, these outcomes demonstrate that the developed Volunteer Management System effectively addresses the identified challenges, improving efficiency for organisers and creating a more engaging environment for volunteers.

# Chapter 7
# Conclusion and Recommendation

## 7.1 Conclusion

Traditional volunteer management systems often face significant limitations that hinder the effectiveness of volunteer programs. Many existing systems rely on manual or keyword-based methods for matching volunteers to events, which can result in poor alignment between volunteer skills and event requirements. In addition, geographic constraints and accessibility are often overlooked, making it difficult for volunteers to participate in events that are suitable and nearby. Furthermore, these systems typically lack features that recognise and reward volunteer contributions, which can lead to decreased motivation, lower engagement, and ultimately, higher volunteer turnover. Such challenges highlight the need for a more intelligent and comprehensive approach to volunteer management.

To address these issues, this project developed a web-based Volunteer Management System (VMS) that combines both skill-based and distance-based criteria for volunteer-event matching, along with an intuitive user interface and structured database management. The system allows organisers to define the importance of specific skills for each event, while volunteers self-assess their proficiency levels. These inputs are first processed through a weighted skill-matching algorithm to obtain a skill score, and then a distance-based proximity algorithm calculates a proximity score based on the volunteer's location relative to the event. Both scores are combined to generate a final matching score, ensuring that volunteers are assigned to events that not only suit their abilities but are also accessible, thereby improving overall participation and engagement.

Beyond matching, the system provides comprehensive functionalities for both volunteers and organisers, including event creation, event modification, application management, attendance tracking, volunteer profiles, and a leaderboard to track volunteer contributions. The leaderboard serves as a gamification element to encourage sustained engagement, foster healthy competition, and acknowledge volunteer efforts in a meaningful way through recognition of hours volunteered or events participated. This is crucial because it provides visible recognition of contributions, motivating volunteers to remain active, improve their

skills, and participate in more events, which ultimately enhances overall volunteer retention and program effectiveness.

Overall, this project demonstrates a significant advancement over traditional volunteer management approaches by addressing the key challenges of skill-event mismatch, accessibility, and volunteer recognition. The developed VMS not only enhances operational efficiency for organisers but also creates a more engaging and rewarding environment for volunteers. Its contributions include providing accurate and dynamic event recommendations and streamlining volunteer and event management processes. Additionally, the system promotes long-term volunteer motivation and retention. Together, these features support the growth and impact of community-based initiatives.

## 7.2 Recommendation

While the development and implementation of this Web-based Volunteer Management System (VMS) have successfully addressed challenges related to volunteer-event matching, accessibility, and volunteer recognition, there remain several opportunities for future enhancement. These improvements could further increase volunteer engagement, streamline organiser operations, and enhance the overall effectiveness and impact of the system.

Firstly, the system can be enhanced by incorporating real-time notifications and alerts. This feature would automatically notify volunteers about upcoming events, changes in schedules, application status updates, or new opportunities matching their skills and preferences. By keeping volunteers informed in real-time, the system can reduce missed events, improve overall participation rates, and help organisers maintain better coordination. Notifications could be delivered via email, or even in-app messages and push notifications if a mobile version is developed in the future, providing timely reminders and increasing user engagement. Overall, this enhancement would ensure volunteers stay informed and engaged, contributing to smoother event participation and better system utilisation.

Since many volunteers primarily rely on smartphones and other mobile devices, expanding the system to include dedicated mobile application support would significantly enhance accessibility. A mobile version of the VMS would allow volunteers to browse upcoming events, submit applications, track their participation hours, and receive updates directly from their

mobile devices. This convenience enables volunteers to interact with the system anytime and anywhere, reducing barriers to engagement and encouraging more frequent participation. Additionally, a mobile platform could incorporate location-based features, allowing volunteers to view the exact location of events, and receive step-by-step navigation to the event site using integrated GPS services. This feature would simplify commuting to events, reduce lateness, and improve overall convenience. Overall, mobile support would broaden the user base, enhance user experience, and strengthen volunteer engagement.

