Sabah Smart Travel Companion: AI-Based Scrapbook Generation with Integrated Itinerary Planning

By

Bong Xin Ai

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

ACKNOWLEDGEMENTS

I would like to express my heartfelt gratitude to Prof Ts Dr Liew Soung Yue, my supervisor, for his invaluable guidance, continuous support, and encouragement throughout the development of this project. His insightful feedback and deep technical knowledge have been instrumental in shaping the Sabah Smart Travel Companion: AI-Based Scrapbook Generation with Integrated Itinerary Planning, from conceptualization to implementation. I am truly thankful for the opportunity to work under his supervision and to explore the intersection of AI and travel technology.

I would also like to extend my sincere thanks to my family and close friends for their unwavering support, patience, and motivation throughout this journey. Their encouragement helped me stay focused during challenging times. Lastly, I appreciate everyone who has contributed directly or indirectly to the successful completion of this project.

COPYRIGHT STATEMENT

This Final Year Project Report is submitted in partial fulfillment of the requirements for the degree of Bachelor of Computer Science (Honours) at Universiti Tunku Abdul Rahman (UTAR). This Final Year Project Report represents the work of the author, except where due acknowledgment has been made in the text. No part of this Final Year Project Report may be reproduced, stored, or transmitted in any form or by any means, whether electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the author or UTAR, in accordance with UTAR's Intellectual Property Policy.

ABSTRACT

This project presents the development of the Sabah Smart Travel Companion: AI-Based Scrapbook Generation with Integrated Itinerary Planning, an interactive web application designed to enrich the travel experience by combining intelligent itinerary planning with automated travel documentation. Unlike existing travel platforms that emphasize only trip scheduling, this system highlights memory preservation through an AI-powered digital scrapbook. Core functionality includes automatic scrapbook page generation from user-uploaded photos and captions, supported by Large Language Models (LLMs) for image recognition, content categorization, caption generation, and dynamic layout design. Personalization is achieved through the Preference Weight Adjustment Model, which adapts scrapbook layouts based on user interactions and continuously refines future suggestions. The itinerary module functions as a streamlined, integrated support system, generating lightweight, customizable travel schedules through LLMs to complement the scrapbook experience rather than compete with it. Sabah, Malaysia, is chosen as the case study due to its cultural diversity and natural attractions, providing an ideal context for showcasing both itinerary guidance and memory documentation. This system is implemented as a web application to ensure accessibility across devices, making it practical for modern travellers. By addressing gaps in existing travel applications, this project aims to deliver a comprehensive, end-to-end solution that that enhances travel planning while preserving personal experiences through intelligent digital storytelling.

Area of Study: Artificial Intelligence (AI), Natural Language Processing (NLP)
Keywords: Scrapbook, Itinerary Planning, Image Recognition, Web Application, Personalization

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
NLP	Natural Language Processing
LLM	Large Language Model
API	Application Programming Interface

CHAPTER 1
Introduction

Malaysia's tourism sector plays a vital role in the national economy, serving as a major source of foreign exchange earnings and a catalyst for economic growth [1]. Malaysia attracts travellers with its diverse cultural heritage, natural landscapes, rich biodiversity, and vibrant local experiences. However, there is currently a lack of digital tools that not only assist tourists in planning their trips but also enhance their overall travel experience, particularly in capturing and sharing memories. This gap represents a missed opportunity to both improve visitor satisfaction and promote Malaysian tourism. Furthermore, a study by Monitor Deloitte [2] further emphasizes the potential of 'Smart Tourism' to increase Malaysia's tourism revenue from USD 25 billion to USD 110 billion by 2030, highlighting the need for innovative digital travel solutions.

In today's digital age, travellers increasingly rely on online platforms and mobile applications to simplify their travel planning. Typically, traveller begin by searching for travel information online, then compile itineraries using websites or apps. This process has become more efficient and intuitive with the integration of Artificial Intelligence (AI) technologies, particularly Large Language Models (LLMs). For instance, tools like ChatGPT can help tourists search for information and make informed decisions [3]. It can offer instant answers, personalized recommendations, and structured itineraries based on traveller's input such as budget, destination, and group size. As LLMs become increasingly embedded into chatbots on travel agency platforms and apps, travellers are likely to interact with them more often either intentionally or unintentionally [3], for trip planning support and recommendations.

Despite the advancements in trip planning applications, there remains a significant gap in the post-travel experience. While existing apps effectively assist traveller in planning itineraries, they often overlook the sentimental and reflective side of travel, the documentation and preservation of memories. After completing a trip, many travellers seek to capture meaningful moments, record their experiences, and reflect on their journeys. However, there is currently no comprehensive travel application that guides traveller from **planning a trip to concluding it** with a meaningful record of their adventures.

A well-structured travel itinerary is undoubtedly essential for maximizing a trip's efficiency and enjoyment, as it helps travellers optimize time, manage budget, navigate destinations with ease, prioritize attractions [4], However, having a digital space for post-trip reflection and storytelling is equally important, as it enhances the emotional and cultural value of travel. Traditional methods of documenting travel, such as photobooks or handwritten journals, are often time-consuming, cumbersome, and can detract from the enjoyment of the journey itself.

This project aims to develop a web-based travel application with an integrated scrapbook, providing traveller with a seamless, end-to-end travel experience from initial planning to post-trip reflection within a single platform. Unlike conventional platforms that focus solely on planning, this application will also support travellers in preserving and presenting their memories through an intelligent, automated digital photobook.

By leveraging AI technologies, the proposed system will automate the creation of a dynamic scrapbook. It will analyze and categorize user inputs such as travel preferences, photos, and notes, and generate an initial photobook layout that is both personalized and customizable. Travellers can then modify this layout based on their preferences, and these modifications will be used to refine future suggestions, thereby continuously improving personalization in alignment with the user’s preference. This intelligent automation enables travellers to effortlessly capture, organize, and preserve their travel memories without disrupting their journey, resulting in a more immersive, enjoyable, and meaningful travel experience.

Sabah, a well-known tourist destination in Malaysian Borneo, is chosen as the pilot region for this project. With its diverse natural beauty and rich cultural heritage, Sabah serves as an ideal showcase to demonstrate the app’s capabilities. The features and functionalities developed for Sabah are designed to be adaptable and scalable, enabling future expansion to other Malaysian states and other destinations.

This chapter outlines the background of the project, outlines the key problem and motivations, rese defines the research objectives, describes the scope and expected contributions, and explains the overall structure of the report.

1.1 Problem Statement and Motivation

The primary issue addressed in this project is the inadequacy of existing travel apps, which tend to prioritize itinerary planning while ignoring real-time travel documentation. Travelers often need to use multiple separate to both plan their journeys and preserve their memories, leading to a fragmented and inconvenient experience. Moreover, current scrapbook apps generally static and lack intelligent personalization. They rarely consider users' preference, and provide limited support for dynamic, customizable layouts. The manual effort required to arrange photos, write captions, and organize content is time-consuming and unengaging, especially during or immediately after a trip. Consequently, many travellers abandon the idea of documenting their trips altogether, missing the opportunity to create meaningful and lasting records of their journeys.

The motivation behind this project is to bridge the gap between trip planning and post-travel reflection through the development of an integrated AI-driven platform. While a lightweight itinerary planner is included to guide travellers with structured suggestions, the primary focus lies in automated travel documentation. By leveraging Large Language Models (LLMs), the system will transform user-generated content—such as photos, notes, and captions into customizable digital photobooks. As travellers edit and refine these layouts, the system learns from their preferences to improve future scrapbook generation. Ultimately, the project aims to enrich the overall travel experience, particularly for tourists visiting Sabah, by unifying simple itinerary support with dynamic and personalized memory preservation.

1.2 Research Objectives

The main objective of this project is to develop an end-to-end AI-powered travel companion system that enables users to seamlessly plan, document, and preserve their journeys, with the AI-based scrapbook as the central feature. The system aims to provide a unified platform where travellers can not only generate intelligent digital scrapbooks but also access complementary itinerary planning, personalization, and editing features, ensuring a complete and user-centric travel experience.

To achieve this goal, the project defines the following sub-objectives:

1. **AI-Powered Scrapbook Generation**
    - Automate the transformation of user-uploaded photos, captions, and travel notes into structured digital scrapbook pages.
    - Use Large Language Models (LLMs) for image recognition, content categorization, caption generation, and layout design.
2. **Personalization and Preference Learning**
    - Implement a **Preference Weight Adjustment Model** that learns from user edits and applies these preferences to future scrapbook generations.
    - Enhance personalization so that the scrapbook evolves to reflect each traveller's unique style and documentation habits.
3. **Lightweight Itinerary Planning**
    - Provide an integrated but streamlined itinerary planner that leverages LLMs to generate **personalized pre-itineraries**.
    - Support inputs such as travel style, budget, and duration, ensuring flexible and editable travel schedules that complement scrapbook documentation.
4. **Seamless and Accessible User Experience**
    - Ensure the web application is mobile-responsive, intuitive, and accessible across devices.
    - Support offline draft saving and efficient data handling for large image uploads.

Through these objectives, the project delivers a solution where itinerary planning seamlessly supports the documentation process, while personalization ensures that each scrapbook authentically reflects the traveller's unique journey—creating a streamlined end-to-end platform that integrates both trip planning and memory preservation.

1.3 Project Scope and Direction

This project, Sabah Smart Travel Companion: AI-Based Scrapbook Generation with Integrated Itinerary Planning, is designed to enrich the travel experience by emphasizing the documentation and preservation of personal journeys, with a special

focus on promoting Sabah as a culturally and ecologically diverse destination. The final deliverable will be a web-based application that combines intelligent itinerary planning with an automated, personalized travel documentation system. While itinerary support is included to assist travellers in planning trips based on travel styles, interests, budgets, and durations, the primary focus lies in capturing and creatively showcasing the unique stories and moments from each journey. The goal is to deliver a comprehensive, user-friendly platform that not only simplifies trip planning but also elevates the way travellers reflect on and share their experiences, especially those exploring Sabah's rich cultural and natural landscapes.

The project will utilize a Scrapbook Engine that incorporates LLM to analyze user inputs such as photos, captions, and preferences, and transform them into structured JSON used to generate photobook pages in formats like grids and timelines. Users can personalize their digital photobooks by repositioning photos, editing or requesting AI-generated captions. These user interactions will feed back into the system, allowing it to learn and improve future scrapbook generation, creating an increasingly personalized experience over time. By combining preference weight algorithms with user feedback (user edit actions), the application will deliver a responsive, user-centric solution designed to work seamlessly across both desktop and mobile devices. This ensures that travellers can effortlessly plan and preserve their travel journeys in Sabah, whether at home or on the move, enjoying a consistent and engaging experience throughout their journey.

1.4 Contributions

This project advances smart tourism by delivering a web-based platform that integrates lightweight itinerary support with AI-driven travel documentation. Its primary innovation lies in the scrapbook engine, which employs Large Language Models (LLMs) for photo categorization, caption generation, and dynamic layout construction. The system enables automatic creation of digital photobooks that adapt to user edits, thereby improving personalization over time through iterative learning.

By prioritizing intelligent scrapbook generation over static documentation, the project reduces manual effort while enhancing user engagement and cultural storytelling. Unlike existing travel applications that emphasize scheduling or

journaling, this platform introduces adaptive, travel-style-based layouts and editable AI-generated content. The result is a streamlined, end-to-end solution that not only assists in trip organization but also enriches post-travel reflection, contributing to both tourism technology and AI-assisted creative systems.

1.5 Report Organization

This report is structured into seven chapters, each addressing a key component of the research and development process. Chapter 2 presents the literature review, including relevant algorithms, existing studies, and a comparative analysis of current travel and scrapbook applications against the proposed system. Chapter 3 explains the methodology and approach, including the development model, tools, technologies, and performance criteria guiding the project. Chapter 4 presents the overall system design, covering architectural flow, module interactions, and interface structures. Chapter 5 details the system implementation, describing the integration of core functionalities and technical configurations. Chapter 6 evaluates the system through testing and discussion of results, assessing its effectiveness and limitations. Finally, Chapter 7 concludes the report by summarizing key contributions and proposing directions for future work.

CHAPTER 2
Literature Reviews

The intersection of smart tourism, AI, and digital storytelling has opened new avenues for enhancing traveller experiences. While traditional travel platforms focus on logistics and static information, recent advances enable deeper personalization, memory preservation, and interactive planning. This literature review synthesizes key domains: smart tourism technologies, AI-driven planning, multimedia storytelling, travel memory documentation, traveller personalization, hybrid content generation, and UX/UI design to position the foundation of this proposed solution: Sabah Smart Travel Companion: AI-Based Scrapbook Generation with Integrated Itinerary Planning.

To begin, smart tourism leverages advanced Information and Communication Technologies (ICT) to enhance the quality of travel services and experiences. Core components include the Internet of Things (IoT), ubiquitous sensors, immersive media (AR/VR), AI-powered analytics, and mobile digital platforms [5]. These technologies enable real-time navigation, contextual information delivery, and co-creation of travel content through apps and social media platforms [5][6]

Despite the availability of these tools, existing smart-tourism systems often emphasize infrastructure or static guides, neglecting the personal and reflective dimensions of travel. Importantly, few applications facilitate dynamic user storytelling or support diary-keeping in meaningful ways. This project addresses this gap by combining intelligent itinerary planning with a personalized scrapbook, fostering both functional convenience and emotional engagement.

At the core of this advancement lies AI, particularly Large Language Models (LLMs), which are revolutionizing travel planning. Unlike traditional platforms that rely on static databases and rule-based queries (e.g., Expedia), LLMs like GPT-4 can generate natural-language recommendations and narratives. However, LLMs often produce itineraries that are incoherent or violate user constraints. Studies such as TravelAgent highlight the need for hybrid systems [7]. These combine LLMs with structured planners and tool modules to ensure validity, personalization, and user satisfaction.

Building on this, this project integrates LLM-generated content for travel suggestions and narratives but enhances practicality through structured constraint checking. A key innovation is the incorporation of a "memory" module—an interactive scrapbook that evolves with user preferences and past experiences. This allows future recommendations to be deeply personalized and grounded in context, mitigating common weaknesses of generic AI systems.

Beyond planning, tourism is not merely transactional, it is experiential. Research shows that multimedia storytelling, including photos, audio, video, and AR, significantly boosts engagement, emotional connection, and information retention among travelers [8][9].

Yet, despite these proven benefits, most tourism applications only offer fragmented media sharing without coherent story structures or personalization. This proposed solution treats the scrapbook as a dynamic multimedia storybook. As users explore Sabah, they are prompted to document their journey through integrated text, and images, curating a living photobook. This narrative-centered design aims to heighten engagement and create a lasting emotional connection with the destination.

Furthermore, the act of documenting one's journey through journaling, photobooking, or photo logs has long been associated with enhanced memory retention and emotional satisfaction. Studies indicate that combining personal media with reflective prompts (e.g., "What was your favorite moment today?") helps travellers relive and express meaningful moments [10][11].

In contrast, while contemporary social media platforms facilitate rapid sharing of travel experiences, the resulting content is often fragmented, superficial, and lacks contextual depth. This project addresses the limitation by introducing a structured photobook system supported by AI-generated captions and adaptive layouts. The approach transforms unstructured travel media into coherent narratives that preserve both factual and emotional dimensions of the journey, thereby enhancing long-term memory retention and reflective value.

Achieving the customization requires a content generation strategy. Creating well-structured scrapbook requires thoughtful content layout and generation mechanisms. Traditional rule-based systems use rigid templates, which often fail to accommodate the variability of user experiences. Conversely, pure generative AI may lack coherence or structure [7]. A hybrid approach is therefore ideal.

Lastly, user experience design plays a critical role in adoption. Effective UI design is central to ensuring accessibility and satisfaction. A responsive, mobile-friendly interface with intuitive features can significantly enhance user engagement. Research in co-design and interactive storytelling highlights expectations such as drag-and-drop content management, memory prompts, timeline views, and social sharing capabilities [10].

In response to these needs, the project adopts these principles by offering a streamlined interface where users can add multimedia content with minimal friction. The system supports both online and offline use, with autosave and caching for travel in remote areas. By aligning with best practices in digital journaling and mobile interaction design, we ensure that the experience remains both powerful and user-friendly.

In sum, this literature review demonstrates the convergence of smart tourism, AI planning, digital storytelling, and personalized UX design as foundational to the next generation of travel applications. Current systems fall short by treating travel as either a logistical or media-sharing problem. This project fills this gap by integrating intelligent itinerary generation with a reflective, multimedia scrapbook—tailored to the traveller's identity and evolving preferences. This dual emphasis on function and emotion represents a novel approach in the smart tourism landscape.

2.1 Review of Technologies

2.1.1 Hardware Platform

Mobile travel applications can be deployed on various hardware platforms, ranging from Android and iOS smartphones to desktop devices. In some cases, applications may rely on physical servers or high-performance desktop machines to handle intensive tasks such as image recognition or itinerary optimization. While these approaches ensure high processing capability, they also increase costs and limit accessibility for general users.

For this project, the hardware platform is focused on consumer-level devices, particularly smartphones and web browsers, which provide accessibility for the travellers. The application is designed to run efficiently on mid-range hardware, ensuring smooth performance during scrapbook rendering and itinerary planning.

2.1.2 Firmware/OS

Travel applications can be developed natively for Android using Java/Kotlin or for iOS using Swift, which ensures high performance and full access to device-level features. However, native development requires maintaining separate codebases for each platform, increasing complexity and effort. Cross-platform frameworks like Flutter or React Native reduce this overhead by using a single codebase, but often introduce trade-offs in performance or platform-specific optimizations.

For this project, a web-based, mobile-responsive approach using React and Tailwind CSS was adopted. This eliminates the need for platform-dependent builds, ensuring compatibility across Windows, macOS, Android, and iOS via browser access. While native solutions offer deeper integration, the browser-based design achieves the broader goal of accessibility and ease of deployment without the maintenance burden of multiple native applications.

2.1.3 Database

Database selection in travel applications typically depends on balancing structured storage, scalability, and flexibility. Relational systems such as MySQL or PostgreSQL excel at handling structured, transaction-heavy data, while NoSQL systems like MongoDB are better suited for dynamic, user-generated content. Cloud

solutions like Firebase combine real-time synchronization with integrated services, making them attractive for rapid prototyping.

In this project, a dual-storage strategy was implemented. Local Storage ensures offline persistence for scrapbook drafts and itineraries, a feature that relational databases alone cannot provide effectively. Firebase complements this by enabling synchronization across devices. This hybrid model was preferred over fully relational systems, as it provides both the flexibility required for multimedia scrapbook data and the convenience of real-time updates without complex backend setup.

2.1.4 Programming Language

Programming languages are selected based on target platforms and system requirements. Native mobile applications typically rely on Swift or Kotlin, while backend systems often adopt Python or Java for performance-intensive workloads. JavaScript and its superset TypeScript, however, dominate web application development due to their versatility, ecosystem, and strong integration with modern frameworks.

For this project, TypeScript with React was chosen for frontend development, combining type safety with efficient, scalable UI construction. This decision aligns with the web-based deployment model discussed in Section 2.1.2, ensuring consistency across platforms. Node.js and JavaScript support backend logic and API integration with services such as Gemini API and Firebase, offering lightweight yet robust connectivity. Tailwind CSS complements the stack by enabling responsive, consistent layouts. Together, these technologies provide a cohesive development environment that prioritizes maintainability, accessibility, and rapid iteration, in contrast to the higher complexity of maintaining separate native stacks.

2.1.5 Algorithm: Training User Preferences

Our system learns each user's layout and text-style preferences through an *incremental weighting* approach. We maintain a weight for each possible layout, font family, and font color. Initially these weights are set to default values (often uniform or heuristically seeded by domain knowledge). When the user saves a scrapbook, the system treats the final chosen layout and text styles as implicit positive feedback. In effect, each saved choice "trains" the user profile by increasing the weight of the selected option. This is analogous to content-based recommendation: the user's profile is represented by weights on item features (e.g. layout types, font attributes), and each interaction updates that profile [12]. Over time the weights converge to a stable probability distribution reflecting the user's preferences [13].

- Weighted sampling of layouts: The layout generator picks a template by weighted random selection, sometimes it is called "roulette-wheel" selection. Each layout option has a weight, and a layout is chosen at random in proportion to its weight. This ensures more preferred layouts are more likely to be used, while still allowing exploration. For example, Zyxware explains that "roulette wheel selection that randomly selects an item based on its weight, giving each item a fair chance, while still considering its relevance to the user" [14]. After selection, the chosen layout is shown as the default template. If the user does not change it, the system treats that as implicit approval; if the user switches to a different layout, that change is logged as explicit feedback.

- Logging and implicit feedback: Each time a scrapbook is saved, we log the *final* state – including the layout type and any text choices (font and color) the user made. This final selection serves as the feedback signal. This is similar to the approach in adaptive recommenders where the "favourite item" selected by the user is used as feedback to update the user model [13]. Here, the layout displayed by default vs. the layout chosen by the user allow the system to infer which template the user truly prefers.

- Updating weights: Upon save, we reward the chosen option by incrementing its weight (for example, adding 1 to that layout's count). If the user changed the default layout to a new one, the new layout's weight is increased; if the user kept the default, we reward that layout's weight. This simple update rule biases

the probability distribution toward the user's favoured choices. (Over many interactions, the weights can be normalized to probabilities so that the sum remains 1.) In practice, this is a lightweight form of reinforcement learning: the final "reward" is the saved choice, and the system updates its policy (weights) accordingly. Similar weighted-update schemes have been studied in the literature – for example, Zhang et al. describe an algorithm that adjusts attribute weights by analyzing which items a user selects as favourites [13]. By continuously adjusting the weights based on each interaction, the system tailors itself to the user's long-term preferences.

- Font and color preferences per category: In parallel with layouts, we also adapt font-family and font-color choices. Photos are categorized (e.g. Nature, Food, People, etc.), and for each category the system maintains separate weight tables for font families and for font colors. These tables are initialized with domain-specific defaults and minimal weights (so that even untried fonts/colors have a nonzero chance). When text is added to a scrapbook, the font and color are chosen by weighted sampling from the category's current weights (again using roulette-wheel selection). If the user edits the text style, the final font and color become the "chosen" items to reward. The system logs the final font and color, then increments the corresponding weights for that category.

- Ensuring all options remain selectable: To avoid permanently ruling out any option, we keep a minimal positive weight on every allowed font and color (even those never chosen), similar to Laplace smoothing. This follows the pattern in personalization systems of handling "cold start" or new items. By ensuring every option has at least a small weight, the weighted random selection can still occasionally pick a new font or color, allowing user exploration and preventing zero-probability cases.

- Outcome: The result of this training flow is a user-specific probability model for layouts and for text styles per category. At any point, we can compute the probability of each layout or font/color by normalizing the weights. The system then uses these probabilities to generate new scrapbooks: it tends to present layouts and styles that the user has historically preferred, while still leaving room for variety. In user interface research, such adaptive interfaces – which

learn from user behavior to adjust layout and styling – have been shown to enhance user experience by aligning the interface with individual preference.

Each element of this algorithm draws on well-known personalization principles. Content-based recommendation methods maintain a user profile of preferred item features (here, layouts/font family/font colors) derived from past actions [12]. The weighted update mechanism is a simple but effective variant of that idea: every user action biases the profile toward chosen attributes [13]. In sum, our implementation applies these ideas in the context of a scrapbook editor: layouts and typographic styles are treated like "items" with features, and user saves continually refine the weight-based profile that guides future suggestions.

Our approach is inspired by standard recommender-system techniques for modelling and updating user preference profiles [13]. The use of weighted random (roulette-wheel) selection is a common strategy in adaptive sampling [14], and interface adaptation via machine learning has been explored in recent HCI research [15]. Furthermore, empirical studies confirm that users appreciate personalized recommendations (e.g. tailored layouts or styling) as it helps them find desired options more quickly.

2.1.6 Summary of the Technologies Review

In summary, traditional travel applications often depend on rigid native development frameworks, relational databases, and rule-based personalization methods. While such technologies ensure stability and predictable performance, they demand extensive development resources and struggle to provide seamless adaptability across platforms. Moreover, rule-based systems typically lack the capacity to evolve with user behaviour, resulting in generic outputs that limit personalization.

This project instead employs a lightweight, adaptive stack built around React, TypeScript, and Tailwind CSS for responsive frontend development, supported by Local Storage and Firebase for persistence and synchronization, and the Gemini API for AI-driven functionality. Personalization is achieved through an incremental weighting algorithm that models user preferences for layouts, fonts, and colors using weighted sampling and reinforcement-style updates. This enables the system to adapt scrapbook generation dynamically, reflecting user-specific stylistic tendencies while preserving variety. Compared to conventional approaches, this architecture emphasizes scalability, accessibility, and cultural adaptability, aligning with the project's goal of delivering a personalized scrapbook experience tailored to Sabah tourism.

2.2 Review of the Travel Application and Website

This subsection reviews various travel apps, focusing on their strengths, weaknesses, and possible improvements. The evaluation includes popular apps like Wanderlog, Trip.com, and Travel Malaysia, which provide a range of travel-related services, such as itinerary planning, booking, and destination discovery.

2.2.1 Wanderlog

Wanderlog is a popular travel planning app that supports itinerary creation, expense tracking, and real-time collaboration with live syncing and editing, making it ideal for road trips and group travel. The app includes expense tracking and cost splitting, making it especially useful for group trips. It connects to email to import reservations and integrates with Google Maps for route optimization. Wanderlog also provides smart recommendations based on the itinerary to help users discover new places. The Wanderlog has the following strengths, which are stated below:

- **Collaborative Tools:** The real-time collaboration feature sets it apart from other travel planners, multiple users can edit the same trip simultaneously and making group travel coordination much easier.
- **Comprehensive Planning**: The app covers everything from itinerary building to budget tracking, and even provides recommendations, making it an all-in-one tool
- **Strong Map Integration:** The ability to visualize your trip on a map helps optimize routes and plan efficiently.
- **Customizable Lists:** You can create lists of places you want to visit, even if they’re not part of your daily plans.
- **AI Assistant:** Wanderlog offers an AI assistant to help streamline your planning process.

The Wanderlog has the following weakness, which are stated below:

- **Limited Offline Features (Free Version):** Some key features like offline access and unlimited attachments are locked behind a premium subscription.

- **Performance Issues**: Some users have reported occasional glitches or slow performance, especially with complex itineraries.
- **Overwhelming for Casual Users:** The wide array of features can be overwhelming for travelers who prefer a simpler or more spontaneous approach to trip planning

The ways of solving the weakness are stated below:

- **Expanding Offline Capabilities**:
  Offer basic offline features like viewing itineraries and maps to all users; allow premium users to edit and download full itineraries offline
- **Simplifying the User Interface for Casual Users**:
  Add a "basic mode" focusing on core features for casual users, with the option to switch to full mode as needed because the current interface overwhelming users with too many options.

2.2.2 Trip.com

Trip.com is a comprehensive travel app that offers a seamless experience for travelers looking to explore Malaysia. The app provides a wide range of services, including hotel bookings, flight reservations, car rentals, and destination guides. It's designed to be a one-stop solution for all your travel needs, making it easy to discover and plan your trip to Malaysia. The app's user-friendly interface and extensive database of travel-related information ensure that you can find everything you need to make your trip enjoyable and hassle-free.

Trip.com has the following strengths, which are stated below:

- **Comprehensive Travel Services**

  Trip.com offers an all-in-one platform where users can book flights, hotels, car rentals, and more. This makes it convenient for travelers to plan and manage their entire trip from a single app.

- **Personalized Recommendations**

The app provides personalized recommendations based on your location, preferences, and past bookings. This feature helps users discover relevant attractions, accommodations, and activities tailored to their interests.

- **User-Friendly Interface**

  The app is designed with a clean and intuitive interface, making it easy for users to navigate through various options and find the information they need quickly.

- **Real-Time Updates and Alerts**

  Trip.com offers real-time updates on flight statuses, hotel bookings, and other travel-related information. This helps users stay informed and make adjustments to their plans if needed.

Trip.com has the following weakness, which are stated below:

- **Limited Activity Filtering**

  The app currently offers limited filtering options for activities, focusing mainly on states or regions rather than specific types of activities, such as adventure, relaxation, or cultural experiences. This makes it harder for users to find activities that align with their specific interests.

- **Lack of Itinerary Planning Features**

  While the app provides information on attractions and activities, it lacks robust itinerary planning tools. Users cannot easily organize their day-to-day plans or integrate different elements of their trip into a cohesive schedule.

- **Overloaded Information on Single Pages**

  The app's interface sometimes combines too much information on a single page, making it difficult for users to navigate and explore details efficiently. This can lead to a cluttered experience and overwhelm users who are trying to find specific information.

The ways of solving the weakness are stated below:

- **Enhanced Filtering Options**

  Introduce more detailed filtering options for activities, allowing users to sort by categories such as adventure, relaxation, family-friendly, or cultural experiences. This will help users find activities that match their preferences more easily.

- **Integrated Itinerary Planner**

  Add a built-in itinerary planner that allows users to create and manage their daily schedules. The planner could automatically suggest activities, restaurants, and attractions based on the user's preferences and bookings, making trip planning more streamlined and personalized.

- **Improved Interface Organization**

  Redesign the interface to better organize information across multiple pages or sections. Implement collapsible menus or tabs to separate different types of content, making it easier for users to access the information they need without feeling overwhelmed by excessive details on a single page.

2.2.3 Travel Malaysia

The Digital Brochure Apps for Malaysia offer a modern approach to travel planning by providing a curated selection of the best attractions and activities across the country. Designed to enhance the travel experience, these apps allow users to discover fantastic things to do in Malaysia based on their current location or planned destinations. From pristine beaches to rich cultural heritage sites, these apps provide an array of options to explore Malaysia's unique attractions. With user-friendly features like offline viewing, location-based suggestions, and customized saved places, these apps aim to make exploring Malaysia more convenient and enjoyable.

Travel Malaysia has the following strengths, which are stated below:

- **Expertly Curated Content**
  The apps feature unique, hand-picked lists of top attractions and activities, which ensures that users receive high-quality recommendations tailored to their travel interests.

- **Location-Based Discover**
  Users can easily find nearby attractions and activities, making it convenient to explore the best that Malaysia has to offer, no matter where they are.

- **Free and Accessible**
  The apps are completely free of charge, with no hidden costs, making them accessible to all users.

Travel Malaysia has the following weakness, which are stated below:

- **Limited Activity Filtering**
  The apps only allow filtering by state, which can be restrictive for users looking for specific types of activities, such as adventure sports, cultural experiences, or family-friendly outings.
- **Lack of Itinerary Planning Features**
  The apps primarily provide information but lack the functionality to help users plan their itineraries, which limits their utility for comprehensive trip planning.
- **Unfriendly User Interface**

The user interface is cluttered, with all information combined on one page, making it difficult for users to navigate and explore the available content efficiently.

There are several ways to solve the weaknesses, which are

- **Enhanced Filtering Options**
  Introduce more detailed filtering options that allow users to sort activities by category, such as nature, adventure, culture, or family-friendly activities. This would help users find exactly what they're interested in more quickly and efficiently.
- **Integrated Itinerary Planning**
  Incorporate itinerary planning features within the apps, allowing users to select and organize their chosen activities into a cohesive travel plan. This could include suggested routes, estimated travel times, and the ability to customize daily schedules.

- **User Interface Redesign**
  Redesign the user interface to be more intuitive and user-friendly. Organize the content into clearly defined sections with easy navigation tabs or a search function, enabling users to explore information without feeling overwhelmed.

2.3 Review on travel diaries apps

2.3.1 Traveldiariesapp

Traveldiariesapp is a digital platform designed to cater to travel enthusiasts who wish to document and share their travel experiences. The application combines the functionalities of a traditional scrapbook with modern blogging capabilities, offering users the tools to create visually appealing and content-rich travel blogs [17]. It allows users to record their journeys, upload photos, write descriptions, and organize their content into well-structured entries that can be easily shared with others. Traveldiariesapp has the following strengths, which are stated below:

- **Customization Options:**

  Users can personalize their travel diaries with various fonts, styles, and layouts, allowing for a highly personalized and aesthetically pleasing final product.

- **Integration of Photos and Maps:**

  The ability to add photos and maps, with options to mark locations and routes, enhances the storytelling experience and makes the diary more engaging and informative.

- **Cross-Device Compatibility:**

  Users can seamlessly switch between devices, whether working on a computer, tablet, or smartphone, providing flexibility and convenience.

Traveldiariesapp has the following weakness, which are stated below:

- **Restricted Layout Options:**

  While the app offers customization, the range of layouts and templates might feel limited to users who are looking for more diverse design options.

- **Basic Photo Editing Tools:**

  The photo editing features might be too basic for users who want to make more advanced adjustments to their images before including them in their diaries.

- **Limited Social Integration:**

  The app's sharing features are relatively basic, focusing mainly on sharing via direct links only.

The ways of solving the weakness are stated below:

- **More Template and Design Options**

  Offer a wider variety of templates and layouts, possibly through an online marketplace or community where users can share or purchase designs.

- **Advanced Photo Editing Tools**

  Integrate more advanced photo editing tools, such as filters, cropping, and color adjustments, to give users more control over the images in their diaries.

- **Improved Social Media Integration**

  Expand sharing options like popular social media platforms such as Instagram, Facebook, and Twitter. This would allow users to share their travel experiences more broadly and with more engagement.

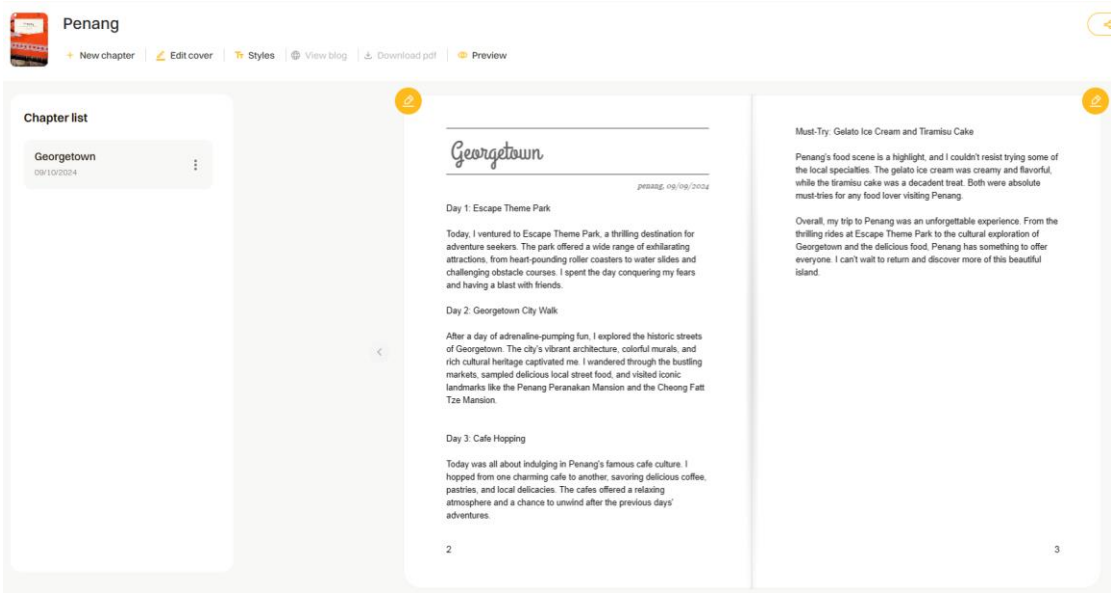

Figure 2.1 Traveldairiesapp

2.3.2 Time Diary

Time Diary is an application designed to seamlessly record activities, experiences, thoughts, and ideas throughout the day. It offers users the convenience of syncing data across all their devices, making it easy to maintain and organize diaries on the go [16]. With powerful features for organizing and customizing diary entries, Time Diary aims to simplify the process of preserving memories and experiences.

Time Diary has the following strength, which are stated below:

- **Auto-Generated Photobook**

  Users can effortlessly create photobooks by simply inputting photos. The app handles the rest, generating a polished photobook automatically.

- **Variety of Templates**

  Time Diary offers numerous templates for users to choose from, allowing them to personalize their photobooks according to their preferences.

- **Customization Options**

  The app allows users to add elements like stickers, edit photo sizes, change templates, edit text, and even draw directly on their photobooks.

- **Photo Editor**

  Integrated photo editing tools enable users to enhance and modify their images before adding them to their diaries or photobooks.

Time Diary has the following weakness, which are stated below:

- **Limited Templates in Free Version**

  The free version only offers a limited number of templates, which restrict users' ability to fully customize their photobooks without upgrading to a paid version.

Figure 2.2 Time diary photobook customization (can add photos, choose template, edit background and text, add stickers)

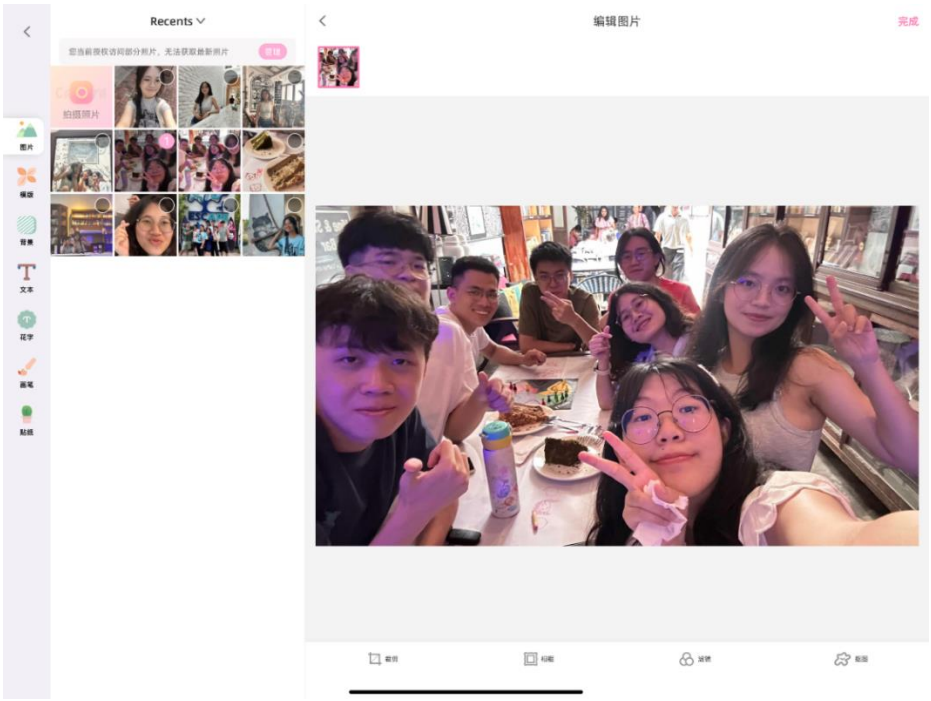

Figure 2.3 Time diary photo editor

Figure 2.4 Time diary

2.3.3 Journi Blog

Journi Blog is a scrapbook and photo book creation app designed to help users effortlessly capture their adventures, track their daily life, and preserve memories in beautifully crafted photo books. The app allows users to record their journeys with photos, notes, and maps, all while offering the convenience of offline functionality, collaboration with friends, and syncing across devices. With Journi, travelers can easily document their experiences and share them with loved ones back home.

Journi Blog has the following strength, which are stated below:

- **Automatic Timeline Creation**

  Journi's ability to automatically group photos by time and location data simplifies the process of creating a scrapbook, making it easy for users to relive their journeys without manual effort.

- **High-Quality Image Storage**

  The app offers the ability to save photos in high resolution, ensuring that users can preserve their memories in the best possible quality, which is especially valuable for creating photo books.

- **Collaboration and Sharing**

  Journi's collaborative features allow users to create shared journals with friends, making it easy to gather the best images for photo books and share the journey with others.

Journi Blog has the following weakness, which are stated below:

- **Subscription-Based Features**

  Many of Journi's most valuable features, such as high-resolution uploads, offline functionality, and cloud backup, are only available with a premium subscription, which may deter users who prefer free apps.

- **Limited Customization in Free Version**

The free version of the app may lack customization options, potentially leading to a less personalized experience for users who do not opt for the premium subscription.

- **No Photo Editing Tools:**

  There is no more advanced adjustments to user's images before including them in their diaries.

There are several solutions to solve weaknesses, which are

- **Enhanced Free Features**

  Offering more customization options and higher resolution uploads in the free version could increase user engagement and satisfaction, potentially converting more users to premium subscribers in the long run.

- **Flexible Subscription Plans**

  Introducing more affordable or flexible subscription plans could make premium features more accessible, attracting a broader user base and reducing the deterrent effect of the subscription model.

- **Integrate Photo Editing Tools**

  Integrate photo editing tools, such as filters, cropping, and color adjustments, to give users more control over the images they include in their diaries.

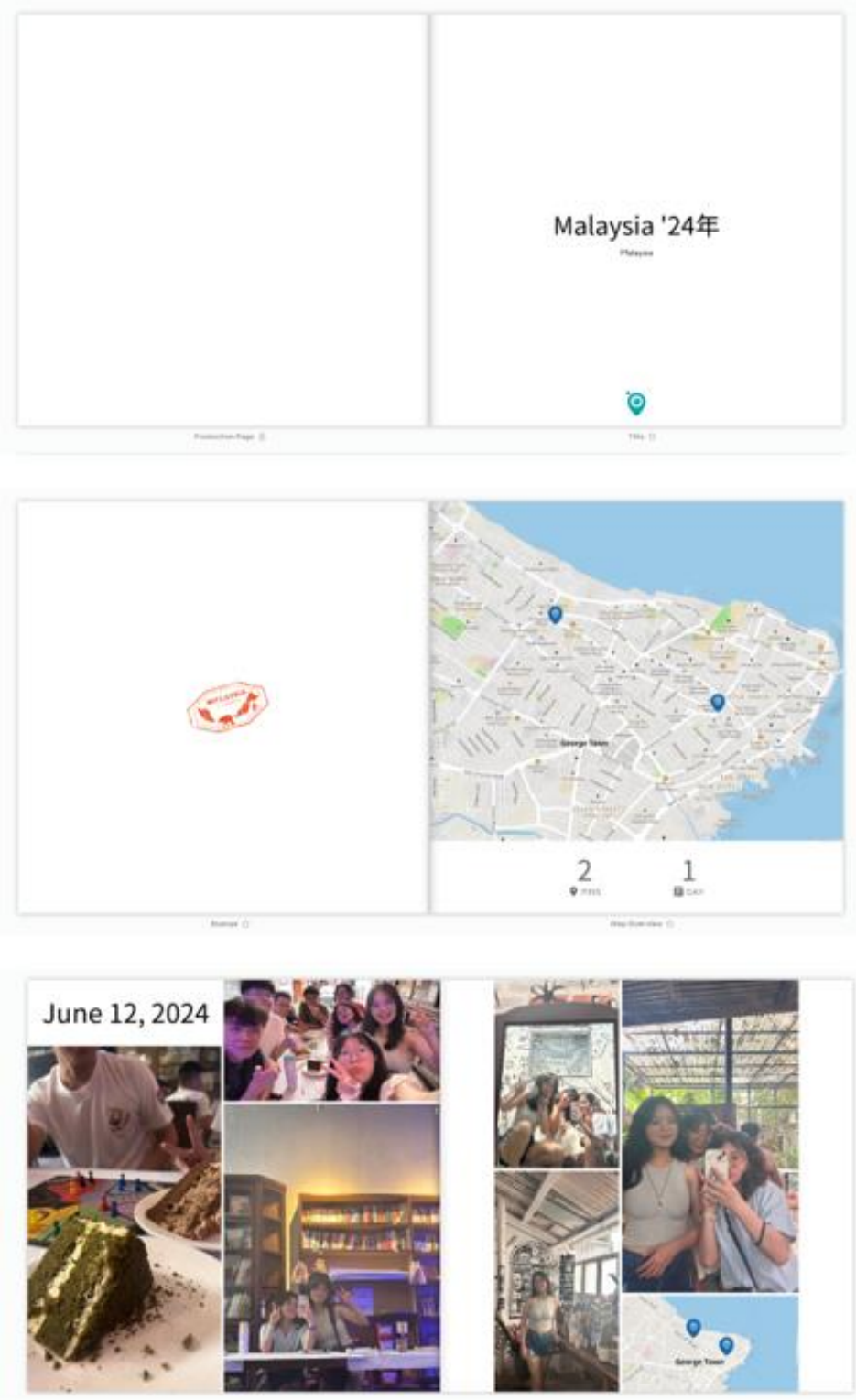

Figure 2.5 Journi Blog

2.6 Journi Blog Timeline

2.4 Comparison between Application Reviews with Proposed Application

Features	Wanderlog	Trip.com	Travel Malaysia	Propose System
Create itinerary	Auto generate	Auto generate	No	Auto generate
Customize Itinerary	Yes	No	No	Yes
Integrated Scrapbook	No	No	No	Yes
AI Assistant	Yes	No	No	Yes

Table 2.1: Travel App Functions Comparison

Features	Traveldiariesapp	Time Diary	Journi Blog	Propose System
Generate photobook	Manually	Auto	Auto	Auto
Customize photobook	Yes	Yes	No	Yes
Photo editor	Basic	Yes	No	Yes
Generated caption	Manually type	Manually type	Manually type	Auto generate/ Manually type
Dynamic template & layout	No	Provided	Provided	Provided
Image recognition and categorization	No	No	No	Yes
AI Assistant	No	No	No	Yes

Table 2.2: Scrapbook functions Comparison

2.5 Proposed System

The proposed system, Sabah Smart Travel Companion: AI-Based Scrapbook Generation with Integrated Itinerary Planning, offers an all-in-one platform that simplifies travel planning and creatively documents user experiences. It focuses on two key areas: itinerary creation and scrapbook generation.

In the planning phase, users can view list of Sabah places, foods and activities. Then, user can create personalized travel itineraries in the system by entering details such as destination, travel dates, preferences, and budget. The system will auto-generate suggested plans that users can further customize, adding or removing activities, adjusting travel routes, and selecting accommodations.