Finally, the system can be enhanced by integrating social and collaborative features within the platform. By introducing volunteer forums, discussion boards, or group participation options, the system can foster a sense of community and shared purpose among volunteers. Additionally, integration with social media platforms such as Facebook or Instagram could allow volunteers to share their activities, achievements, and event participation with their wider networks. These features would enable users to coordinate as a team for events, motivate one another through peer interactions, and increase the visibility of volunteer efforts. By creating opportunities for social engagement and wider outreach, the system can strengthen volunteer retention, enhance the overall experience, and encourage sustained participation over the long term.

By adopting these recommendations, the VMS can evolve into a more comprehensive and adaptive platform that better serves both organisers and volunteers. This would further support community initiatives and enhance overall volunteer management effectiveness.

# REFERENCES

[1] Bloomerang, "15+ Top Volunteer Management Software Options for Nonprofits," Aug. 9, 2024. [Online]. Available: https://bloomerang.co/blog/volunteer-management-software/

[2] I. M. Jones, "A complete guide to volunteer management," *CHARITY DIGITAL,* Aug. 9, 2024. [Online]. Available: https://charitydigital.org.uk/topics/a-complete-guide-to-volunteer-management- systems-9021#list3

[3] PMO Global Institute, "What Is a Volunteer Management System?," Medium, Aug. 9, 2024. [Online]. Available: https://medium.com/@seo.pmaspire_64183/what-is-a-volunteer-management-system- f2801865b718

[4] VolunteerHub, "Learn More About VolunteerHub," Aug. 10, 2024. [Online]. Available: https://volunteerhub.com/about

[5] VolunteerHub, "Recruit More Volunteers with VolunteerHub," Aug. 10, 2024. [Online]. Available: https://volunteerhub.com/platform/volunteer-recruitment

[6] VolunteerHub, "Streamline Volunteer Scheduling with VolunteerHub," Aug. 10, 2024. [Online]. Available: https://volunteerhub.com/platform/volunteer-scheduling

[7] VolunteerHub, "Easily Track Volunteer Hours with VolunteerHub," Aug. 11, 2024. [Online]. Available: https://volunteerhub.com/platform/volunteer-

hour-tracking

[8] VolunteerHub, "Access an Intuitive Volunteer Database with VolunteerHub," Aug. 11, 2024. [Online]. Available: https://volunteerhub.com/platform/volunteer-database

[9] VolunteerHub, "Make Running Reports a Breeze with VolunteerHub," Aug. 11, 2024. [Online]. Available: https://volunteerhub.com/platform/reporting

[10] VolunteerHub, "Reward and Recognize Volunteers with VolunteerHub," Aug. 12, 2024. [Online]. Available: https://volunteerhub.com/platform/rewards-and-recognition

[11] VolunteerHub, "Ask Volunteers to Donate with VolunteerHub," Aug. 12, 2024. [Online]. Available: https://volunteerhub.com/platform/volunteer-fundraising

[12] VolunteerHub, "Easily Manage Liability Waivers with VolunteerHub," Aug. 12, 2024. [Online]. Available: https://volunteerhub.com/platform/volunteer-liability-waivers

[13] VolunteerHub, "Reduce Redundancy with Multi-Event Editor," Aug. 13, 2024. [Online]. Available: https://volunteerhub.com/platform/multi-event-editor

[14] VolunteerHub, "Organize Volunteers with Group Manager," Aug. 13, 2024. [Online]. Available: https://volunteerhub.com/platform/group-organization

[15] VolunteerHub, "Collect Better Data with Configurable Forms," Aug. 13, 2024. [Online]. Available: https://volunteerhub.com/platform/configurable-forms

[16] VolunteerHub, "Send Automated or Real-Time Emails to Volunteers with

VolunteerHub," Aug. 14, 2024. [Online]. Available: https://volunteerhub.com/platform/automated-emails

[17] VolunteerHub, "Integrate VolunteerHub with Your Social Media Accounts," Aug. 14, 2024. [Online]. Available: https://volunteerhub.com/platform/social-media-integration

[18] POINT, "One app to volunteer or give to any cause," Aug. 14, 2024. [Online]. Available: https://pointapp.org/volunteers/

[19] POINT, "Website Integrations," Aug. 14, 2024. [Online]. Available: https://pointapp.org/nonprofit-features/website-integration/

[20] POINT, "Easy Event Sign Up and Check-In," Aug. 14, 2024. [Online]. Available: https://pointapp.org/nonprofit-features/sign-up-and-check-in/