The post-travel phase is supported through the integrated scrapbook feature, which allows users to compile and relive their memories in the form of a dynamic digital photobook. Users upload their photos, and the system uses AI to recognize and categorize images. Captions and descriptions can be auto generated or manually entered. Dynamic templates and layouts are provided for a unique layout per trip.

To further enhance personalization, the system includes a built-in photo editor. This allows users to adjust their images by using editing tools such as cropping, rotating, position changing of image, and applying filters. Different font family, size, colour and weight can be adjusted for the captions. Once the photos and captions are finalized, the system composes the photobook in a visually coherent format, enabling users to experience their journey as a story. The changes of users will also be saved to analyse for user preferences. A weight preference model is implemented to adjust recommendations and scrapbook customization based on each user's interaction history, ensuring a more personalized experience over time.

Compared to existing travel applications that typically focus on either itinerary planning or journaling, the Sabah Smart Travel Companion provides a more holistic and integrated experience. While many current solutions lack features like automated scrapbook generation and flexible layout customization, this system addresses those gaps by offering personalized scrapbooks, preference-based adjustments, and customizable content. Together, these features create a richer and more engaging experience than conventional travel apps.

CHAPTER 3
System Methodology/Approach

3.1 Design Specification

This section outlines the design specifications of the Sabah Smart Travel Companion, illustrating how the system's components interact and support the overall travel experience. It includes visual representations and descriptions of the system architecture, use cases, and activity flows, providing a clear overview of the platform's functionality. The diagrams and flowcharts demonstrate the sequence of operations from itinerary planning to scrapbook generation, highlighting key user interactions, system processes, and data flow within the application.

3.1.1 System Architecture Diagram

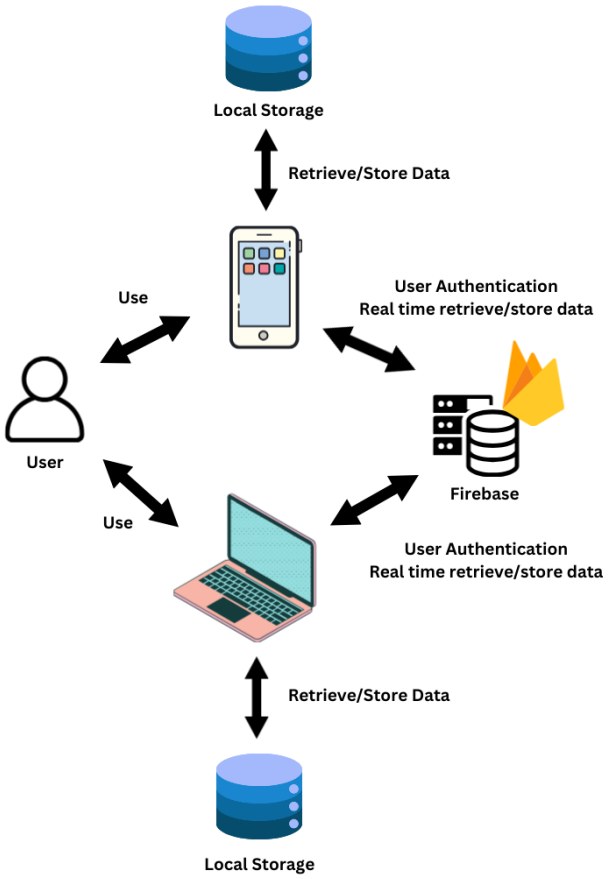

Figure 3.1: System Architecture Diagram

The Sabah Smart Travel Companion adopts a hybrid architecture that combines device-based local storage with cloud-based Firebase services, enabling an end-to-end workflow for itinerary planning and scrapbook generation. Users interact with the system through either smartphones or laptops, depending on the travel phase. During pre-trip planning, larger screens and keyboards support detailed itinerary creation, while mobile devices become essential during the trip for capturing photos, accessing real-time updates, and making quick adjustments. The frontend is implemented with React, ensuring a responsive interface across both platforms.

Local storage is utilized as a temporary workspace for photo handling and scrapbook editing. When users upload images, they are stored directly on the device to allow immediate access, smooth performance, and reduced dependency on network connectivity. This design supports rapid editing operations such as captioning, date entry, and layout arrangement. Once a scrapbook or itinerary is finalized, users can export the content as a high-quality PNG file, which is saved locally on their device for permanent retention. This approach preserves personal content under the user's control, while also optimizing the responsiveness of editing tasks.

Firebase complements this local-first design by providing persistent, cloud-based services for authentication and real-time data synchronization. Through secure login, users can maintain personalized profiles across devices. The Firebase real-time database manages structured data such as itinerary records, scrapbook metadata, and learned user preferences. These preferences stored as incremental weight updates for layouts, fonts, and colors, ensure that personalization persists across sessions and devices. By combining local photo management with centralized preference storage, the system balances flexibility with continuity.

In summary, the architecture leverages local storage for responsive, device-bound editing and export, while Firebase ensures long-term data consistency, cross-device availability, and secure identity management. This hybrid integration delivers a unified platform where travellers can seamlessly plan, personalize, and document their journeys, aligning technical efficiency with user-centred design.

3.1.2 Use Case Diagram and Description

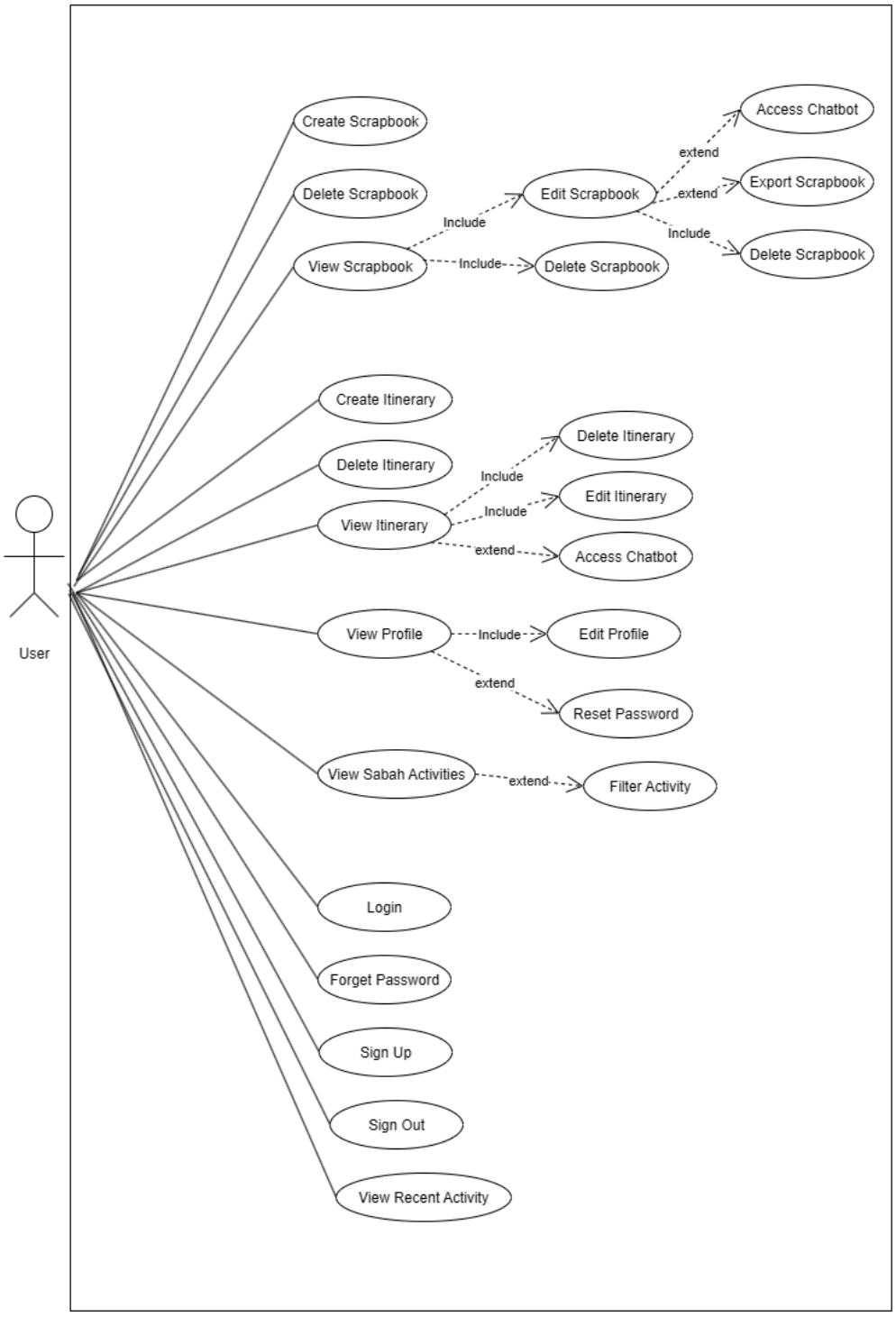

Figure 3.2: Use Case Diagram and Description

Use Case ID	UseCase 01
Use Case Name	Sign Up
Brief Description	To allow users to sign up an account.
Actor	User
Trigger	Users click the "Sign Up" button in the home screen or login screen
Main Flow	1. Users click "Sign up" button in "login" screen. 2. The system navigates users to "sign up" screen. 3. Users enter display name, email address, password, and confirm password. 4. Users click "Sign Up" button. 5. The system validates users' input. 6. The system navigates users to home screen.
Sub-flow	-
Alternate/Exception flow	1a. The system will display "Please fill out this field." error message when empty field is provided. 1b. The system will display error message "Please enter a part following '@'. Your email is incomplete.

Table 3.1: Sign Up Use Case

Use Case ID	UseCase 02
Use Case Name	Login
Brief Description	To allow users login into the system.
Actor	User
Trigger	Users click the “Sign In” button in the home screen
Main Flow	1. Users enter email address. 2. Users enter password. 3. Users click the “Sign In” button. 4. The system validates user’s input. 5. The system navigates users to home screen.
Sub-flow	-
Alternate/Exception flow	2a. The system will display “Please fill out this field.” error message when empty field is provided. 2b. The system will display “Invalid username or password. Please reenter your username or password again.” error message when username/password is wrong.

Table 3.2: Login Use Case

Use Case ID	UseCase 03
Use Case Name	Forgot Password
Brief Description	To allow users to reset password through email if they forget their password.
Actor	User
Trigger	Users click the "Forget Password" button in the home screen.
Main Flow	1. Users enter email address. 2. Users click "Send Reset Link button". 3. The system sends "Reset your TravelMate password" email to user. 4. Users click the reset password link. 5. User set new password and click "Confirm Reset" button.
Sub-flow	-
Alternate/Exception flow	-

Table 3.3: Forget Password Case

Use Case ID	UseCase 04
Use Case Name	Sign Out
Brief Description	To let users sign out
Actor	User
Trigger	Users click the "Logout" button from the navigation bar.
Main Flow	1. Users click the "Logout" button from the navigation bar. 2. User successfully sign out
Sub-flow	-
Alternate/Exception flow	-

Table 3.4: Sign Out Case

Use Case ID	UseCase 05
Use Case Name	View Recent Activity
Brief Description	To let users view their recent activities.
Actor	User
Trigger	Users view 5 recent activities through homepage or click “View All” to view all recent activities.
Main Flow	1. Users view 5 recent activities through homepage 2. Users click “View All” to view all recent activities.
Sub-flow	5.1 Users filter the activity type and date to view the recent activities.
Alternate/Exception flow	

Table 3.5: View Recent Activity Case

Use Case ID	UseCase 06
Use Case Name	Create Scrapbook
Brief Description	To let users create their Scrapbook.
Actor	User
Trigger	Users click the “Create New Scrapbook” button in the home screen or scrapbook page.
Main Flow	1. Users enter scrapbook title. 2. Users choose start date and end date. 3. Users click next. 4. Users upload photos. 5. Users click create scrapbook.
Sub-flow	5.1 User can edit the photo title or delete the photo at the upload photo page.
Alternate/Exception flow	6a. The system will display “Please enter a title” error message when Scrapbook Title field is empty. 6b. The system will display “Please select both start and end dates” error message when Start Date and End Date field is empty. 6c. The system will display “End date must be after start date” error message when tart Date is later than the End Date. 6d. The system will display “Please upload at least one photo” error message when user didn’t input photo and click create scrapbook.

Table 3.6: Create Scrapbook Case

Use Case ID	UseCase 07
Use Case Name	View Scrapbook
Brief Description	To let users view their Scrapbook.
Actor	User
Trigger	Users click the "View" button from the Scrapbook page for the specific scrapbook.
Main Flow	1. User navigates to the Scrapbook page. 2. The system displays a list of all available scrapbooks. 3. User identifies the scrapbook they want to view. 4. User clicks the **"**View**"** button for the selected scrapbook. 5. Users view the scrapbook in a detailed view.
Sub-flow	7.1 Users choose to edit or delete scrapbook when viewing the scrapbook. 7.2 Users navigate back to the Scrapbook Page through the "Back to All Scrapbooks" button. 7.3 Users export the scrapbook as PNG.
Alternate/Exception flow	-

Table 3.7: View Scrapbook Case

Use Case ID	UseCase 08
Use Case Name	Edit Scrapbook
Brief Description	To let users edit their Scrapbook.
Actor	User
Trigger	Users click the "Edit" button besides the Preview button to get into Edit Mode.
Functions in Edit Mode	1. Users can edit the text styling such as font family, font size, font weight and text colour for caption, location and description through the "Edit Text" button when user click on the edit toggle button. 2. Users can crop, zoom, rotate and add filter to the image through the "Edit Image" button when user click on the edit toggle button. 3. Users can swap the position of a specific image through the "Swap Position" button when user click on the edit toggle button. 4. Users can edit the scrapbook through the AI Assistant by clicking the AI Assistant floating button. 5. Users can access the quick actions like undo, redo, remove all caption, location, description, reset to original photos, removes all filters and shuffling the photos by hovering on the quick action floating button. 6. Users can delete specific photo by clicking on the delete toggle button. 7. Users can change the layout by clicking on the drop-down list of layouts.

	8. Users can change the background image by clicking on the background preview small icon. 9. Users edit the scrapbook title through the edit toggle button beside the scrapbook title.
Flow	1. Users edit the scrapbook. 2. Users click "Save Layout" and preview the scrapbook.
Sub-Flow	8.1. Users export the scrapbook if they want to.
Alternate/Exception flow	-

Table 3.8: Edit Scrapbook Case

Use Case ID	UseCase 09
Use Case Name	Delete Scrapbook
Brief Description	To let users delete their Scrapbook.
Actor	User
Trigger	Users click the "Delete" button from the Scrapbook page for the specific scrapbook or click the "Delete Scrapbook" button in the Scrapbook detail page.
Main Flow	1. Users click "Delete" for the selected scrapbook. 2. The system pops out the dialog box to confirm the deletion process. 3. Users click OK and delete the scrapbook.
Sub-flow	-
Alternate/Exception flow	-

Table 3.9: Delete Scrapbook Case

Use Case ID	UseCase 10
Use Case Name	View Itinerary
Brief Description	To let users view their itineraries.
Actor	User
Trigger	Users click the itinerary card they want to view in the itinerary page. The itinerary page can be accessed through the "Itinerary" button from the navigation bar or "View All" from the itinerary module in the home page.
Main Flow	1. Users navigate to the Itinerary page. 2. The system displays a list of all available itineraries. 3. Users identify the itinerary they want to view. 4. Users click the itinerary card they want to view. 5. Users view the itinerary in a detailed view. 6. User can view each day activities by clicking on the day card. Then, users will be navigated to the selected day.
Sub-flow	7.1 Users choose to edit or delete itinerary when viewing the list of the itineraries. 7.2 Users navigate back to the Scrapbook Page through the "Back to All Scrapbooks" button.
Alternate/Exception flow	-

Table 3.10 View Itinerary Case

Use Case ID	UseCase 11
Use Case Name	Edit Itinerary
Brief Description	To let users edit their itinerary.
Actor	User
Trigger	Users click into the itinerary detail page.
Edit Functions	1. Users click on the "Add Activity" button to add Activity by fill in the time, day, activity, venue and description. Then, click Save. 2. Users click on the edit toggle button on the activity they want to edit to edit the time, day, activity, venue and description. Then, click Save. 3. Users click on the delete toggle button on the activity they want to delete. 4. Users can access the chatbot to edit the itinerary.
Sub-flow	-
Alternate/Exception flow	-

Table 3.11: Edit Itinerary Case

Use Case ID	UseCase 12
Use Case Name	Delete Itinerary
Brief Description	To let users delete their Itinerary.
Actor	User
Trigger	Users click the delete toggle button from the Itinerary list page for the specific itinerary.
Main Flow	1. Users click delete for the selected itinerary to delete. 2. The system pops out the dialog box to confirm the deletion process. 3. Users click OK 4. The itinerary is deleted.
Sub-flow	7.1 Users choose to edit or delete scrapbook when viewing the scrapbook. 7.2 Users navigate back to the Scrapbook Page through the “Back to All Scrapbooks” button. 7.3 Users export the scrapbook as PNG.
Alternate/Exception flow	-

Table 3.12: Delete Itinerary Case

Use Case ID	UseCase 013
Use Case Name	Cretae Itinerary
Brief Description	To let users view their Scrapbook.
Actor	User
Trigger	Users click the “Create New Itinerary” button from the Itinerary list page or home page.
Main Flow	1. Users input the travel style, budget, trip duration, destination in Sabah and Trip Description. 2. Users click the “Generate My Sabah Itinerary” button.
Sub-flow	-
Alternate/Exception flow	13a. The system will display “Please fill out this field” error message when there are empty field.

Table 3.13: Create Itinerary Case

Use Case ID	UseCase 14
Use Case Name	View Profile
Brief Description	To let users view their profile.
Actor	User
Trigger	Users click their name that appears in the navigation bar.
Main Flow	1. User navigates to the profile page through clicking their name in the navigation bar.
Sub-flow	7.1 Users edit their display name. 7.2 Users change password.
Alternate/Exception flow	-

Table 3.14: View Profile Case

Use Case ID	UseCase 15
Use Case Name	View Sabah Activities
Brief Description	To let users view Sabah activities.
Actor	User
Trigger	Users click the “Activities” from the navigation bar.
Main Flow	1. User navigates to the Activities page. 2. The system displays a list of activities.
Sub-flow	7.1 Users filter the activities according to the type and area.
Alternate/Exception flow	-