[21] POINT, "Volunteer Platform," Aug. 14, 2024. [Online]. Available: https://pointapp.org/nonprofit-features/volunteer-experience/

[22] POINT, "POINT Page (Volunteer Hub)," Aug. 16, 2024. [Online]. Available: https://pointapp.org/nonprofit-features/point-page-volunteer-hub/

[23] POINT, "Volunteer Activity Profiles," Aug. 16, 2024. [Online]. Available: https://pointapp.org/nonprofit-features/volunteer-activity-reports//

[24] POINT, "Volunteer Programs," Aug. 16, 2024. [Online]. Available: https://pointapp.org/nonprofit-features/program-management/

[25] POINT, "Opportunity, Event and Shifts," Aug. 17, 2024. [Online]. Available: https://pointapp.org/nonprofit-features/shift-management/

[26] POINT, "Calendar View," Aug. 17, 2024. [Online]. Available:

https://pointapp.org/nonprofit-features/calendar-view/

[27] POINT, "Cohosts," Aug. 17, 2024. [Online]. Available: https://pointapp.org/nonprofit-features/corporate-partner-collaboration/

[28] POINT, "Background Checks," Aug. 18, 2024. [Online]. Available: https://pointapp.org/nonprofit-features/background-checks/

[29] POINT, "Email Messages," Aug. 18, 2024. [Online]. Available: https://pointapp.org/nonprofit-features/email-messages/

[30] POINT, "Volunteer Time Tracking," Aug. 18, 2024. [Online]. Available: https://pointapp.org/nonprofit-features/volunteer-time-tracking/

[31] POINT, "Checkin Stations: Kiosk + QR," Aug. 19, 2024. [Online]. Available: https://pointapp.org/nonprofit-features/check-in-stations/

[32] POINT, "Volunteer Self Submission," Aug. 19, 2024. [Online]. Available: https://pointapp.org/nonprofit-features/self-submitted-volunteer-hours//

[33] POINT, "Auto Impact Reports," Aug. 19, 2024. [Online]. Available: https://pointapp.org/nonprofit-features/volunteer-demographics/

[34] POINT, "Donation Button + Form Integrations," Aug. 19, 2024. [Online]. Available: https://pointapp.org/nonprofit-features/donation-button-form-integrations/

[35] GetConnected by Galaxy Digital, "We Make Volunteer Management Easy," Aug. 23, 2024. [Online]. Available: https://www.galaxydigital.com/about-us

[36] GetConnected by Galaxy Digital, "W Volunteer Hours Tracking," Aug. 23, 2024. [Online]. Available: https://www.galaxydigital.com/features/volunteer-

tracking-software

[37] GetConnected by Galaxy Digital, "Skills-Based Volunteerism," Aug. 23, 2024. [Online]. Available: https://www.galaxydigital.com/features/skills-based-volunteerism/

[38] GetConnected by Galaxy Digital, "Volunteer Group Engagement," Aug. 23, 2024. [Online]. Available: https://www.galaxydigital.com/features/volunteer-group-engagement/

[39] GetConnected by Galaxy Digital, "Volunteer Waiver Tracking," Aug. 23, 2024. [Online]. Available: https://www.galaxydigital.com/features/volunteer-waiver-tracking/

[40] GetConnected by Galaxy Digital, "eSign Documents," Aug. 23, 2024. [Online]. Available: https://www.galaxydigital.com/features/esign-documents/

[41] GetConnected by Galaxy Digital, "Event Landing Pages," Aug. 23, 2024. [Online]. Available: https://www.galaxydigital.com/features/event-landing-pages/

[42] GetConnected by Galaxy Digital, "Volunteer Calendar," Aug. 23, 2024. [Online]. Available: https://www.galaxydigital.com/features/volunteer-calendar/

[43] GetConnected by Galaxy Digital, "Easily Schedule Your Volunteers," Aug. 23, 2024. [Online]. Available: https://www.galaxydigital.com/features/volunteer-scheduling-software/

[44] GetConnected by Galaxy Digital, ""#1 Volunteer Check-In App," Aug. 24, 2024.