Table 3.15: View Sabah Activities Case

3.2 Activity Diagram

Create Scrapbook

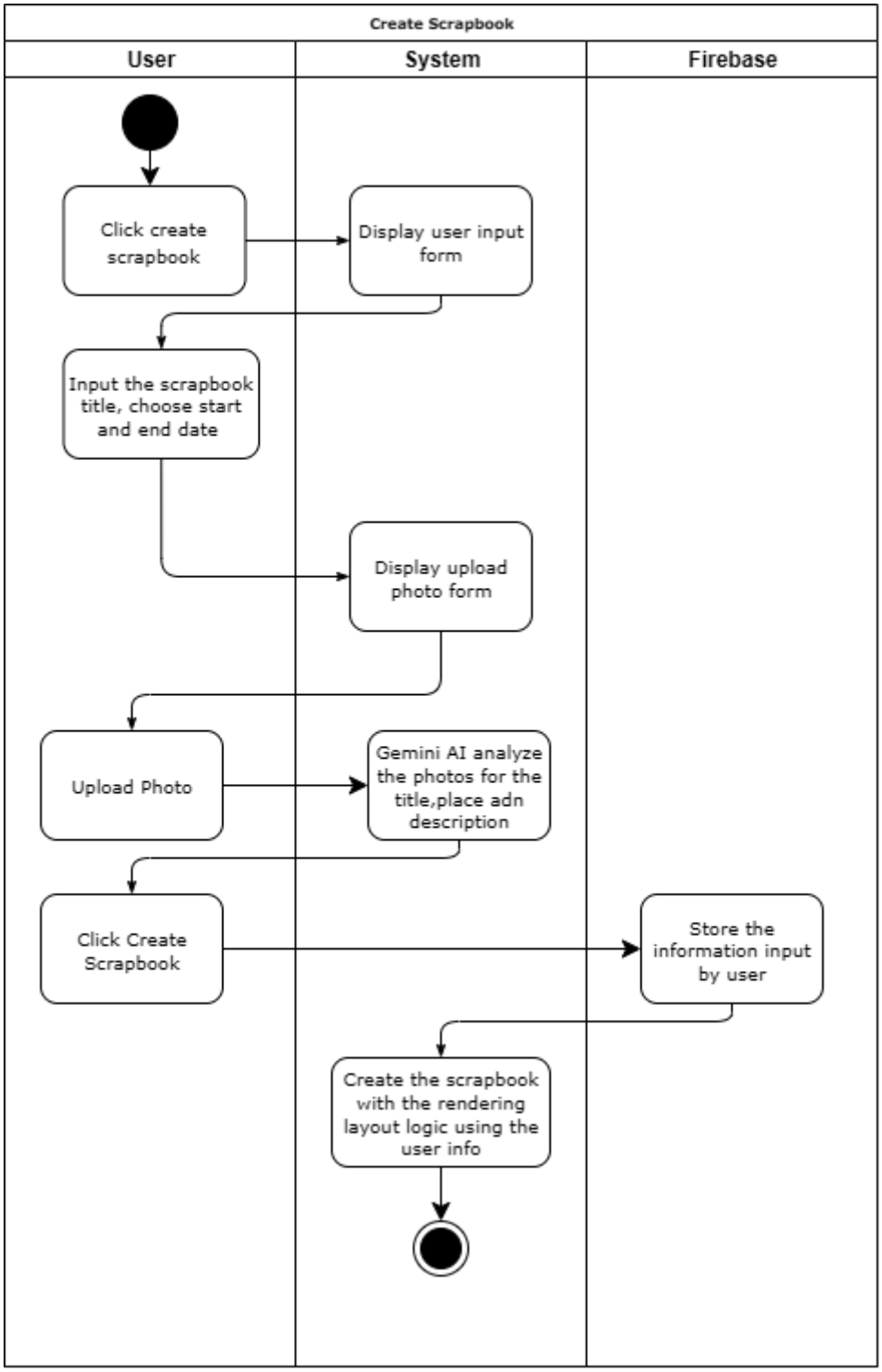

Figure 3.3: Create Scrapbook Activity Diagram

View Scrapbook

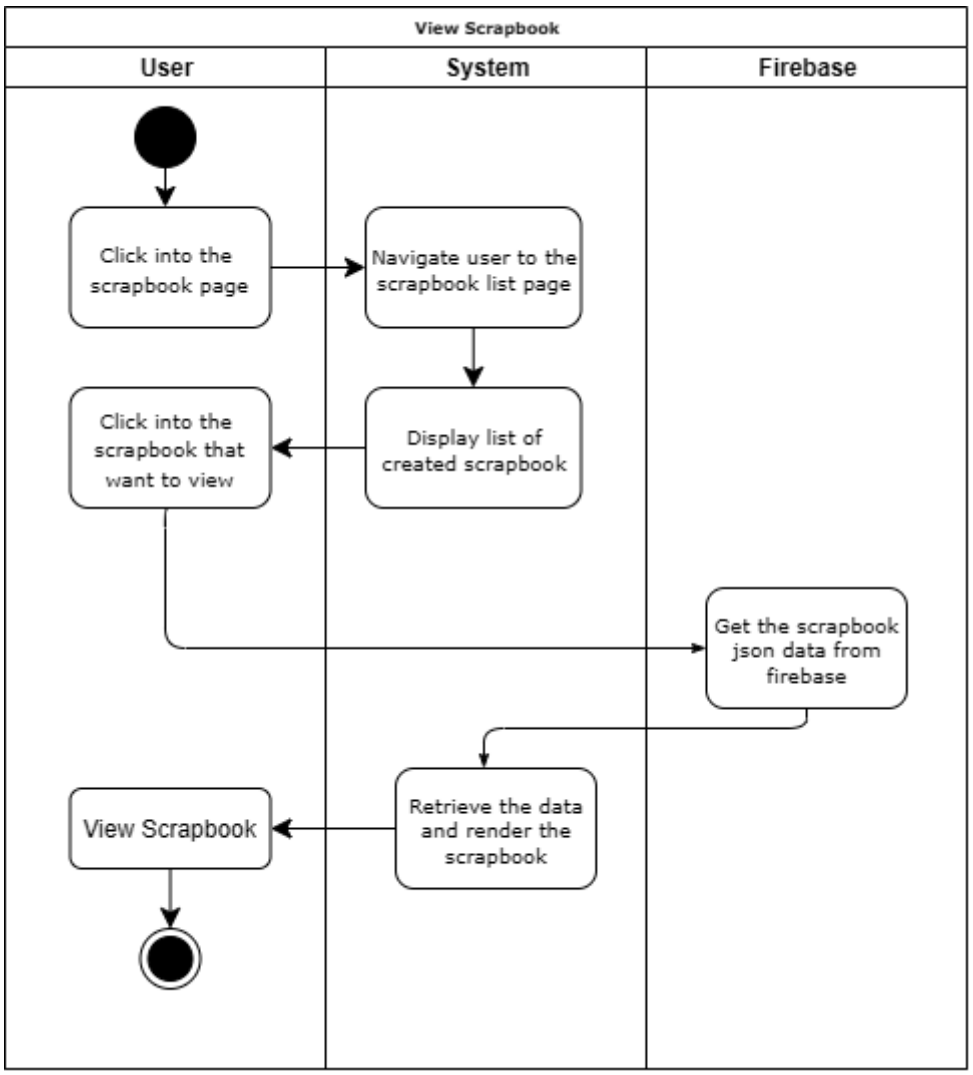

Figure 3.4: View Scrapbook Activity Diagram

Edit Scrapbook

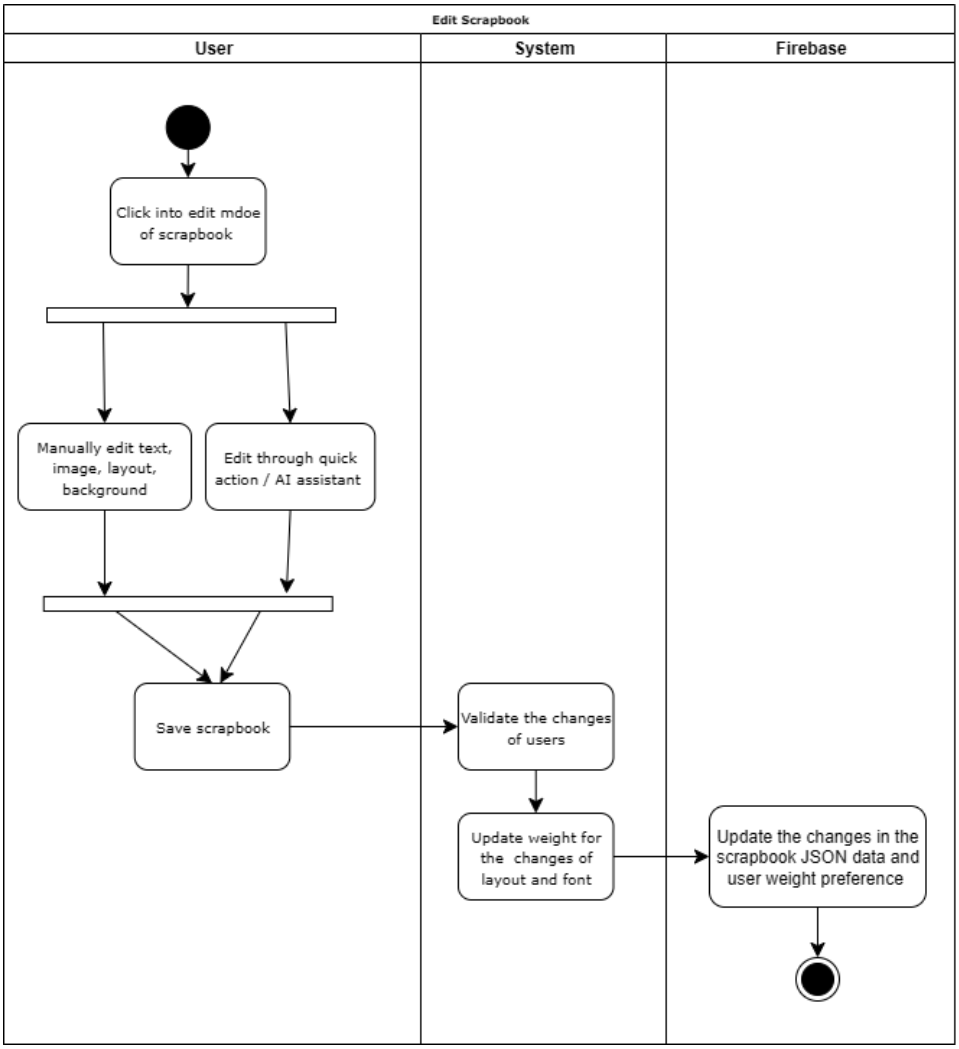

Figure 3.5: Edit Scrapbook Activity Diagram

Delete Scrapbook

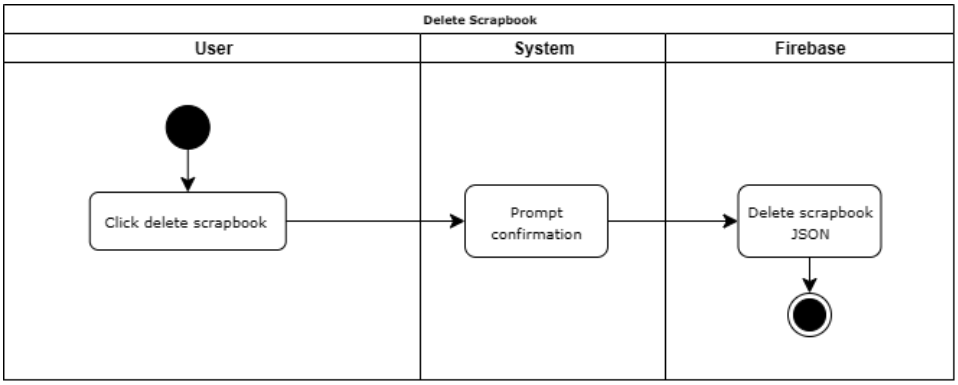

Figure 3.6: Delete Scrapbook Activity Diagram

Create Itinerary

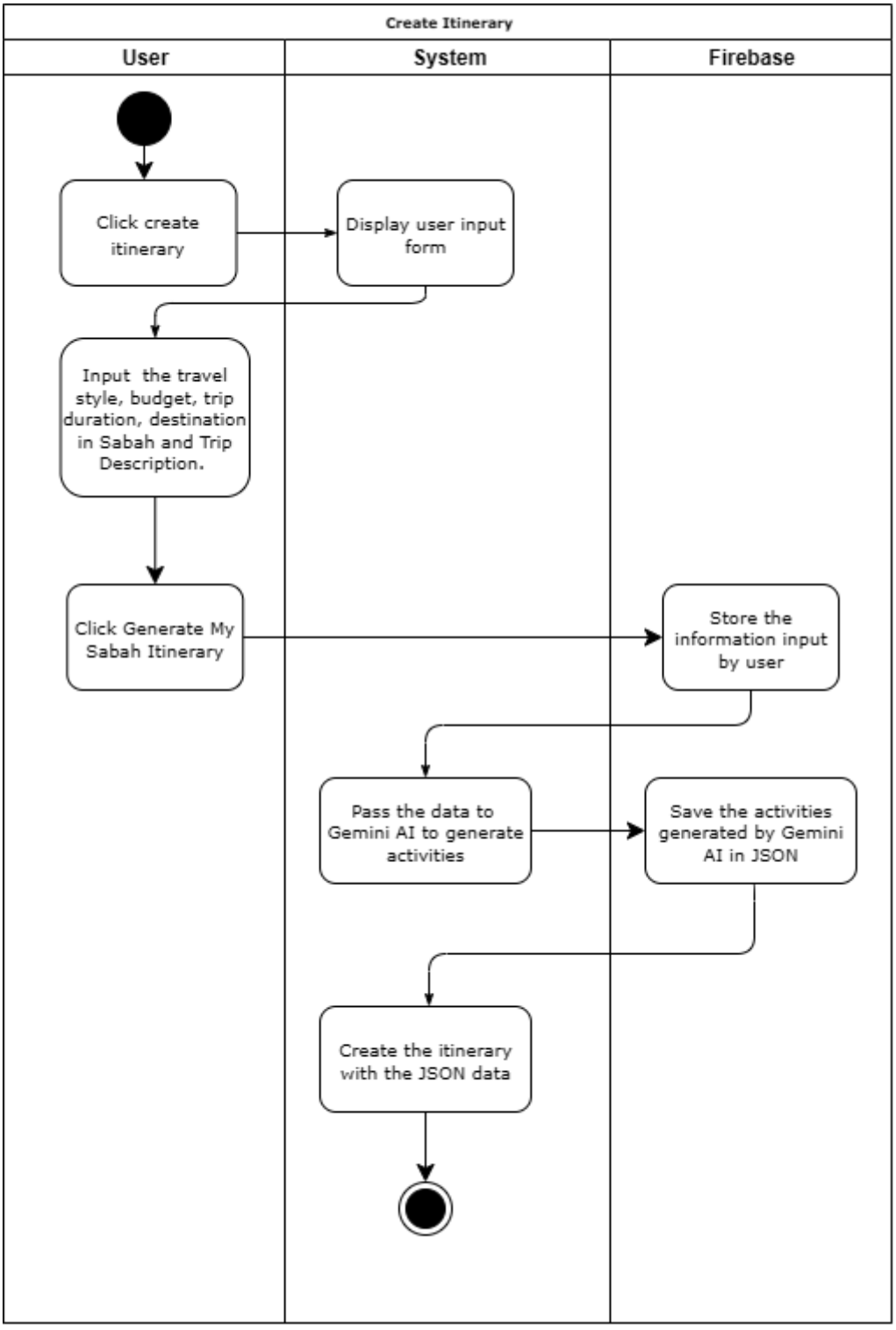

Figure 3.7: Create Itinerary Activity Diagram

View Itinerary

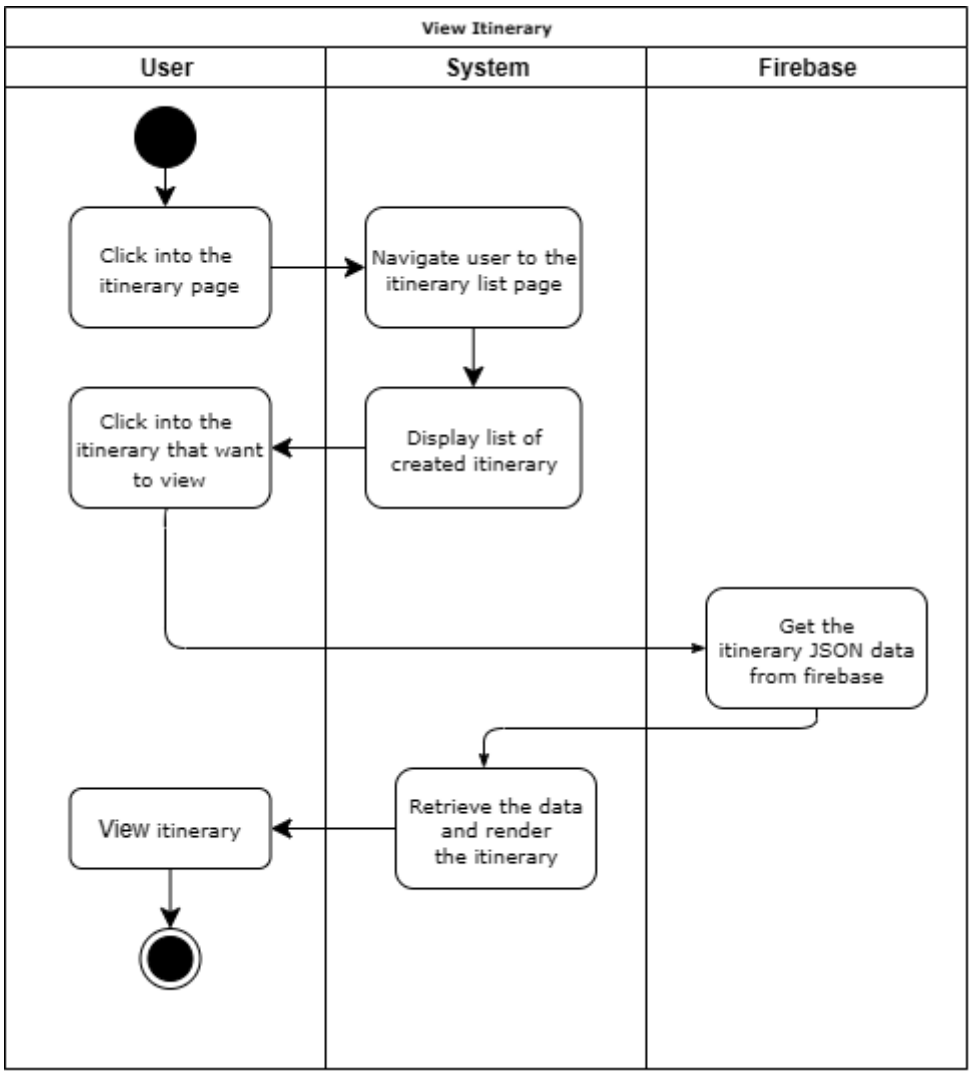

Figure 3.8: View Itinerary Activity Diagram

Edit Itinerary

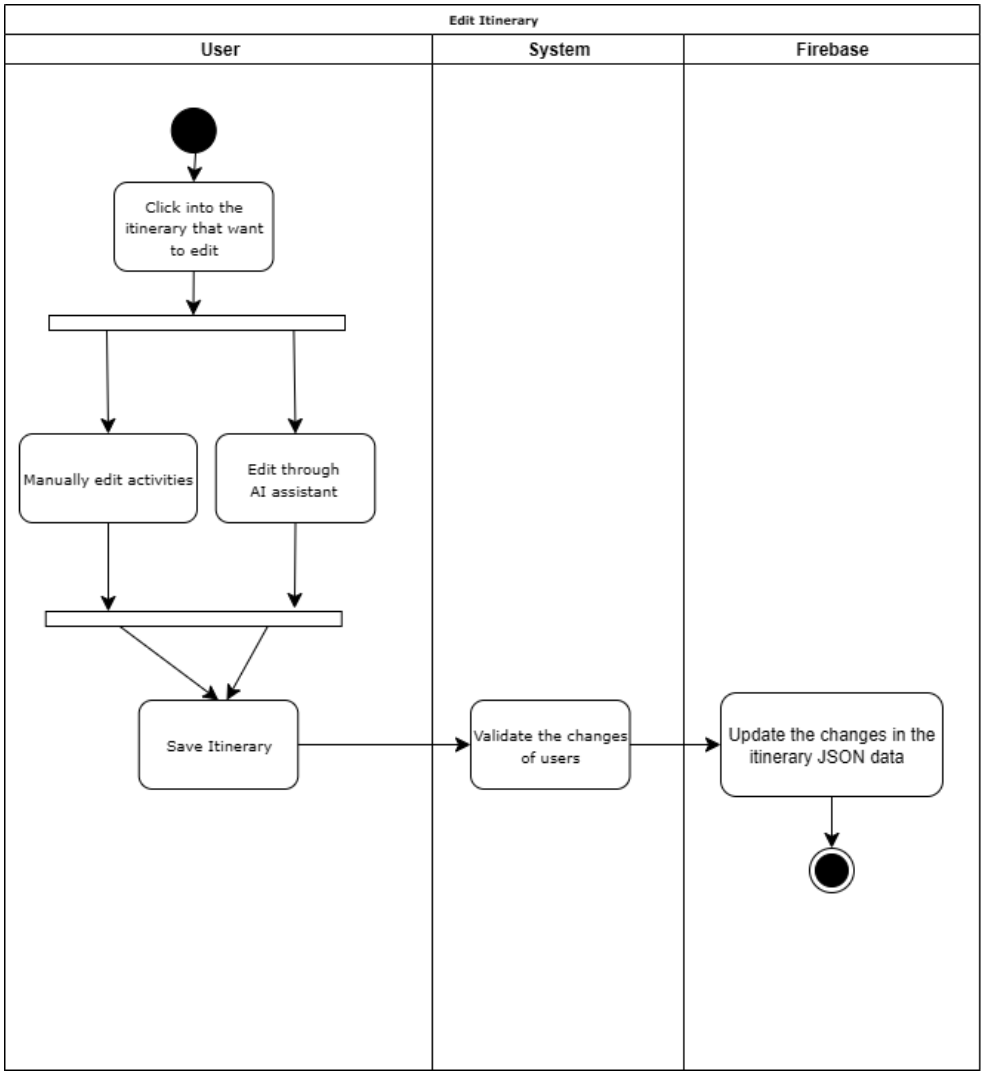

Figure 3.9: Edit Itinerary Activity Diagram

Delete Itinerary

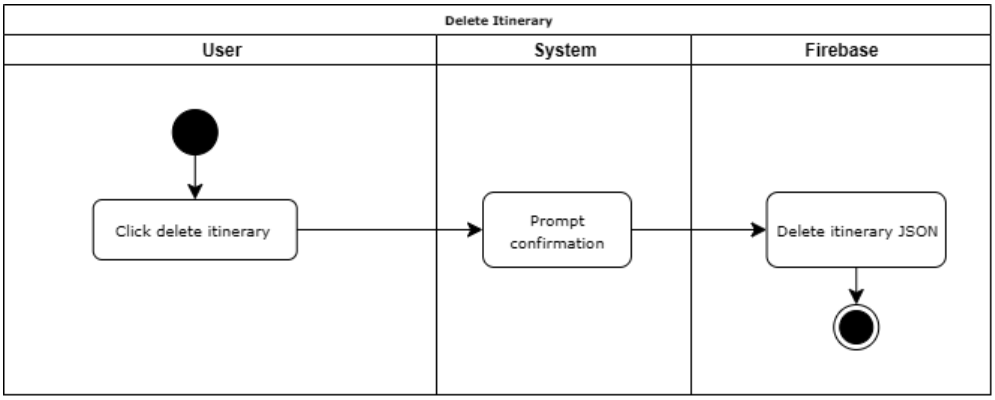

Figure 3.10: Delete Itinerary Activity Diagram

View Sabah Activities

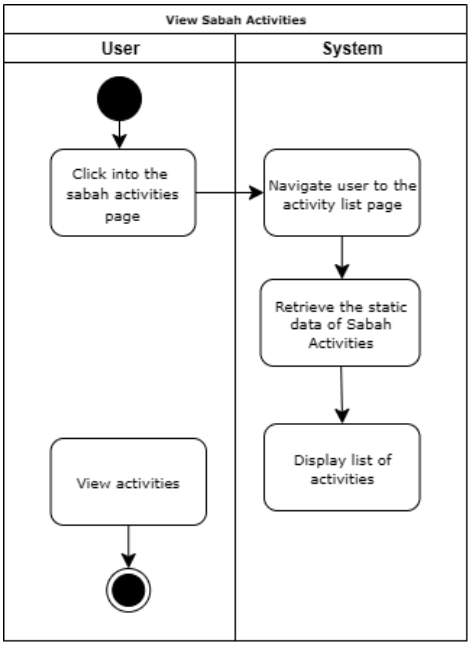

Figure 3.11: View Sabah Activities Activity Diagram

Sign In

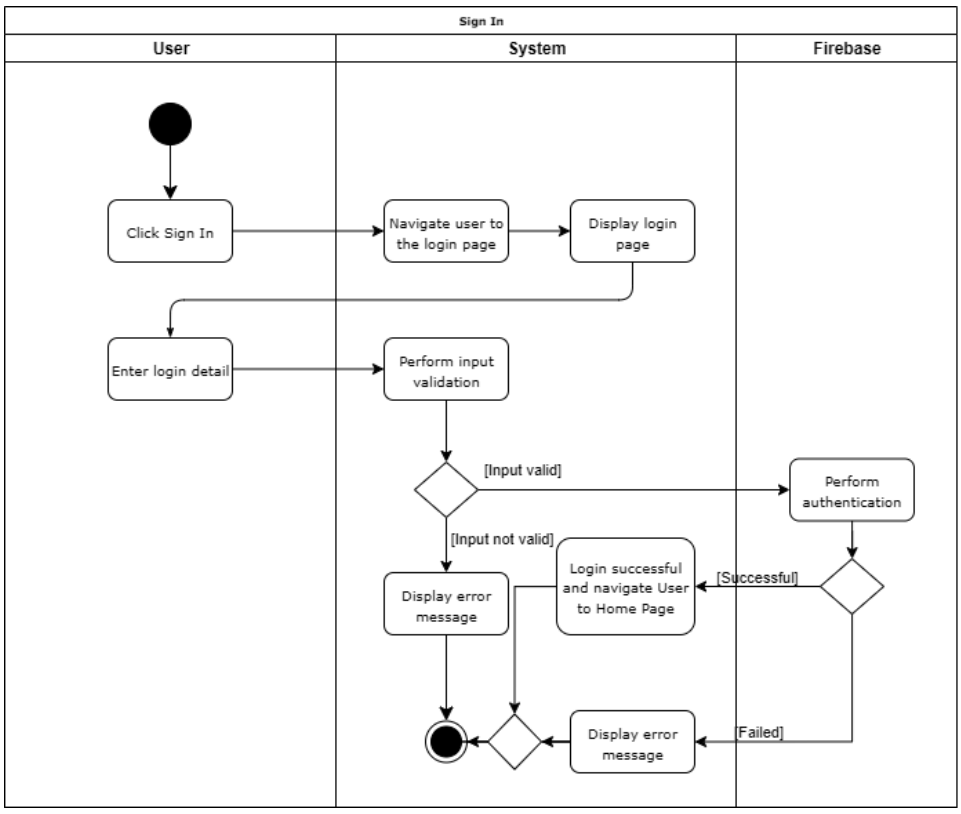

Figure 3.12: Sign In Activity Diagram

Sign Up

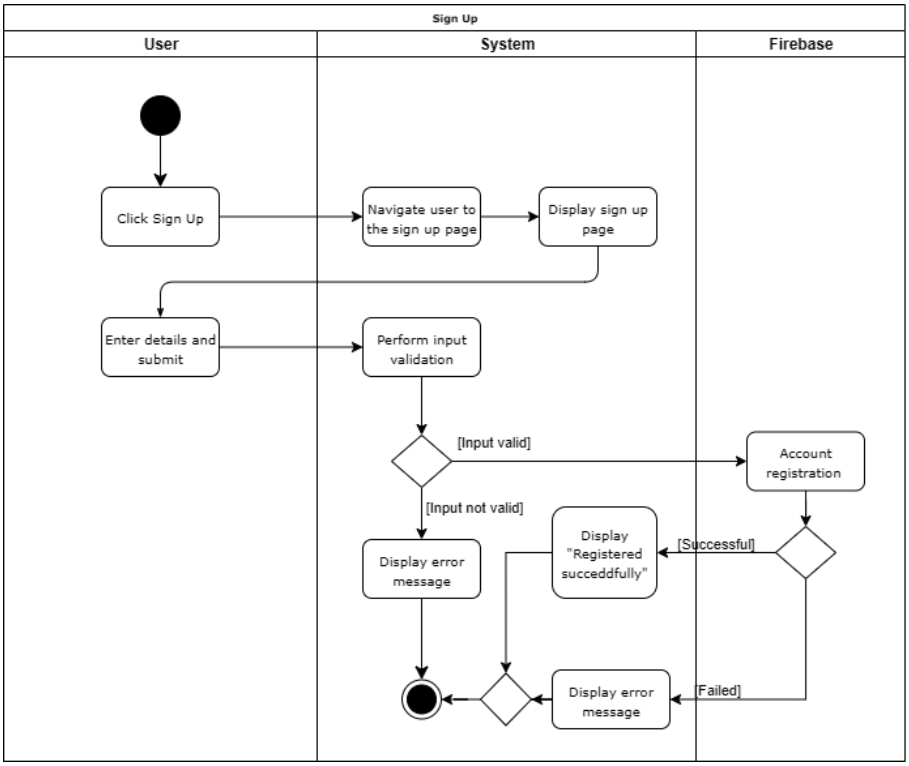

Figure 3.13: Sign Up Activity Diagram

Forgot Password

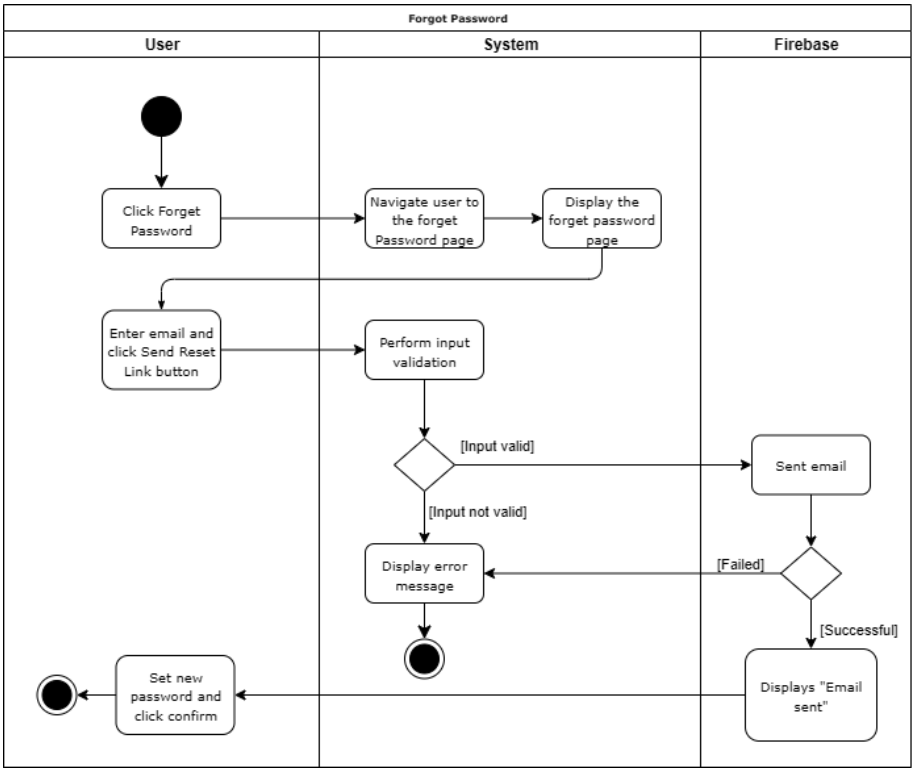

Figure 3.14: Forgot Password Activity Diagram

Reset Password

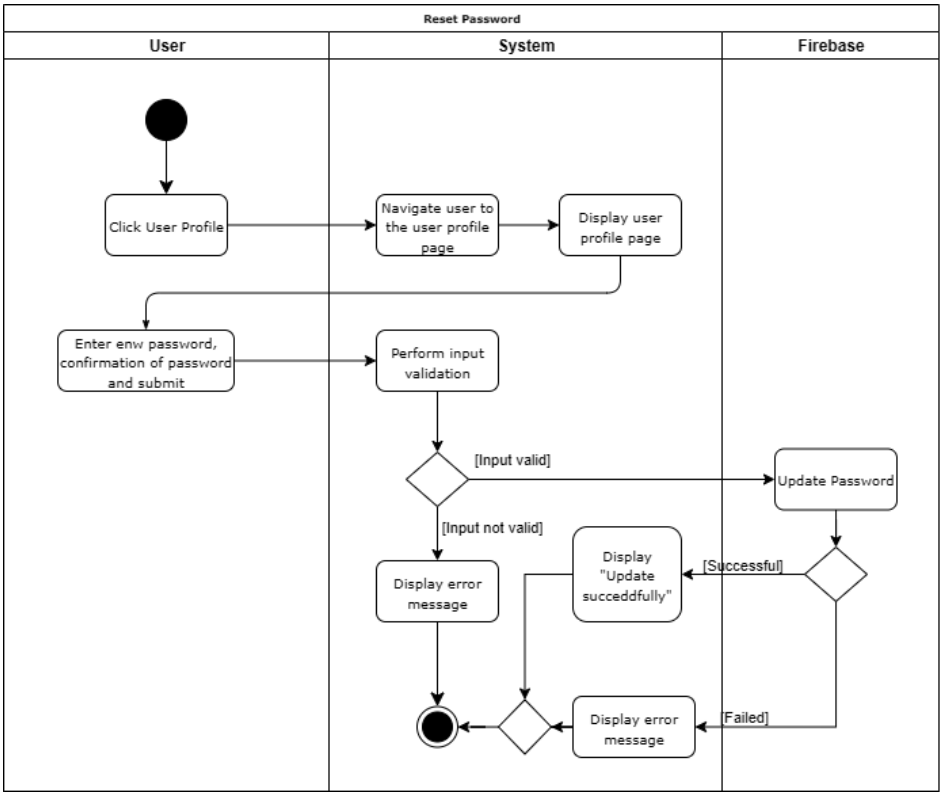

Figure 3.15: Reset Password Activity Diagram

Sign Out

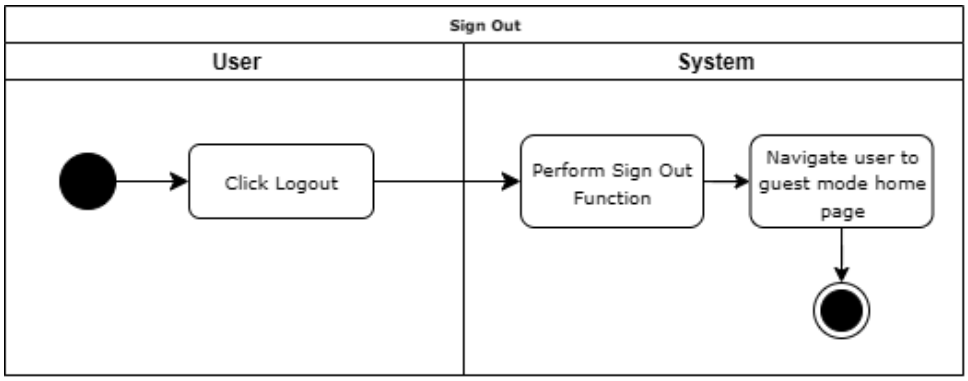

Figure 3.16: Sign Out Activity Diagram

3.3 Flowchart

Complete Travel Experience Platform: From Planning to Memories

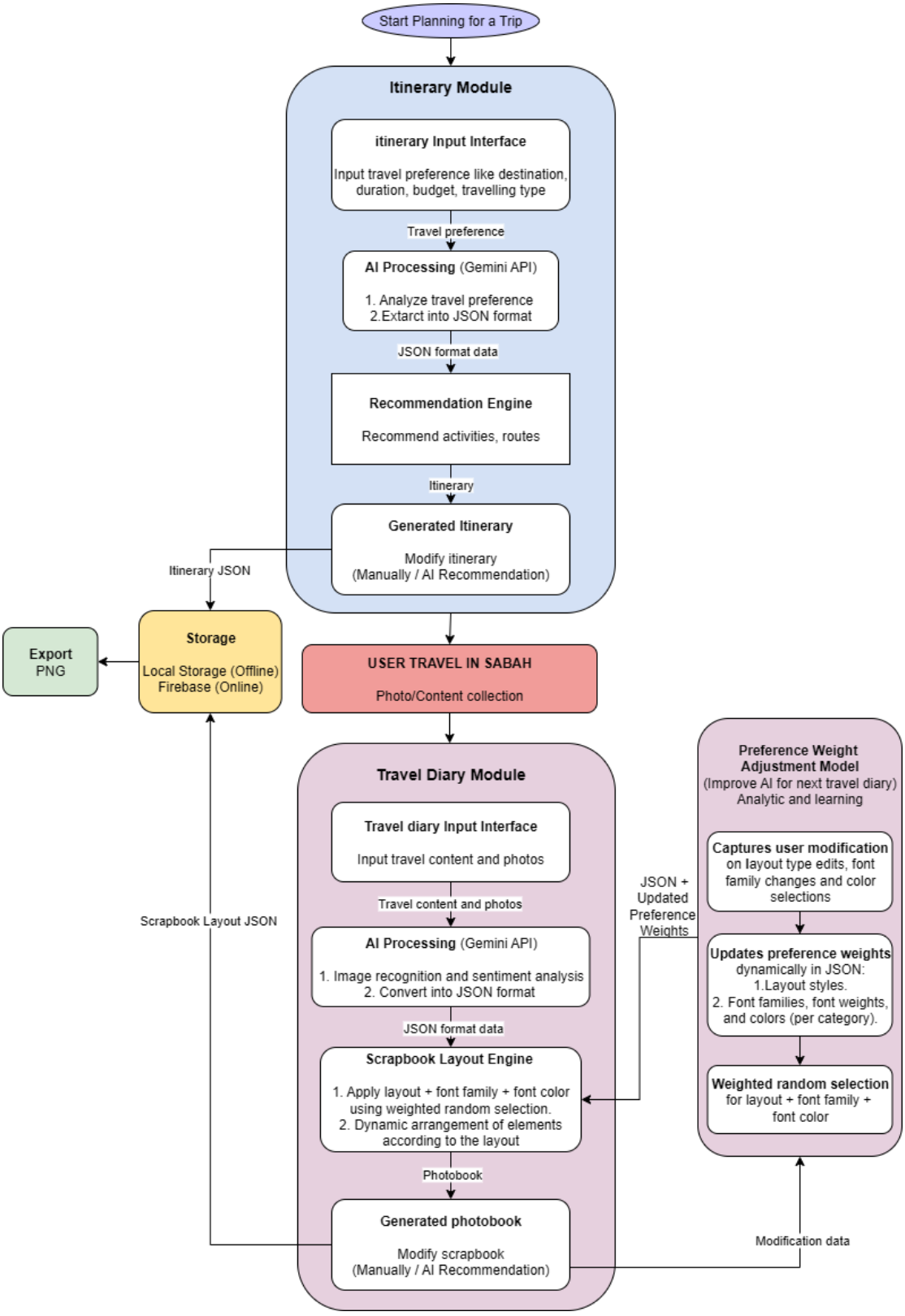

Figure 3.17: System Flow Overview Diagram

The system flow diagram illustrates the complete journey of users, beginning with pre-trip planning through the Itinerary Module and extending into post-trip documentation using the Travel Diary (Scrapbook) Module. The design highlights how AI assistance, storage mechanisms, and a preference weight adjustment system work together to provide a personalized, adaptive, and end-to-end travel experience.

The Itinerary Module manages the pre-trip planning phase by guiding users in generating customized travel itineraries. It begins with an intuitive input interface where users specify preferences such as destination, travel duration, budget, and travel type, including options like adventure, food trips, or cultural exploration. These inputs are processed by the Gemini API, which transforms the raw data into a structured JSON format suitable for further analysis. A recommendation engine then interprets this structured information to suggest relevant attractions, activities, routes, and schedules tailored to the user's preferences. From these suggestions, a personalized itinerary is generated, which remains flexible as users can manually edit the details or adjust them with the support of the AI assistant. Once finalized, the itinerary can be exported in PNG format for quick reference locally and backed up in Firebase to ensure cross-device synchronization and availability. This structured workflow allows the module to balance automation and personalization, delivering a lightweight yet adaptive itinerary planning system that supports the scrapbook-focused travel companion.

The integration point, "Travel in Sabah," serves as the transition between itinerary planning and journaling, bridging the pre-trip and post-trip phases. Once users complete their journey, they return to the application with photos, captions, and travel notes. This content then feeds directly into the scrapbook generation workflow, transforming travel experiences into a structured and visually engaging narrative.

The Travel Diary (Scrapbook) Module manages the post-travel phase by converting raw travel content into an AI-assisted scrapbook. The process begins with an input interface where users upload photos, captions or notes, and specify travel dates. This data is then analyzed through the Gemini API, which performs image recognition to categorize photos, sentiment analysis on textual input, and transforms the results into a structured JSON format. The Scrapbook Layout Engine interprets this structured data alongside stored preference weights to select photobook templates, apply layout configurations, and dynamically arrange photos and text. An initial scrapbook is

generated and presented to the user, who can either edit it manually or refine it further through AI assistance. Once finalized, the scrapbook is stored as a Scrapbook Layout JSON and made available for export in PNG format, enabling offline storage and easy sharing. To ensure accessibility, the scrapbook is saved in both local storage (PNG) for quick offline access and Firebase for cross-device synchronization and backup.

The Preference Weight Adjustment Model enhances personalization by adapting scrapbook layouts to reflect individual editing behaviours. Each user modification such as layout adjustments, font family changes, or color selections is captured and dynamically incorporated into updated JSON-based preference weights. These updated weights are then fed back into the Scrapbook Layout Engine, ensuring that future scrapbook generations better align with the user's creative style and aesthetic preferences. This feedback loop establishes a continuous improvement cycle, allowing the scrapbook to evolve into a truly personalized digital memory book that mirrors the traveller's unique journey.

CHAPTER 4

System Design

4.1 System Block Diagram

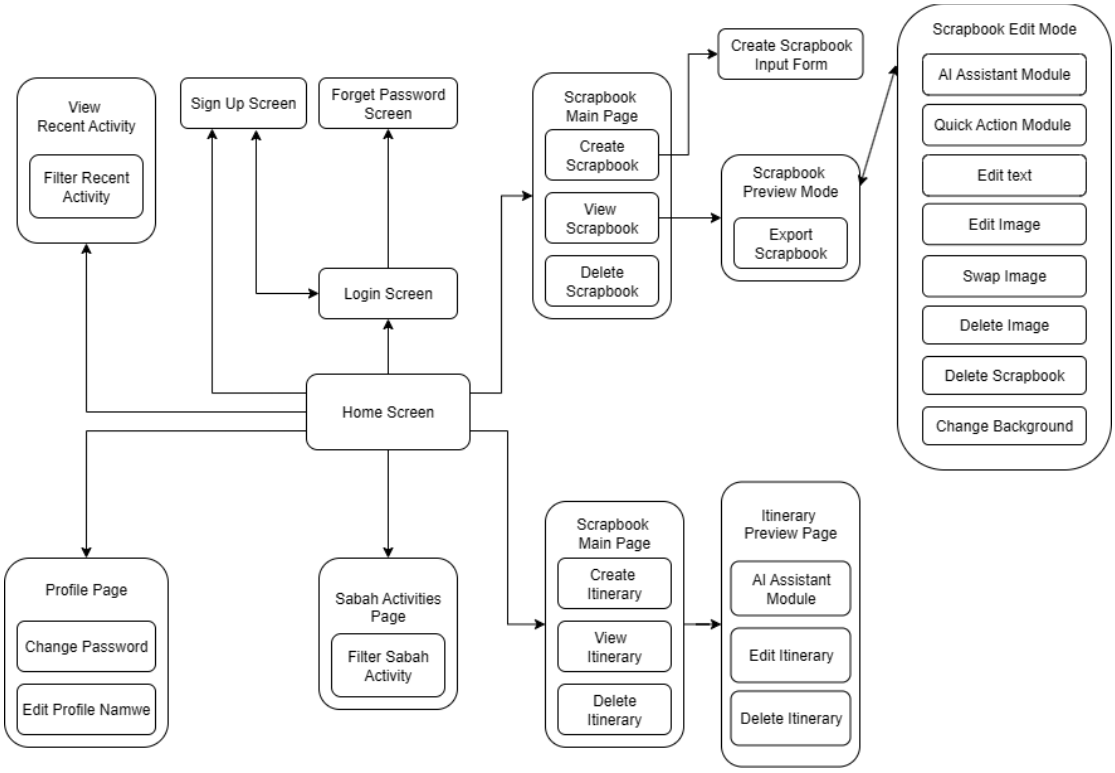

Figure 4.1 System Block Diagram

The system block diagram illustrates the overall flow and interactions of the Sabah Smart Travel Companion, highlighting its two main functional phases: itinerary planning and post-travel scrapbook generation. Each module is designed to provide users with an integrated travel experience, from trip preparation to memory preservation.

In the planning phase, users can access the Sabah Activities Page to view available destinations, foods, and activities. The system enables users to create personalized travel itineraries by specifying destinations, travel dates, and preferences. With the support of an AI Assistant Module, itineraries can be automatically generated

and later customized by users to better fit their travel expectations. Users can edit itineraries, add or remove activities, and adjust details before saving them in the system. The itinerary preview page further enhances flexibility by offering options to refine or delete itineraries at any time.

In the post-travel phase, the scrapbook feature enables users to document and relive their travel experiences. From the Scrapbook Main Page, users can create, view, or delete scrapbooks. Upon selecting "Create Scrapbook," users are directed to the Scrapbook Input Form to upload images and provide details. The system applies AI techniques to recognize and categorize images while also offering automatic caption suggestions. Once the scrapbook is generated, users enter Scrapbook Edit Mode, which contains multiple modules such as text editing, image editing, swapping, deletion, and background customization. Additionally, the scrapbook integrates both a Quick Action Module for direct editing and an AI Assistant Module for smart recommendations. Users can preview their scrapbook in Scrapbook Preview Mode and export it into a high-quality format for local storage, ensuring long-term memory preservation.

Supporting modules such as authentication (login, sign-up, and password recovery), profile management, and recent activity filtering provide an additional layer of personalization and security. The system ensures that users can manage their profiles, update passwords, and track their activities effectively.

The hybrid architecture of the system balances local storage and Firebase cloud integration. Local storage supports scrapbook responsiveness by enabling user can immediately access to uploaded photos. Firebase ensures synchronization across devices by managing authentication, itineraries, scrapbook metadata, and user preferences. This design provides seamless transitions between desktop and mobile devices, giving users consistent access to their data.

Overall, the system block diagram reflects a comprehensive travel companion that goes beyond traditional travel applications. By combining itinerary planning with scrapbook creation in single integrated framework, it delivers a comprehensive travel experience. The integration of AI-powered assistance, scrapbook editing tools, and a hybrid storage design ensures that users not only plan effectively but also preserve their memories in a dynamic, customizable, and secure way.

4.2 System Components Specifications

The system architecture is composed of several interconnected modules, each with a distinct role in ensuring seamless scrapbook creation, itinerary planning, and data persistence. These components collectively provide the functionality required to capture user input, process content, store information reliably, and enable cross-device access. The design emphasizes modularity, with each module handling specific responsibilities while contributing to the overall goal of delivering a personalized and efficient travel documentation experience.

4.2.1 Scrapbook Module

The Scrapbook Module is responsible for generating visually appealing scrapbook layouts that combine user photos, captions, travel dates, and optional filters. This module manages both the creative arrangement of photos and the flexibility for users to make edits. It supports photo filters, caption customization, and exporting scrapbook pages in shareable formats such as PNG.

Inputs: Photos, captions, travel dates, optional filters

Outputs: Scrapbook pages

Responsibilities:

- Generate scrapbook layouts based on predefined design rules.
- Apply user-selected filters to photos.
- Enable editing of layouts and captions.
- Export completed scrapbook pages (PNG) for sharing or storage.

The module integrates with the Preference Weight Adjustment Model by recording user actions such as accepting or rejecting layouts and applying filters, which are later used to refine future default suggestions.

4.2.2 Itinerary Planning Module

The Itinerary Planning Module handles the creation of personalized travel itineraries based on user input. It processes trip details such as location, dates, and activity interests through the Gemini API, generating structured travel schedules. The itineraries are presented to users for interactive review and editing to ensure alignment with their expectations.

Inputs: User trip details (location, start/end dates, activity interests

Outputs: Day-by-day structured travel itinerary

Responsibilities:

- Collect and process trip details from the user.
- Use the Gemini API to generate activity recommendations.
- Convert processed results into structured JSON for system interpretation.
- Visualize itineraries for user review and adjustments.
- Export completed scrapbook pages (PNG) for sharing or storage.

4.2.3 Preference Weight Adjustment Model

The Preference Weight Adjustment Model ensures that scrapbook generation evolves according to user behaviour. It captures scrapbook edit actions such as filter application, layout changes, or approval decisions, and translates them into updated probability weights. These weights influence future scrapbook generations, making layouts and design suggestions more aligned with user preferences over time.

Inputs: User scrapbook edits (layout modifications, fonts modification)
Outputs: Updated probability distributions per scrapbook (e.g., layout style, font type)

Responsibilities:

- Log scrapbook editing actions from users.
- Adjust weight/probability values for each editable category.
- Store updated preferences in Firebase for persistence.

- Apply refined probabilities in subsequent scrapbook generation.

4.2.4 Sabah Activities Module

The Sabah Activities Module provides users with a curated directory of attractions, activities, and local food experiences across Sabah. It supports an interactive filtering system that enables users to browse content based on **category** (e.g., Nature, Adventure, Food, Culture) and **geographical area** (e.g., Kota Kinabalu, Sandakan, Tawau). This module enhances user exploration by delivering structured information with descriptions, images, and categorization for easy discovery.