[Online]. Available: https://www.galaxydigital.com/features/volunteer-check-in/

[45] GetConnected by Galaxy Digital, "Automated Communication," Aug. 24, 2024. [Online]. Available: https://www.galaxydigital.com/features/automated-communication/

[46] GetConnected by Galaxy Digital, "Online Donations," Aug. 24, 2024. [Online]. Available: https://www.galaxydigital.com/features/online-donations/

[47] GetConnected by Galaxy Digital, "Volunteer App," Aug. 24, 2024. [Online]. Available: https://www.galaxydigital.com/features/volunteer-app/

[48] GetConnected by Galaxy Digital, "Volunteer Opportunity Promotion," Aug. 24, 2024. [Online]. Available: https://www.galaxydigital.com/features/volunteer-opportunity-promotion/

[49] GetConnected by Galaxy Digital, "Volunteer Profile," Aug. 24, 2024. [Online]. Available: https://www.galaxydigital.com/features/volunteer-profile/

[50] GetConnected by Galaxy Digital, "Volunteer Rewards and Recognition," Aug. 24, 2024. [Online]. Available: https://www.galaxydigital.com/features/volunteer-rewards-and-recognition/

[51] GetConnected by Galaxy Digital, "Email Messaging," Aug. 25, 2024. [Online]. Available: https://www.galaxydigital.com/features/email-marketing-for-nonprofits/

[52] GetConnected by Galaxy Digital, "Volunteer Text Messaging," Aug. 25, 2024. [Online]. Available: https://www.galaxydigital.com/features/volunteer-text-

messaging/

[53] GetConnected by Galaxy Digital, "Volunteer Hours Reporting," Aug. 25, 2024. [Online]. Available: https://www.galaxydigital.com/features/volunteer-hours-reporting/

[54] GetConnected by Galaxy Digital, "Volunteer Impact Reporting," Aug. 25, 2024. [Online]. Available: https://www.galaxydigital.com/features/volunteer-impact-reporting/

[55] GetConnected by Galaxy Digital, "Exportable Volunteer Reports," Aug. 25, 2024. [Online]. Available: https://www.galaxydigital.com/features/exportable-volunteer-reports/

[56] GetConnected by Galaxy Digital, "Volunteering Resume," Aug. 25, 2024. [Online]. Available: https://www.galaxydigital.com/features/volunteering-resume/

[57] GetConnected by Galaxy Digital, "Social Media Integration," Aug. 25, 2024. [Online]. Available: https://www.galaxydigital.com/features/social-media-integration/

[58] GetConnected by Galaxy Digital, "Disaster Management Software," Aug. 25, 2024. [Online]. Available: https://www.galaxydigital.com/disaster-management-software

[59] Merriam-Webster, "gamification," Aug. 27, 2024. [Online]. Available: https://www.merriam-webster.com/dictionary/gamification

[60] Great Learning, "History of ReactJS," Aug. 27, 2024. [Online]. Available:

https://www.mygreatlearning.com/react-js/tutorials/history-of-reactjs

[61] MDN Web Docs, "SPA (Single-page application)," Aug. 27, 2024. [Online]. Available: https://developer.mozilla.org/en-US/docs/Glossary/SPA GetConnected by Galaxy Digital, "Disaster Management Software," Aug. 27, 2024. [Online]. Available: https://www.galaxydigital.com/disaster-management-software

[62] React, "A JavaScript library for building user interfaces," Aug. 27, 2024. [Online]. Available: https://legacy.reactjs.org/

[63] AppMaster, "React's Component-Based Architecture: A Case Study," Aug. 27, 2024. [Online]. Available: https://appmaster.io/blog/react-component-based-architecture

[64] Great Learning, "Virtual DOM," Aug. 27, 2024. [Online]. Available: https://www.mygreatlearning.com/react-js/tutorials/virtual-dom

[65] R. Purohit, "The Power of React Virtual DOM: Revolutionizing DOM Manipulation," *DhiWise,* Aug. 27, 2024. [Online]. Available: https://www.dhiwise.com/post/the-power-of-react-virtual-dom-revolutionizing-dom-manipulation

[66] A. Bijarniya, "Difference Between Imperative and Declarative in React.js," *C# Corner,* Aug. 27, 2024. [Online]. Available: https://www.c-sharpcorner.com/article/difference-between-imperative-and-declarative-in-react-js/