Responsibilities:

- Provide a structured database of Sabah attractions, activities, and food listings.
- Support filtering by **category** (Nature, Food, Culture, etc.) and **area** (districts or regions of Sabah).
- Display detailed descriptions, images, and metadata (e.g., location, category, highlights).
- Offer responsive search and filtering functionality for improved user experience.
- Integrate seamlessly with the Itinerary Planning Module by allowing selected items to be added into a user's travel itinerary.

4.3 Components Design

4.3.1 Scrapbook Module

The Scrapbook Module functions as a post-travel journaling engine that converts user-provided media such as photos, captions, and travel dates into structured scrapbook pages. Its design strikes a balance between automated generation and manual control. Automation is achieved through AI-assisted photo analysis, JSON structuring, and layout generation that applies user-specific preference weights. At the same time, the module empowers users to personalize outputs by overriding captions, choosing custom fonts, applying filters, and reorganizing layouts before exporting the final scrapbook.

The architecture of this module follows a pipeline design. User will upload photos, captions and dates, then the uploaded photo will be classified into categories and enriched with metadata such as location and timestamps by Gemini API. The results are stored in structured JSON for consistency. The layout engine then generates scrapbook pages, drawing on design rules and existing preference weights. User edits and customizations are layered on top of these automatically generated layouts, enabling real-time previews of personalized scrapbook pages. Completed pages can then be exported in formats such as PNG, while metadata and weight adjustments are saved in Firebase for persistence and long-term personalization.

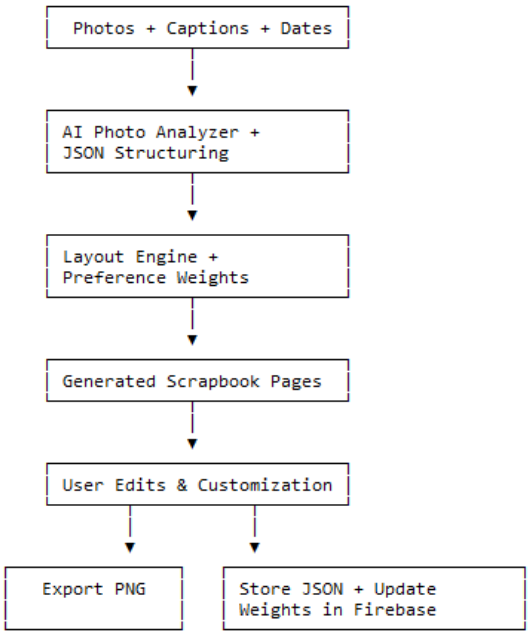

Figure 4.2 Architecture flow of Scrapbook Module

Scrapbook Generation Process

The scrapbook generation process is structured into multiple phases to ensure a seamless workflow from user input to final rendering.

Initial Data Collection and Structuring

The scrapbook creation process begins with the ScrapbookForm.tsx component, which serves as the main interface for collecting user inputs. This component gathers essential information such as the scrapbook title, travel dates, and photo uploads through a streamlined multi-step form. Uploaded photos are passed to PhotoAnalyzerService.tsx, where Gemini LLM analyzes each image and generates descriptive metadata, including custom titles, locations, and descriptions. The form manages complex state to ensure that all photos retain their associated metadata, implements duplicate detection to avoid redundant processing, and limits uploads to 15 photos to maintain performance. Upon submission, the inputs are structured into a standardized format for further processing in subsequent services.

AI-Powered Photo Analysis

Following data collection, the system leverages the PhotoAnalyzerService.tsx to perform comprehensive AI analysis on uploaded images. This service integrates with Gemini API to extract meaningful information from each photograph. The analysis process examines multiple aspects of the images, including visual content categorization, location detection, and aesthetic assessment. For a Sabah travel scrapbook, the AI might analyze a photograph of Mount Kinabalu and automatically categorize it as a "Nature" photo, extract location data indicating "Kota Kinabalu," and generate descriptive text highlighting the mountain's majestic appearance.The photo analysis service employs machine learning algorithms to determine appropriate photo categories from a predefined set including Nature, Cityscape, Landmark, Food, People, Selfie, Culture, Shopping, Activity, Animal, Art, and Others. This categorization proves crucial for subsequent layout decisions and

theme applications. The service also extracts dominant color information, which informs the visual styling decisions made by the layout generation system.

```
If you're uncertain about the category, use 'Others'.`;

    // Add location request if not provided
    if (userLocationInput) {
      promptText += `\n\nVerify if the location appears to be consistent with the user-specified location: "${userLocationInput}". If it's clearly different or impossible to determine, provide your best estimate.`;
    } else {
      promptText += `\n\nAlso, try to identify the potential location shown in this photo. If you can't determine a specific location, provide a general description of the place type.`;
    }

    promptText += `\n\nIn your description, be very specific about what is visible in the photo.
    For example, if it's a mountain, specify "Mount Kinabalu peak with clouds" rather than just "A mountain landscape". If it's a beach, mention "Sunset at Kota Kinabalu Beach" rather than just "A beach scene".

Your description should focus on distinctive features visible in the photo that would make someone recognize the exact place or scene.

MOST IMPORTANTLY: Write your description in FIRST PERSON perspective, as if you're the traveler who took this photo.
Use an excited, conversational tone like someone writing in their travel journal.
Include personal reactions and feelings.

Examples:
- "I finally tried the famous Sabah laksa at this local restaurant! The broth was so rich and flavorful with fresh seafood."
- "The view from Mount Kinabalu was absolutely breathtaking! I couldn't believe how beautiful the sunrise looked from up here."
- "We had such an amazing time exploring this night market in KK! The local street food was incredible."

Respond ONLY in this exact JSON format with no additional text:
{
  "category": "SINGLE_CATEGORY_NAME",
  "location": "SPECIFIC_LOCATION_NAME_OR_TYPE",
  "description": "FIRST_PERSON_EXCITED_DESCRIPTION_OF_WHAT_YOU_SEE",
  "title": "VERY_SHORT_SPECIFIC_TITLE_3_TO_5_WORDS"
}

The title should be extremely specific and relate directly to what's in the photo, like "Peak of Mount Kinabalu", "Sunset at Railay Beach", or "Night Market Food Stalls".

Where SINGLE_CATEGORY_NAME is ONLY the main category name (Nature, Cityscape, Landmark, etc.) - do not include the description part.`;
```

Figure 4.3 Prompt text that pass to Gemini to analyze photos

Machine Learning-Based Personalization

The system's personalization capabilities are managed through the TrainingService.ts, which implements sophisticated machine learning algorithms to learn from user behavior patterns. This service maintains detailed user preference profiles that track layout choices, font preferences like font family and font color selections across different photo categories. The system employs weighted random selection algorithms rather than deterministic recommendations, ensuring that user preferences guide decisions while maintaining variety and preventing over-optimization. The TrainingService maintains separate preference weights for layout types (grid, full_width, mixed, category_based) and font combinations per photo category. When a user has previously created scrapbooks with a preference for grid layouts and green colors font for nature photos, the system increases the probability weights for these choices while still allowing for occasional variation. This approach balances personalization with user exploration, creating a dynamic system that adapts to user preferences without becoming repetitive.

Layout Generation

The CompactScrapbookLayoutService.ts is responsible for transforming analyzed photo data into structured scrapbook layouts for the user. The service uses four predefined templates—Grid, Mixed, FullWidth, and CategoryBased—each with its own layout styling rules to organize photos, captions, and other visual elements. It also incorporates user preferences derived from the Preference Weight Adjustment Model, adjusting elements like font style, size, color, and photo arrangement according to past modifications and choices. Based on the selected template and user preference, the service generates a structured layout that automatically arranges photos and associated metadata. For example, for a Nature-themed Sabah scrapbook, the system might apply a green color palette with earth-tone accents, select serif fonts that evoke natural elegance, and arrange photos in a grid pattern that emphasizes the landscape's grandeur, fully reflecting the user’s personalization.

File Processing and Storage Management

The UserScrapbookService.ts manages the complex process of file upload, storage, and database persistence. This service handles the upload of image files to Firebase Storage, generating secure download URLs that enable efficient image retrieval. The service implements comprehensive data cleaning processes to ensure that File objects are properly removed from data structures before database storage, preventing serialization errors and maintaining data integrity.The service also manages the creation of comprehensive scrapbook records in Firestore, including all metadata, layout configurations, and user preferences. The database records maintain referential integrity between users and their scrapbooks while implementing efficient querying capabilities for retrieval and management operations.

Scrapbook Editing and Customization System

Interactive Layout Editor

The scrapbook editing functionality is managed through the ScrapbookPreviewEditor.tsx component, providing users with useful tools to customize their scrapbooks after creation. The editor integrates with TrainingService to apply learned user preferences while allowing manual overrides, updating preference weights whenever users modify fonts, colors, or other styling elements to inform future scrapbook suggestions.

Focused editing is provided via modal components:

- EditImageModal.tsx allows zooming, cropping, rotating, and applying filters to individual photos.
- EditTextModal.tsx handles text editing, offering controls for font family, size, weight, color, and styling of captions, locations, and descriptions. Users can also enhance captions using the AI Assistant.

Quick action buttons are also provided for frequently performed tasks, allowing users to streamline repetitive edits efficiently. The quick actions are accessible via a floating button, enabling undo, redo, remove all captions/locations/descriptions, reset photos, remove filters, and shuffle photos for efficient editing.

The ScrapbookChat.tsx component provides an AI Assistant that helps users perform editing actions through a conversational interface. Using natural language processing, the component interprets commands such as "apply a warmer filter to all nature photos" or "clear all descriptions" and executes the edits directly. The system maintains context awareness, recognizing the current state of the scrapbook to accurately apply requested changes.

Key User Functions:

1. Edit text styling (font family, size, weight, color) for captions, locations, and descriptions via the “Edit Text” button.
2. Crop, zoom, rotate, and apply filters to photos using the “Edit Image” button.
3. Swap positions of specific images through the “Swap Position” button.
4. Enhance scrapbook content with AI via the AI Assistant button.
5. Use quick actions (undo, redo, reset, remove captions/filters, shuffle) through the floating button.
6. Delete specific photos via the delete toggle button.
7. Change scrapbook layouts from a drop-down menu.
8. Change the background image via the background preview icon.
9. Edit the scrapbook title through the edit toggle beside the title.

System Integration and Data Flow

The scrapbook system demonstrates seamless integration between multiple specialized services, each handling specific aspects of creation and editing. TravelDiaryApp.tsx acts as the main orchestrator, coordinating data flow between services and managing the overall user experience. It implements complex state management to handle transitions between creation and editing modes while maintaining data consistency. The system uses definitions in ScrapbookTypes.ts to ensure type safety across components and services, establishing a reliable foundation for layout configurations, photo analysis results, and user preferences.

4.3.2 Itinerary Planning Module

The itinerary creation and editing system is a lightweight module designed to support the overall travel system by transforming user preferences and requirements into personalized travel plans. While it leverages AI and machine learning for natural language processing, its primary goal is to provide adaptable itineraries without adding unnecessary complexity, ensuring smooth integration with the scrapbook for travelling system features.

Initial Data Collection and Preference Gathering

The system receives user inputs including travel style, budget, trip duration, destination in Sabah, and a brief trip description. Using this information, the system employs Gemini to automatically generate a personalized travel itinerary tailored to the user's preferences and requirements.

AI-Powered Itinerary Generation

Following data collection, the system receives user inputs including travel style, budget, trip duration, selected destinations in Sabah, and a brief trip description. These inputs are processed by GeminiService.ts, which interacts with the Gemini API to generate personalized travel itineraries. The service produces day-by-day plans that suggest attractions, activities, and basic travel arrangements, ensuring that the itinerary aligns with the user's preferences and trip requirements while providing a practical guide for their visit.

```
  static async generateItinerary(userId: string, preferences: ItineraryPreferences): Promise<Itinerary> {
    try {
      // Create a more concise prompt to avoid truncation
      const prompt = `
Create a ${preferences.duration}-day itinerary for ${preferences.destination}.
Style: ${preferences.style}, Budget: ${preferences.budget}
Description: ${preferences.description}

Each day: 3 meals + 2-3 activities. Keep descriptions brief (max 50 chars).
Return ONLY valid JSON, no markdown:

{
  "days": [
    {
      "day": 1,
      "activities": [
        { "time": "8:00 AM", "activity": "Breakfast", "venue": "Local Cafe", "description": "Traditional dishes" },
        { "time": "10:00 AM", "activity": "City tour", "venue": "Old Town", "description": "Explore landmarks" },
        { "time": "1:00 PM", "activity": "Lunch", "venue": "Restaurant", "description": "Local cuisine" },
        { "time": "3:00 PM", "activity": "Museum", "venue": "History Museum", "description": "Learn history" },
        { "time": "7:00 PM", "activity": "Dinner", "venue": "Night Market", "description": "Street food" }
      ]
    }
  ]
}`;

      // Call Gemini with increased token limit for multi-day itineraries
      const response = await generateText(prompt, {
        maxOutputTokens: 3500,
        temperature: 0.7
      });

      // Parse JSON from Gemini output (handle markdown code blocks)
      let parsed: { days: ItineraryDay[] } = { days: [] };
      let jsonContent = response.trim();
```

Figure 4.4 Prompt text that pass to Gemini to generate itinerary

Itinerary Structuring

The ItineraryService.ts is responsible for structuring the generated itinerary data into a clear day-by-day format. It organizes the user's travel plan into daily schedules, arranging activities, destinations, and basic suggestions in a logical sequence to provide a coherent and easy-to-follow itinerary.

Itinerary Editing and Customization System

Interactive Itinerary Editor

The itinerary editing capabilities are primarily managed through the ItineraryModule.tsx component, which enables for customizing their itineraries after initial generation. User can use Activity button to add activity but input the time, destination and description. Besides, users can also use the edit toggle button to edit a specific activity or delete toggle button to delete the specific activity. The itinerary detail page maintains a live preview system that immediately reflects user changes, providing instant feedback on customization decisions.

AI Assistant Chat Interface

The chat interface component provides users with editing assistance through an interactive chat interface specifically designed for itinerary modifications. This component implements natural language processing capabilities that allow users to request specific itinerary changes through conversational commands. Users can ask the AI to "add more cultural activities" or "Swap Day 2 and day 3 activities," and the system will process these requests and implement the changes.

4.3.3 Preference Weight Adjustment Model

The Preference Weight Adjustment Model is designed using a weight-based personalization mechanism that adapts scrapbook generation to user preferences over time. Each available layout (e.g., Grid, FullWidth, Mixed, CategoryBased) and font style option is assigned an initial base weight. When auto-generating a scrapbook, the system applies a **weighted random selection algorithm**, similar to a roulette wheel, to choose the layout and font combination.

The probability of selecting an option is calculated using the formula:

$$P(option_i) = \frac{W_i}{\sum_{j=1}^{n} W_j}$$

Where:

- $P(option_i)$ = probability of selecting option i
- W_i = weight assigned to option i
- $\sum_{j=1}^{n} W_j$ = total weight of all options

Each time a user edits and saves a scrapbook, the selected layout and font style receive a reward (+1 weight), increasing the probability of that option being chosen in future generations. Over time, this reinforcement learning approach captures the user's style preferences and continuously adapts scrapbook generation to reflect them.

The roulette-wheel style selection implements this weighted randomization by treating each option as a segment proportional to its weight. During scrapbook generation, the system "spins the wheel" to make a random choice, with higher-weighted options occupying larger segments and thus having a greater chance of selection. This ensures a balance between preference reinforcement and exploration, allowing less frequently used layouts to occasionally appear while favoring those the user prefers.

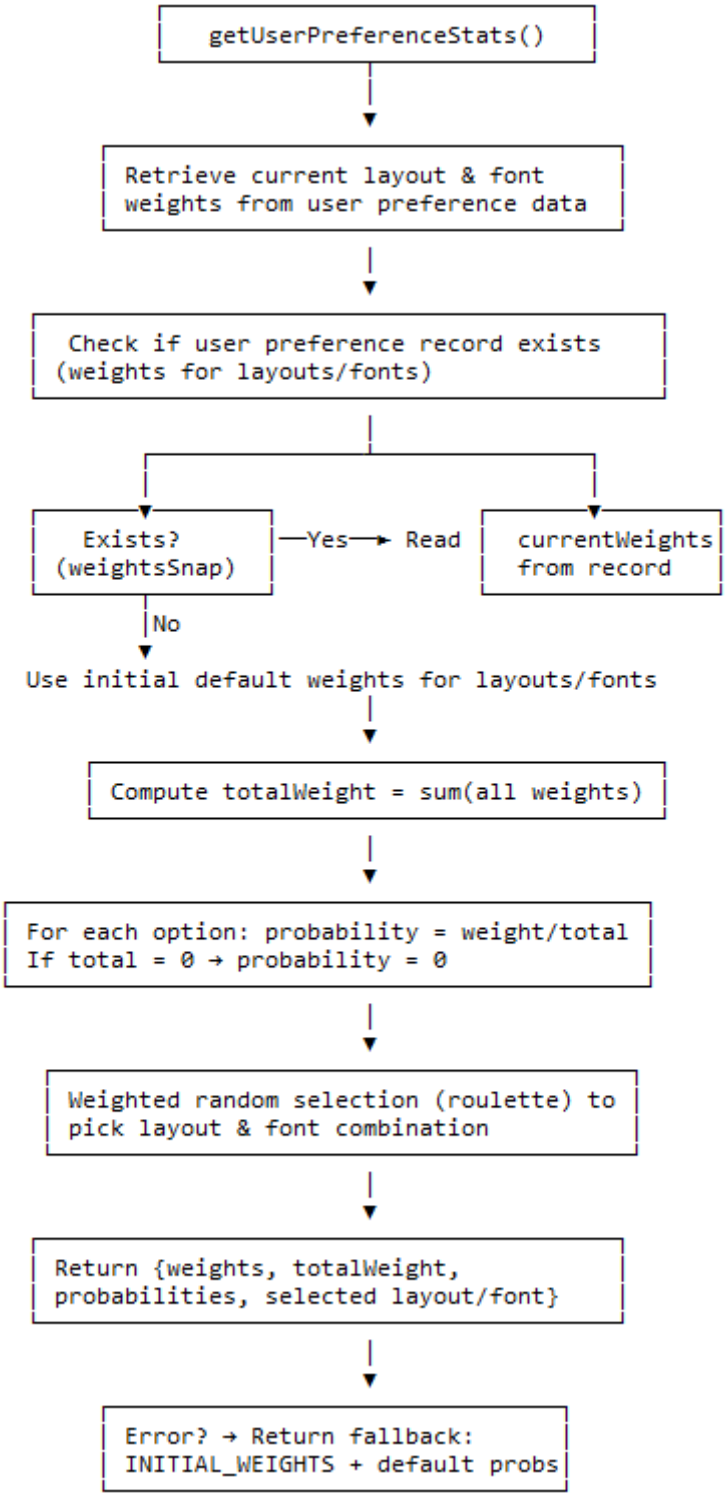

Figure 4.5 Architecture flow of Preference Weight Adjustment Model

Example Scenario Case:

- Initial weights: Grid = 3, FullWidth = 2, Mixed = 3, CategoryBased = 2 (Total = 10)
- Initial probabilities: Grid = 30%, FullWidth = 20%, Mixed = 30%, CategoryBased = 20%

Case 1: User saves with Grid → Grid weight = 4 (Total = 11)

- New probabilities: Grid = 36%, FullWidth = 18%, Mixed = 27%, CategoryBased = 18%

Case 2: User saves with Mixed → Mixed weight = 4 (Total = 12)

- New probabilities: Grid = 33%, FullWidth = 17%, Mixed = 33%, CategoryBased = 17%

This iterative reinforcement allows the system to evolve with each user interaction.

Key Design Features:

1. **Edit Capture** – User modifications (layouts, fonts, colors) are logged each time a scrapbook is saved.
2. **Dynamic Weight Update** – Weights for the selected options are incremented, reinforcing user preferences over time.
3. **Weighted Random (Roulette-Wheel) Selection** – The scrapbook engine uses a roulette-wheel style algorithm, where each layout or font occupies a segment proportional to its weight; higher-weighted options are more likely to be chosen, but less frequently used options still have a chance, balancing personalization and diversity.
4. **Storage Strategy** – Updated weights are stored locally as a lightweight module to support travel system responsiveness and can be synchronized if needed for cross-device consistency.

Through this design, the model ensures a balance between **personalization** and **diversity** (still allowing less frequently chosen layouts to appear occasionally), resulting in a more engaging and evolving scrapbook creation experience.

Layout Preference Component

This component manages scrapbook layout personalization. Each time a scrapbook is saved, the chosen layout is rewarded with an increased weight, representing the user's implicit preference. Over time, this weight-adjustment model biases layout selection towards the user's most frequently chosen layouts. The algorithm ensures adaptability — users' changing habits naturally shift the distribution of layout probabilities.

```
export const saveScrapbookAndUpdateWeights = async (
  state: ScrapbookState
): Promise<void> => {
  try {
    // 1. Save scrapbook final state
    await addDoc(collection(db, "scrapbook_states"), {
      ...state,
      savedAt: serverTimestamp(),
    });

    console.log(`Scrapbook saved for ${state.userId}, layout=${state.layout}`);

    // 2. Fetch current weights
    const weightsRef = doc(db, "user_layout_weights", state.userId);
    const weightsSnap = await getDoc(weightsRef);

    let currentWeights: Record<LayoutType, number> = { ...INITIAL_WEIGHTS };
    let totalSaved = 0;

    if (weightsSnap.exists()) {
      const data = weightsSnap.data() as UserLayoutWeights;
      currentWeights = data.layoutWeights;
      totalSaved = data.totalSavedScrapbooks;
    }

    // 3. Reward the chosen layout (+1)
    currentWeights[state.layout] = (currentWeights[state.layout] || 0) + 1;
    totalSaved += 1;

    // 4. Save updated weights back to Firestore
    await setDoc(weightsRef, {
      layoutWeights: currentWeights,
      totalSavedScrapbooks: totalSaved,
      updatedAt: serverTimestamp(),
    });

    console.log(`Updated layout weights:`, currentWeights);
  } catch (error) {
    console.error("Error saving scrapbook:", error);
    throw error;
  }
};
```

Figure 4.6 Update Layout Weight Function (Triggered when user save scrapbook)

Font and Color Preference Component

This component adapts the scrapbook's typography and color schemes based on category-specific user edits. Each confirmed choice of font family or color increments the associated weight. Over time, the system builds a probabilistic model of stylistic preferences per category (e.g., "Nature" photos leaning towards green tones). This enhances personalization while ensuring visual consistency with user preference.

```
export const updateFontPreference = async (
  userId: string,
  category: PhotoCategory,
  chosenFont: string,
  chosenColor: string
): Promise<void> => {
  try {
    const ref = doc(db, CATEGORY_FONT_COLLECTION, userId);
    const snap = await getDoc(ref);

    const current: UserCategoryFontPrefs = snap.exists()
      ? (snap.data() as UserCategoryFontPrefs)
      : { categories: deepCloneDefaults(), totalEdits: 0 };

    // Ensure the target category exists in user preferences,
    // fall back to default if missing
    const cat = current.categories[category] || { ...categoryFontDefaults[category] };

    // Handle chosen font preference:
    // - Skip if the font is banned (e.g., legacy or disallowed font which is unclear for some cases)
    // - Otherwise, increase weight (frequency) for the chosen font
    if (!isBannedFont(chosenFont)) {
      const safeWeights = filterBanned(cat.fontWeights); // remove banned fonts from weight map
      safeWeights[chosenFont] = (safeWeights[chosenFont] || 0) + 1; // increment count
      cat.fontWeights = safeWeights; // update font weights
    }
    // Handle chosen color preference: Increase weight (frequency) for the chosen color
    cat.colorWeights[chosenColor] = (cat.colorWeights[chosenColor] || 0) + 1;

    // Save updated category back to user's preferences
    current.categories[category] = cat;
    // Track how many edits user has made in total (used for personalization strength)
    current.totalEdits = (current.totalEdits || 0) + 1;

    await setDoc(ref, {
      categories: current.categories,
      totalEdits: current.totalEdits,
      updatedAt: serverTimestamp(),
    });
```

Figure 4.7 Update Font Weight Function (Triggered when user save scrapbook)

Statistical Tracking Component

This component provides analytics by converting raw layout weights into probabilities. It allows the system to display preference statistics, such as "User prefers Grid layout 45% of the time." This statistical layer supports evaluation, monitoring, and research insights, ensuring that personalization remains transparent and measurable.

```
export const getUserLayoutStats = async (userId: string): Promise<{
  weights: Record<LayoutType, number>;
  totalSaved: number;
  probabilities: Record<LayoutType, number>;
}> => {
  try {
    // Retrieve current layout weights for this user (from helper function or DB)
    const weights = await getLayoutWeights(userId);
    // Reference to Firestore document that stores user's layout weights
    const weightsRef = doc(db, "user_layout_weights", userId);
    // Fetch snapshot of document
    const weightsSnap = await getDoc(weightsRef);

    // Default totalSaved is 0; update if Firestore doc exists
    let totalSaved = 0;
    if (weightsSnap.exists()) {
      const data = weightsSnap.data() as UserLayoutWeights;
      totalSaved = data.totalSavedScrapbooks;
    }

    // Compute total weight (sum of all layout weights)
    const totalWeight = Object.values(weights).reduce((sum, w) => sum + w, 0);
    // Initialize probabilities object
    const probabilities: Record<LayoutType, number> = {} as Record<LayoutType, number>;

    // Calculate probability (%) for each layout type
    for (const [layout, weight] of Object.entries(weights)) {
      probabilities[layout as LayoutType] = totalWeight > 0 ? (weight / totalWeight) * 100 : 0;
    }

    return {
      weights,
      totalSaved,
      probabilities
    };
  } catch (error) {
    console.error("❌ Error getting layout stats:", error);
    return {
      weights: INITIAL_WEIGHTS,
      totalSaved: 0,
      probabilities: {
        grid: 30,
        full_width: 20,
        mixed: 30,
        category_based: 20
      }
    };
  }
};
```

Figure 4.8 Statistical Tracking for Layout Function

4.3.4 Sabah Activities Module

The Sabah Activity Page system represents a specialized component of the travel application that provides users with comprehensive access to the activities and attractions specific to the Sabah region of Malaysia. This system implements sophisticated activity discovery, categorization, and presentation mechanisms that enable users to explore and interact with destination-specific content. The architecture demonstrates advanced integration between activity management services, user interface components, and data presentation systems to create an engaging and informative activity browsing experience.

Activity Data Management and Presentation

The Sabah Activity Page system is primarily implemented through the Activities/page.tsx component, which serves as the main interface for users to access and browse Sabah-specific activities and attractions. This component integrates with ActivityService.ts to retrieve activity data, including activity name, descriptions, and location information, all sourced from static JSON files. The page implements sophisticated data presentation techniques that organize activities into logical categories and provide multiple browsing options for users. The ActivityService manages a structured dataset of Sabah activities, including nature-based attractions such as Mount Kinabalu trekking, wildlife viewing in Kinabatangan, and cultural experiences in Kota Kinabalu. It applies intelligent data structuring that categorizes activities based on type, difficulty level, duration, and seasonal availability, enabling users to filter and discover activities that match their interests and capabilities.

Combined Filtering System

The Sabah Activities Module features a dual-layer filtering system that combines category-based and location-based filters to provide a personalized browsing experience. This system dynamically refines the list of activities, attractions, and food options according to user preferences, making discovery both efficient and intuitive.

Each activity is tagged with one or more categories, such as Nature, Culture, Food, City, or Shopping. Selecting a category display

only activities matching that group, while “All” shows the full list, accommodating both single- and multi-category experiences.

Activities are also linked to specific areas in Sabah (e.g., Kota Kinabalu, Sandakan, Ranau, Tawau). Applying a location filter can narrow results to that region, helping tourists plan itineraries efficiently. The system evaluates both filters simultaneously, showing only activities that match the chosen category and location—for example, Nature activities in Ranau like Mount Kinabalu or Poring Hot Springs. Optimized rendering is implemented to ensure results update in real time without page reloads.

CHAPTER 5
System Implementation

5.1 Hardware Setup

The hardware involved in this project is computer and android mobile device The computer will be used for the development and testing phases of the project. The Android mobile device is essential for testing and deploying the travel application. It ensures that the app functions correctly across various Android devices and configurations. The mobile device will be used for field testing, evaluating user interactions, and validating application features.

Description	Specifications
Model	Lenovo Ideapad 3 Slim 3
Processor	AMD Ryzen 3 5300U
Operating System	Windows 10
Graphic	Radeon Graphcis
Memory	12.0 GB DDR4 RAM
Storage	256GB SSD

Table 5.1: Specifications of laptop

Description	Specifications
Model	Oppo Reno 6 Pro 5G
Processor	Qualcomm^{R} Snapdragon™ 870
Operating system	Android 12, ColorOS 12
RAM	12GB
Storage	256GB

Table 5.2: Specifications of Smartphone with Android OS

5.2 Software and Tools Setup

To develop and deploy the scrapbook and itinerary application, several tools and services were configured to create a stable development workflow, enable AI integration, and provide secure storage and deployment. This section explains the step-by-step setup of each tool used in the project.

5.2.1 Development Environment

1. Visual Studio Code (VS Code)

- Setup Steps:
    1. Download and install VS Code from https://code.visualstudio.com/.
    2. Install recommended extensions:
        - *ES7+React/Redux/React-Native snippets* for React productivity.
        - *Prettier* for code formatting.
        - *Tailwind CSS IntelliSense* for utility class suggestions.
        - *Firebase Tools* for easier backend integration.
    3. Configure VS Code settings for auto-save, linting, and integrated Git.
- Role in Project: Used as the primary IDE for managing both frontend and backend code, debugging, and running local servers.

5.2.2 Runtime & Package Management

Node.js & npm

- **Setup Steps**:
    1. Install Node.js (LTS version) from https://nodejs.org/.
    2. Verify installation using:
        - node -v
        - npm -v

3. Initialize project dependencies with:
    - npm install

- **Role in Project**: Provides runtime environment for React (Next.js), manages packages via package.json, and executes build/development scripts.

5.2.3 Frontend Development

React with TypeScript

- **Setup Steps**:
    1. Initialize project with Next.js + TypeScript:
    2. npx create-next-app@latest fyp --typescript
    3. Install additional React utilities (e.g., react-router-dom, uuid).
    4. Configure tsconfig.json for strict type safety.
- **Role in Project**: Powers the UI, scrapbook editor, and itinerary planner with strong type-checking for scalability.

Tailwind CSS

- **Setup Steps**:
    1. Install Tailwind CSS:
    2. npm install -D tailwindcss postcss autoprefixer
    3. npx tailwindcss init -p
    4. Configure tailwind.config.js with project paths.
    5. Import Tailwind styles in globals.css.
- **Role in Project**: Enables responsive, utility-first styling for scrapbook pages, forms, and layouts without writing raw CSS.

5.2.4 AI Integration

Gemini API (Google Generative AI)

- **Setup Steps**:
    1. Obtain an API key from Google AI Studio.
    2. Store API key securely in .env.local:
    3. NEXT_PUBLIC_GEMINI_API_KEY=your_api_key
    4. NEXT_PUBLIC_GEMINI_API_URL=https://generativelanguage.googleapis.com/v1beta/models/gemini-2.0-flash:generateContent
    5. Install Google's client library:
    6. npm install @google/generative-ai
    7. Configure a service adapter (e.g., GeminiApiAdapter.ts) to send requests.
- **Role in Project**: Provides AI-based text rewriting and photo caption generation for scrapbook pages.

5.2.5 Storage and Persistence

Local Storage

- **Role in Project**: Ensures temporary storage of user data and smooth updates to the backend.

Firebase

- **Setup Steps**:
    1. Create Firebase project in Firebase Console.
    2. Enable **Firestore**, **Authentication**, and **Storage**.
    3. Add environment variables in .env.local.
    4. Initialize Firebase in src/firebase.ts

5. Configure Firestore rules for user-scoped access.

- **Role in Project**: Provides secure cloud storage, user authentication, and cross-device synchronization of scrapbook and itinerary data.

5.2.6 Deployment and Version Control

Git & GitHub

- **Setup Steps**:
    1. Initialize repository:
    2. git init
    3. git remote add origin https://github.com/username/fyp.git
    4. Use .gitignore to exclude sensitive files (.env.local).
    5. Commit and push code regularly.
- **Role in Project**: Maintains version control, supports collaboration, and keeps history of development changes.

Vercel Deployment

- **Setup Steps**:
    1. Connect GitHub repo to Vercel.
    2. Configure environment variables in Vercel dashboard.
    3. Deploy automatically on git push.
- **Role in Project**: Provides production hosting with automatic CI/CD pipeline for Next.js app.

5.3 Setting and Configuration

The system was configured with a combination of runtime frameworks, dependencies, and cloud services to support the scrapbook and itinerary application. Next.js and React formed the core frontend framework, while Firebase provided authentication, real-time database, and cloud storage capabilities. Additional libraries and tools were integrated for AI-powered features, API communication, image handling, PDF export, UI styling, and development support. Environment variables and package dependencies were carefully managed to ensure a stable, secure, and scalable application setup.

5.3.1 Runtime, Dependencies, and Environment Configuration

Category	Details
Runtime/Framework	- **Next.js**: ^14.2.28 with React 18 and TypeScript 5 - **Tailwind CSS**: configured via tailwind.config.ts and src/app/globals.css
Key dependencies	- **Firebase** (auth, Firestore, storage, analytics) - **@google/generativeai** - **@google/genai** - **html2canvas/jspdf** (exporting) - **axios** - **lucide-react**
Environment variables	- Required for Gemini: NEXT_PUBLIC_GEMINI_API_KEY NEXT_PUBLIC_GEMINI_API_URL - Required for Firebase: NEXT_PUBLIC_FIREBASE_API_KEY NEXT_PUBLIC_FIREBASE_AUTH_DOMAIN NEXT_PUBLIC_FIREBASE_PROJECT_ID NEXT_PUBLIC_FIREBASE_STORAGE_BUCKET NEXT_PUBLIC_FIREBASE_MESSAGING_SENDER_ID NEXT_PUBLIC_FIREBASE_APP_ID NEXT_PUBLIC_FIREBASE_MEASUREMENT_ID

Table 5.3: Configuration List

5.3.2 Firebase Settings and Configuration

Firebase was integrated as the cloud backend service to support authentication, data storage, and real-time synchronization in the scrapbook application. Its modular services—Firestore, Storage, Authentication, and Analytics—were configured to ensure secure, scalable, and user-specific operations. The following sections detail the environment setup, initialization, security rules, and optimizations applied to each Firebase component.

Environment Variables

The Firebase configuration values were stored securely in the .env.local file to prevent direct exposure of sensitive credentials in the source code. The variables included the API key, authentication domain, project ID, storage bucket, messaging sender ID, app ID, and measurement ID. These values were injected into the application at runtime using the NEXT_PUBLIC_ prefix, enabling access in the Next.js frontend while maintaining a separation between development and production environments.

Firebase Initialization (src/firebase.ts)

A dedicated file, src/firebase.ts, was created to handle Firebase initialization. The configuration object referenced the environment variables, and Firebase services were initialized as singletons to avoid multiple instances during hot reloading in development. The following services were initialized and exported:

- Firestore: for structured, user-scoped database operations.
- Storage: for secure handling of scrapbook images and files.
- Authentication: for managing user sessions and access control.
- Analytics: for usage monitoring and custom event tracking.

This modular setup ensured reusability and avoided redundant initialization across the project.

Firestore Configuration

Firestore was configured as the main database service for storing users' scrapbooks, itineraries, user activities, and training data for scrapbook layout. Security rules, defined in firestore.rules, enforced strict user-scoped access. This configuration ensured data isolation between users, preventing unauthorized access or modification.

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    // Users can only access their own data
    match /scrapbooks/{scrapbookId} {
      // Allow read and write if user is authenticated and owns the scrapbook
      allow read, write: if request.auth != null &&
        (resource == null || request.auth.uid == resource.data.userId);

      // Allow create if user is authenticated and sets their own userId
      allow create: if request.auth != null &&
        request.auth.uid == request.resource.data.userId;
    }

    match /users/{userId} {
      allow read, write: if request.auth != null &&
        request.auth.uid == userId;
    }

    // Add rules for user_activities collection
    match /user_activities/{activityId} {
      // Allow read and write if user is authenticated and owns the activity
      allow read, write: if request.auth != null &&
        (resource == null || request.auth.uid == resource.data.userId);

      // Allow create if user is authenticated and sets their own userId
      allow create: if request.auth != null &&
        request.auth.uid == request.resource.data.userId;
    }

    // Add rules for itineraries collection
    match /itineraries/{itineraryId} {
      // Allow read and write if user is authenticated and owns the itinerary
      allow read, write: if request.auth != null &&
        (resource == null || request.auth.uid == resource.data.userId);

      // Allow create if user is authenticated and sets their own userId
      allow create: if request.auth != null &&
        request.auth.uid == request.resource.data.userId;
    }
```

Figure 5.1: Firestore Rules (a)

```
  // User actions (tracking editor events)
  match /user_actions/{actionId} {
    // Create: user must be signed-in AND set their own userId in the doc
    allow create: if request.auth != null
      && request.auth.uid == request.resource.data.userId;

    // Read/Update/Delete: only the owner
    allow read, update, delete: if request.auth != null
      && request.auth.uid == resource.data.userId;
  }
    // Users can read/write their own preferences (for training system)
  match /user_preferences/{userId} {
    allow read, write: if request.auth != null && request.auth.uid == userId;
  }

  // Layout training: allow user to manage their own layout weights
  match /user_layout_weights/{userId} {
    allow read, write: if request.auth != null && request.auth.uid == userId;
  }