[67] H. Mohsin, "An Introduction To React Programming Language," i*nvozone,* Aug. 28, 2024. [Online]. Available: https://invozone.com/blog/an-introduction-to-

react-programming-language/

[68] D. Burek, "React Native comprehensive guide," *curiosum,* Aug. 28, 2024. [Online]. Available: https://curiosum.com/blog/advantages-of-react-native

[69] R. Purohit, "The Best Back-end For React Web Applications," *Medium,* Aug. 29, 2024. [Online]. Available: https://medium.com/dhiwise/the-best-back-end-for-react-web-applications-2715d3e4cf12

[70] Coding with Sami, "Choosing the Right Database for Your React.js Application," *Medium,* Aug. 29, 2024. [Online]. Available: https://medium.com/@nasaamazement321/choosing-the-right-database-for-your-react-js-application-36e632b09024

[71] Free Learning Platform For Better Future, "Pros and Cons of ReactJS," Aug. 29, 2024. [Online]. Available: https://www.javatpoint.com/pros-and-cons-of-react

[72] Flutter, "Build for any screen," Aug. 30, 2024. [Online]. Available: https://flutter.dev/

[73] D. Montano, "Why Use Flutter: Pros and Cons of Flutter App Development," *WAVERLEY,* Aug. 30, 2024. [Online]. Available: https://waverleysoftware.com/blog/why-use-flutter-pros-and-cons/

[74] Geeks For Geeks, "What is Widgets in Flutter?," Aug. 30, 2024. [Online]. Available: https://www.geeksforgeeks.org/what-is-widgets-in-flutter/

[75] Code Epsilon, "The Ultimate Showdown: Flutter Widgets vs. Traditional UI Elements," Aug. 30, 2024. [Online]. Available:

https://www.codeepsilon.com/flutter-widgets-vs-traditional-ui-elements/#:~:text=Why%20Flutter%20widgets%20are%20a%20game%2Dchanger%20in%20app%20development%3F,-Reusable%20and%20Composable&text=Widgets%20are%20designed%20to%20be,up%20development%20and%20simplifies%20maintenance.

[76] Techdynasty, "Widgets in Flutter: A Brief Overview," *Medium,* Aug. 31, 2024. [Online]. Available: https://techdynasty.medium.com/widgets-in-flutter-a-brief-overview-5f3e807449e6

[77] R. Jedynasty, "Flutter Hot Reload Feature: Benefits & Performance Explained," Aug. 31, 2024. [Online]. Available: https://www.miquido.com/blog/flutter-hot-reload/

[78] Flutter, "Hot reload," Aug. 31, 2024. [Online]. Available: https://docs.flutter.dev/tools/hot-reload

[79] Free Learning Platform For Better Future, "What is Dart Programming," Aug. 31, 2024. [Online]. Available: https://www.javatpoint.com/flutter-dart-programming

[80] O. Obinna, "How does JIT and AOT work in Dart?," *Medium,* Aug. 31, 2024. [Online]. Available: https://onuoha.medium.com/how-does-jit-and-aot-work-in-dart-cab2f31d9cb5

[81] Nehatanwar, "what is Async and Await in Flutter," *Medium,* Sep. 1, 2024. [Online]. Available https://medium.com/@nehatanwar.dev/what-is-async-and-await-in-flutter-17acdcd05b6a

[82] Flutter, "Beautiful apps for every screen," Sep. 1, 2024. [Online]. Available:

https://flutter.dev/multi-platform

[83] P. Shah, "How Does Skia Contribute as One of the Key Graphics Engines in Flutter Apps," *DhiWise,* Sep. 1, 2024. [Online]. Available: https://www.dhiwise.com/post/how-does-skia-contribute-as-graphics-engines-in-flutter-apps

[84] Jacksonmateo, "Integrating Backend Services with Flutter: Best Practices," *Medium,* Sep. 1, 2024. [Online]. Available: https://medium.com/@jacksonmateo625/integrating-backend-services-with-flutter-best-practices-aa784ea7d431

[85] L. Kosman, "Flutter Pros and Cons 2024 - Summary and Recommendations," *Flutter CTO Report 2024,* Sep. 2, 2024. [Online]. Available: https://leancode.co/blog/flutter-pros-and-cons-summary

[86] GeeksforGeeks, "Prototyping Model Software Engineering," *GeeksforGeeks*, Apr. 05, 2018. [Online]. Available: https://www.geeksforgeeks.org/software-engineering/software-engineering-prototyping-model/

[87] "Skill Levels," *iCombine*. [Online]. Available: https://icombine.net/knowledge-base/skill-levels

[88] Vet Finders, "What Skills Do You Need to Work with Animals?," *CharityJob*, Jan. 11, 2021. [Online]. Available: https://www.charityjob.co.uk/careeradvice/what-skills-do-you-need-to-work-with-animals/

[89] Visual Studio Code, "Why did we build Visual Studio Code?" Apr. 24, 2025. [Online]. Available: https://code.visualstudio.com/docs/editor/whyvscode

[90] coursera, "What Is Node.js? Understanding Its Applications" Apr. 24, 2024. [Online]. Available: https://www.coursera.org/articles/node-js?utm_medium=sem&utm_source=gg&utm_campaign=b2c_apac_x_multi_ftcof_career-academy_cx_dr_bau_gg_pmax_pr_s3_en_m_hyb_25-04_x&campaignid=22420706570&adgroupid=&device=c&keyword=&matchtype=&network=x&devicemodel=&creativeid=&assetgroupid=6566444159&targetid=&extensionid=&placement=&gad_source=1&gbraid=0AAAADdKX6Y4_1zeOHpy_QxUKHZjj4PCY&gclid=Cj0KCQjw8cHABhC-ARIsAJnY12ycfGAdBim20OICrfgZv-VDOgLP03h1SJG2Rf9S4n2sbCWUbJ_-1YUaAuudEALw_wcB

[91] MySQL, "MySQL Enterprise Edition" Apr. 24, 2024. [Online]. Available: https://www.mysql.com/products/workbench/

# APPENDIX

## A-1 POSTER

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

# FINAL YEAR PROJECT II: WEB-BASED VOLUNTEER MANAGEMENT SYSTEM

### INTRODUCTION

**What is VMS?**

A Volunteer Management System (VMS) is a platform that helps non-profit organisations to streamline volunteer recruitment, scheduling, and engagement for organisations.

### Problem Statement

**Traditional VMS**

- Poor volunteer-event matching due to outdated keyword matching systems
- Low participation rates due to poor geographical coordination
- High turnover due to low volunteer retention

### RESEARCH OBJECTIVES

- DEVELOP a dual-scale (required competencies + volunteer proficiency) weighted skill-matching algorithm to replace outdated keyword matching systems
- CREATE a distance-based matching algorithm to enhance participation by factoring in both nearby and distant events.
- Implement a gamified leaderboard system that records hours and events participation to recognise volunteer contribution and promote long-term volunteer engagement.

### WEIGHTED SKILL-MATCHING ALGORITHM

**How It Works?**

- Define skill categories and levels definition (tailored to the deployment domain).
- Organisers rate skill importance for each event (1 - 5 scale).
- Volunteers self-rate proficiency for each skill (1–5 scale).
- System calculates Skill Matching Score using:

*Skill Matching Score (%) = (Actual Skill Score)/(Maximum Possible Skill Score) × 100%*

**Why It Works?**

**Precision Matching**

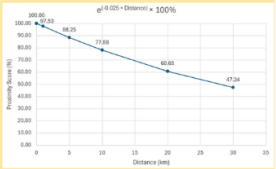
- Considers both skill importance levels and volunteer proficiency
- Replaces subjective keyword searches, ensure matches reflect actual competency

### DISTANCE-BASED MATCHING ALGORITHM

**How It Works?**

- Calculate distance between volunteer's address and event location using Google Maps Distance Matrix API.
- System calculates Proximity Score using:

$$\text{Proximity Score (\%)} = e^{-0.025 \times \text{distance (km)}} \times 100\%$$




**Why It Works?**

**Exponential Decay Function**

- Prioritise nearby events (deep score curve) while still considering slightly farther options (flat score curve).
- Increase participation rate by suggesting nearby events

### CONCLUSION

- Develops an intelligent volunteer-event matching system by integrating both algorithms into a unified score:

*Final Matching Score (%) = (Skill Matching Score × 0.5) + (Proximity Score × 0.5)*

- Ensures volunteers are both qualified for the role (can do) and able to attend conveniently (can go).

BY: CHANG WEI HANG

PROJECT SUPERVISOR: TS TAN TEIK BOON