  // Layout training: allow adding scrapbook state entries
  match /scrapbook_states/{docId} {
    allow create: if request.auth != null && request.resource.data.userId == request.auth.uid;
    allow read: if request.auth != null && resource.data.userId == request.auth.uid;
  }
    // Allow users to read/write their own activities (current collection name)
  match /user_activities/{activityId} {
    allow read, write: if request.auth != null &&
      (resource.data.userId == request.auth.uid ||
       request.resource.data.userId == request.auth.uid);
  }
  // Users can read/write their own itineraries
  match /itineraries/{itineraryId} {
    allow read, write: if request.auth != null &&
      (resource.data.userId == request.auth.uid ||
       request.resource.data.userId == request.auth.uid);
  }
  match /user_category_font_prefs/{userId} {
    allow read, write: if request.auth != null && request.auth.uid == userId;
  }
}
```

Figure 5.2: Firestore Rules (b)

Storage Configuration

Firebase Storage was used to store uploaded scrapbook data, itinerary data, user preference weight and related files. Storage rules enforced authentication, restricting read and write access to logged-in users only. The storage path structure grouped files under scrapbook IDs, ensuring logical organization and traceability of uploaded assets.

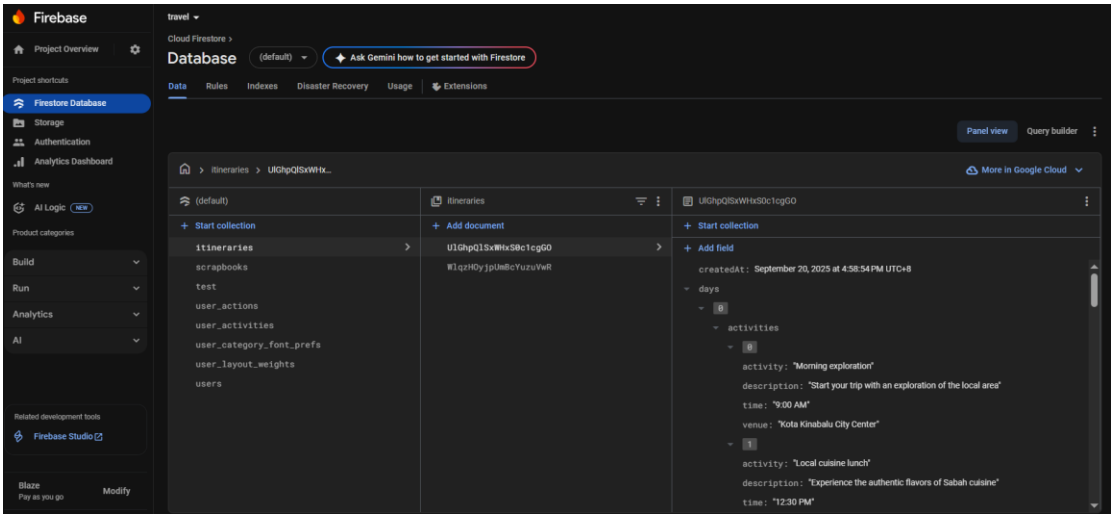

Figure 5.3: Firestore Database Collection

Authentication Configuration

Firebase Authentication was configured with Anonymous sign-in during the development phase to simplify testing. This allowed users to interact with Firestore and Storage without requiring third-party login providers. The authentication setup ensured that every user session was uniquely identified, enabling correct enforcement of Firestore and Storage rules.

Database Indexes

Composite indexes were created in Firestore to optimize query performance for common user actions. The following indexes were added:

- Scrapbooks by User and Creation Date: userId (ascending) + createdAt (descending).
- Scrapbooks by User and Start Date: userId (ascending) + startDate (ascending).
- Scrapbooks by User and Title: userId (ascending) + title (ascending).

These indexes supported efficient filtering and sorting of scrapbooks in the application interface.

5.3.3 Dependencies and Tools

The system relies on a combination of runtime dependencies, development dependencies, and supporting libraries. These are specified in the package.json file of the project.

- **Core Frameworks**
    - next (v14.2.28) – React-based framework for frontend and server-side rendering.
    - react (v18), react-dom (v18) – Core React libraries.
- **AI and API Integration**
    - @google/genai, @google/generative-ai – Gemini API client libraries for generative content.
    - axios – HTTP client for API calls.
    - dotenv – Environment variable management.
- **Database and Authentication**
    - firebase (v10.13.0) – Authentication, Firestore, and cloud storage integration.

- **UI and Styling**
    - tailwindcss, tailwind-merge, tailwindcss-animate – Utility-first CSS framework and styling helpers.
    - lucide-react – Icon library.
    - @radix-ui/react-slot, class-variance-authority, clsx – UI component composition and class utilities.
- **Image and Document Handling**
    - html-to-image, html2canvas – Convert DOM nodes into images.
    - jspdf – Export scrapbook/itinerary content to PDF.
    - react-avatar-editor – Image editing and cropping.
    - uuid – Unique identifier generation.
- **Development Tools**
    - typescript, eslint, eslint-config-next – Type checking and linting.
    - postcss, autoprefixer – CSS processing.

5.4 System Operation (with Screenshot)

5.4.1 Guest Mode & Authentication Pages

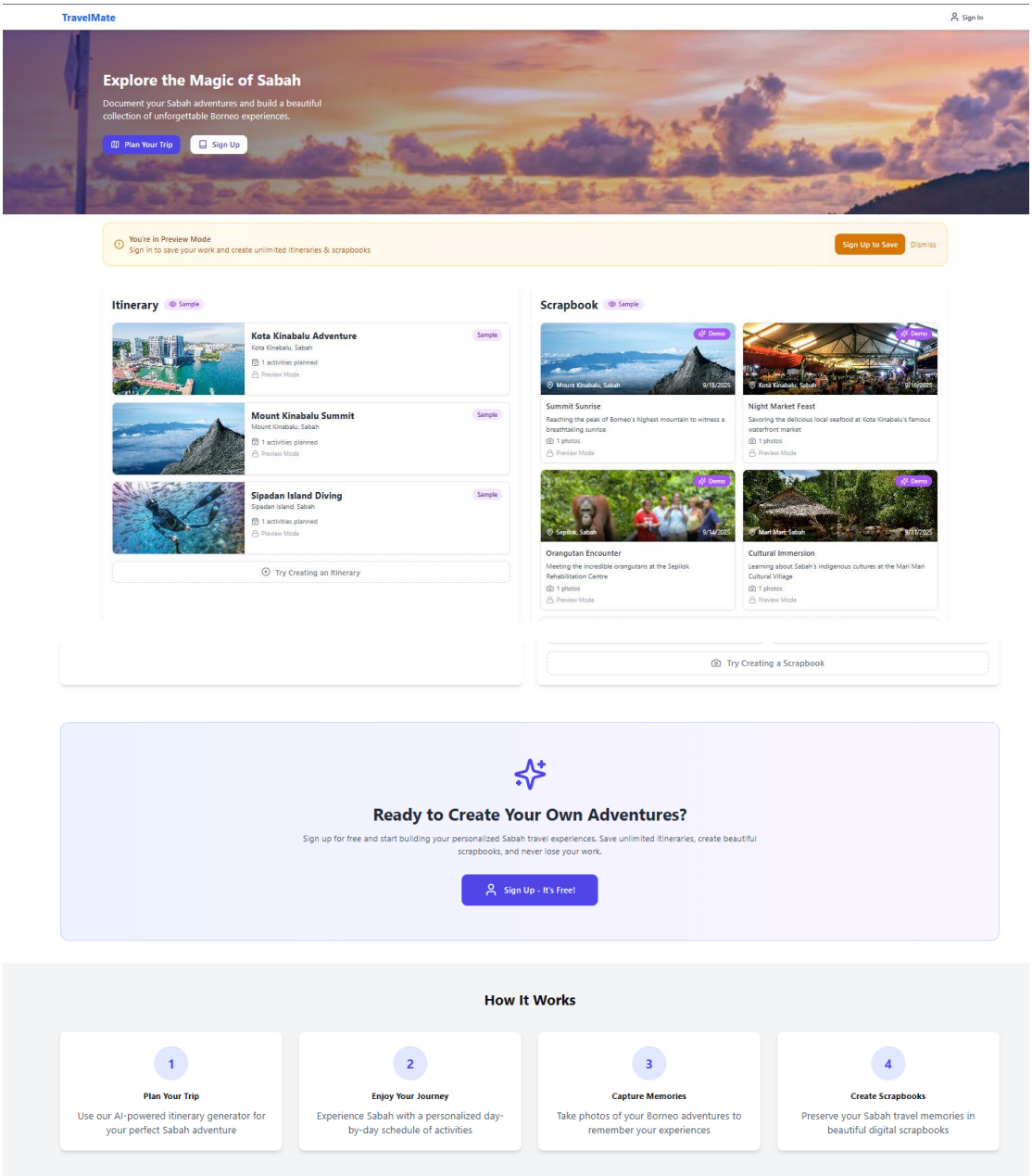

Figure 5.4: Guest Mode Home Page

The Guest Mode Home Page provides new users with a preview of the application while restricting access to full functionality. It serves as an entry point for first-time visitors, guiding them toward signing in or signing up. This design allows users to explore the interface before committing to registration, ensuring a smoother onboarding process.

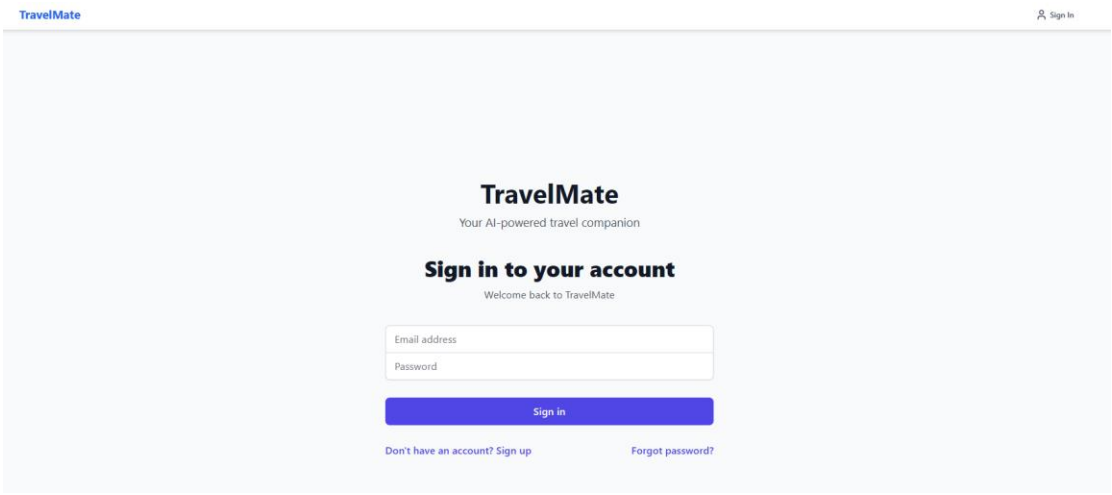

Figure 5.5: Sign in Page

The Sign In Page allows existing users to authenticate using their email and password credentials. It also includes options for creating a new account via the Sign-Up page or resetting a password if forgotten. This page is crucial for ensuring secure access and linking the system's personalization features to the correct user profile.

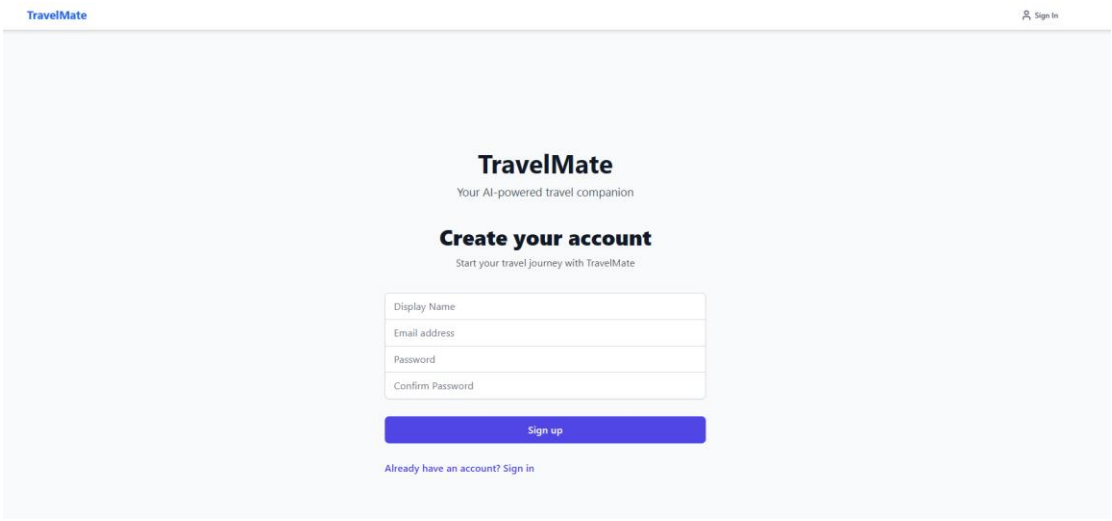

Figure 5.6: Sign up Page

The Sign-Up Page enables new users to create an account by entering their display name, password, and confirming the password. This ensures the creation of unique user profiles, which allow the system to store and retrieve scrapbook and itinerary data tied to individual accounts.

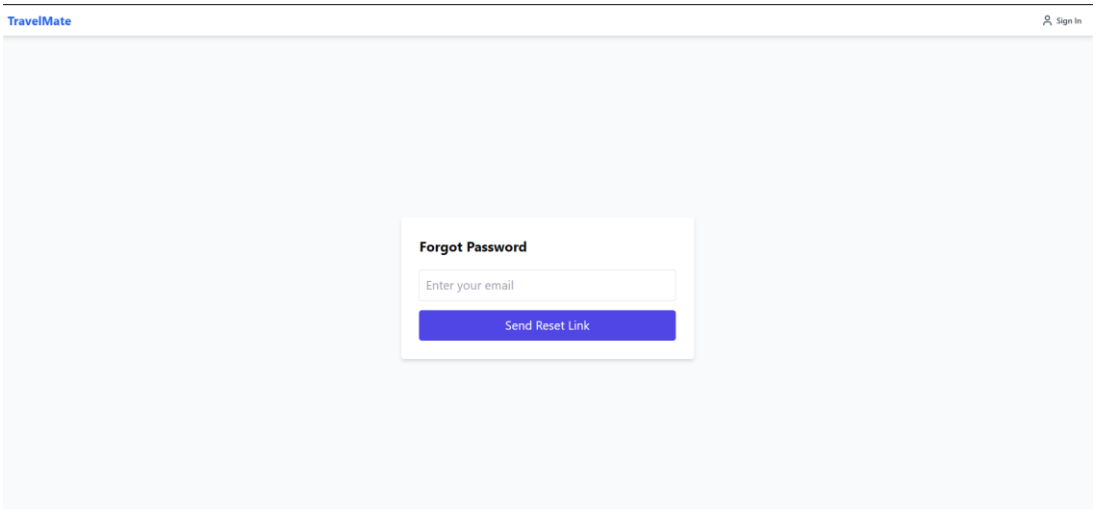

Figure 5.7: Forgot Password Page

The Forgot Password Page allows users to request a password reset by entering their registered email.

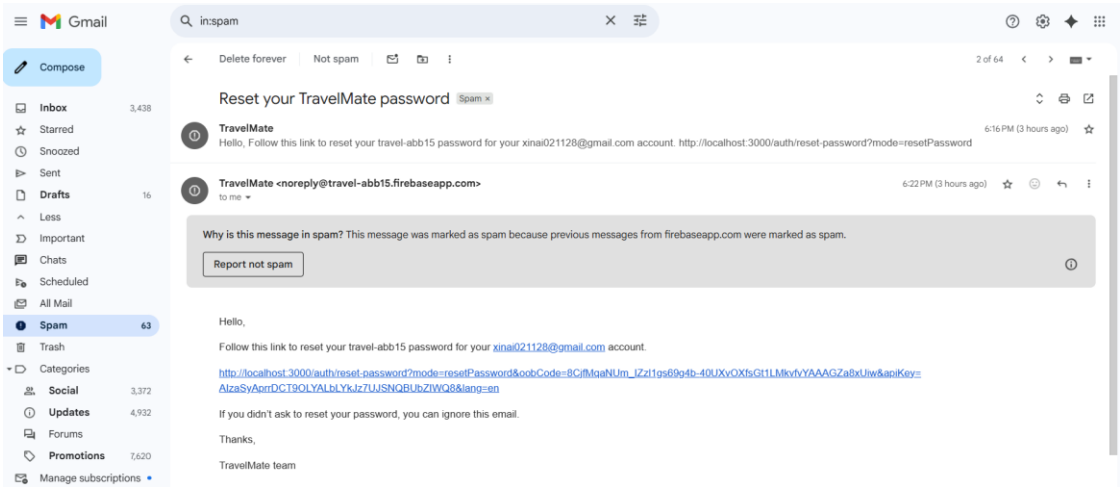

Figure 5.8: Reset Password Link Received at Email

Upon submitting the form, the system sends a password reset link via Firebase Authentication. This ensures secure password recovery and proper linkage to the user's profile.

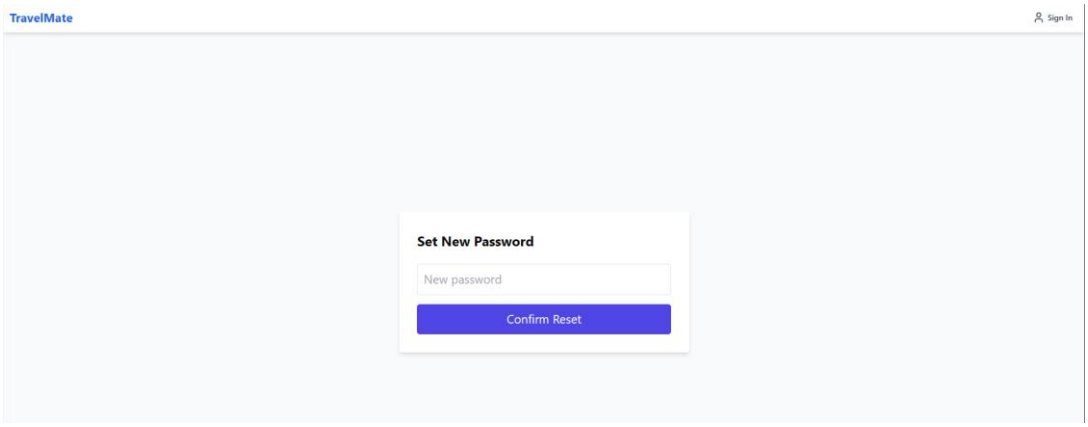

Figure 5.9: Set New Password Page

The Set New Password Page enables users to enter a new password after clicking the reset link received via email. Firebase verifies the link and securely updates the user's password. The page includes feedback messages and redirects users back to the Sign In Page after a successful reset.

5.4.2 Main User Pages

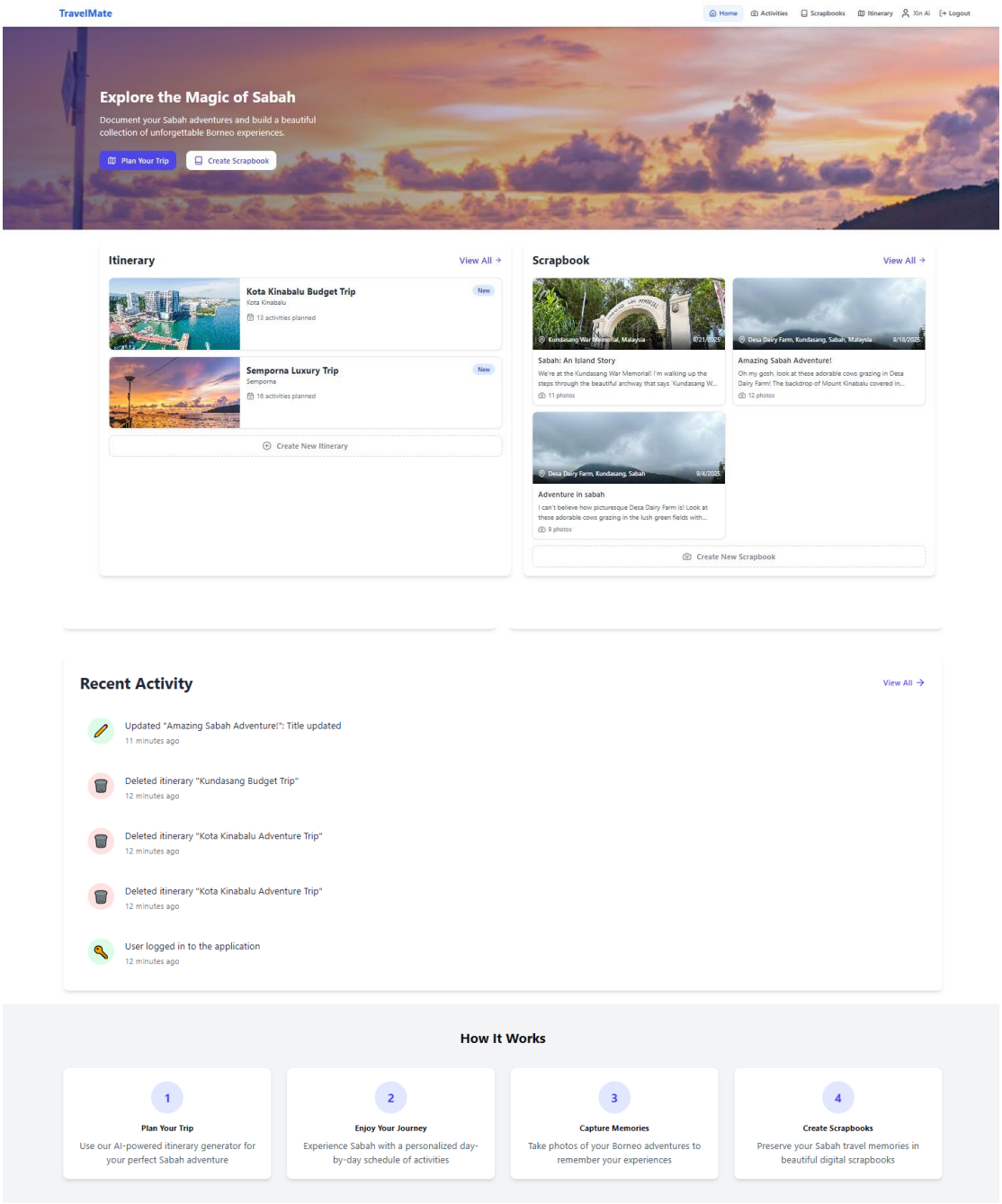

Figure 5.10: Home Page (Logged In)

The Home Page acts as the central dashboard after login. It provides users with an overview of their created itineraries and scrapbooks. By consolidating navigation into a single interface, the system ensures efficient access to both planning and documentation modules.

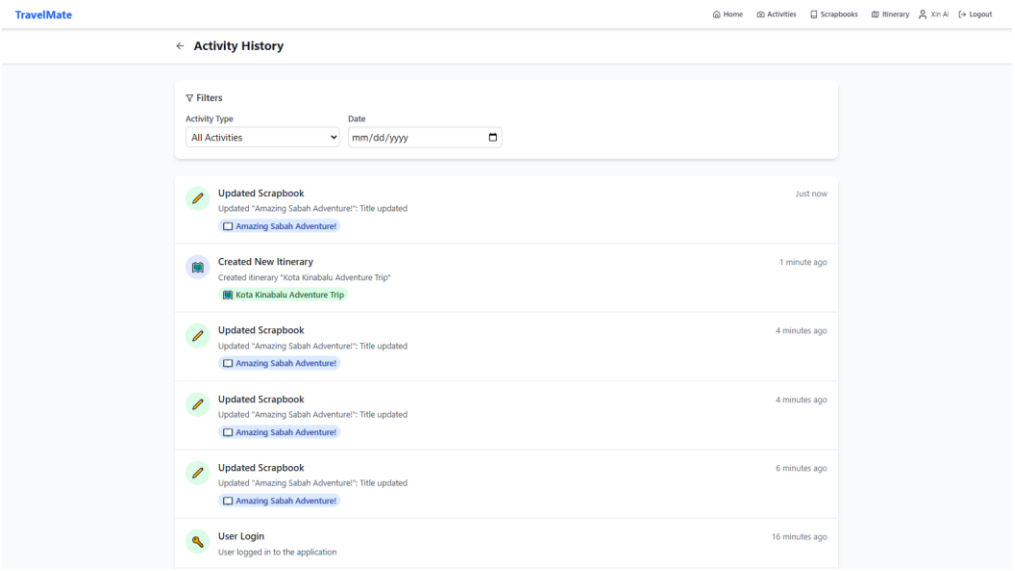

Figure 5.11: Recent Activity Page

The Recent Activity Page records all user actions, including scrapbook creation, updates, deletions, itinerary modifications, and login events. It includes filters for activity type and date, allowing users to track their interactions. This enhances transparency and helps users monitor their engagement with the system.

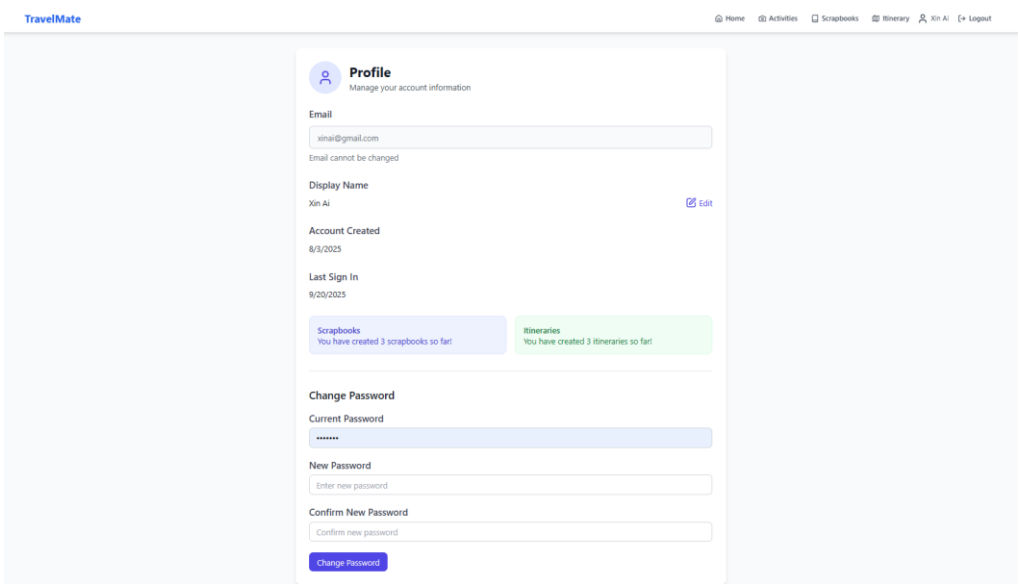

Figure 5.12: User profile Page

The User Profile Page enables users to manage their account settings. Core functions include updating the display name, changing the password, and viewing a summary of the total number of scrapbooks and itineraries created. This provides users with both personalization control and a quantitative overview of their activity.

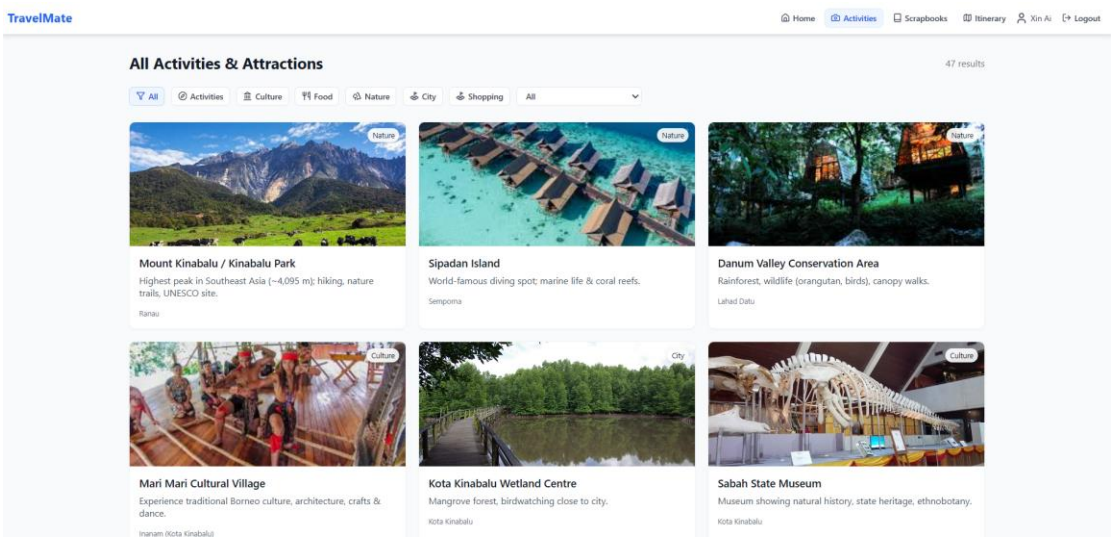

Figure 5.13: Sabah Activities Page

The Sabah Activities Page displays a comprehensive list of activities and attractions across Sabah.

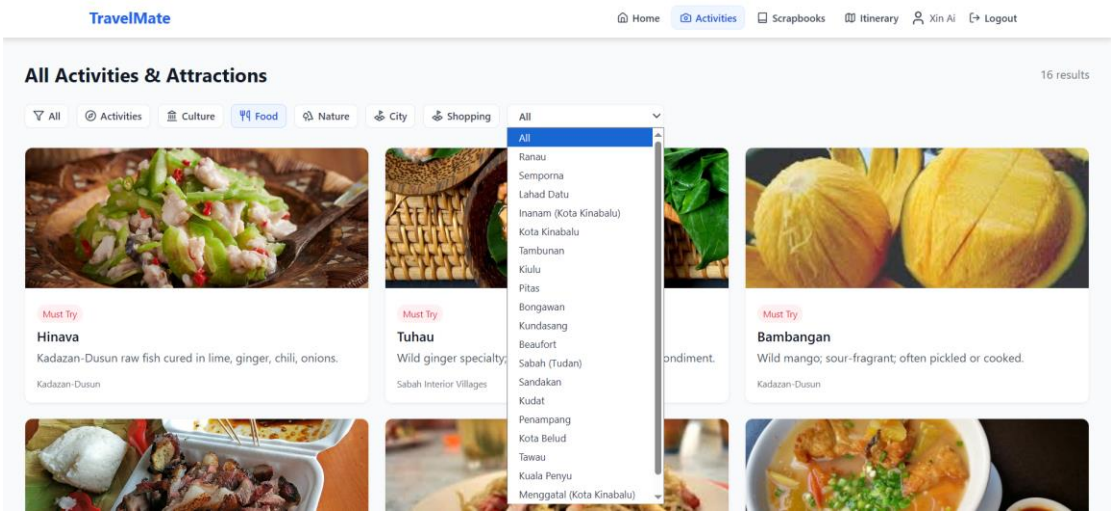

Figure 5.14: Sabah Activities Page with Filtering

Users can filter the content either by the type of activity or attraction, or by specific areas within Sabah, allowing for a more personalized browsing experience. The filtering functionality helps users quickly find points of interest relevant to their preferences and travel itinerary.

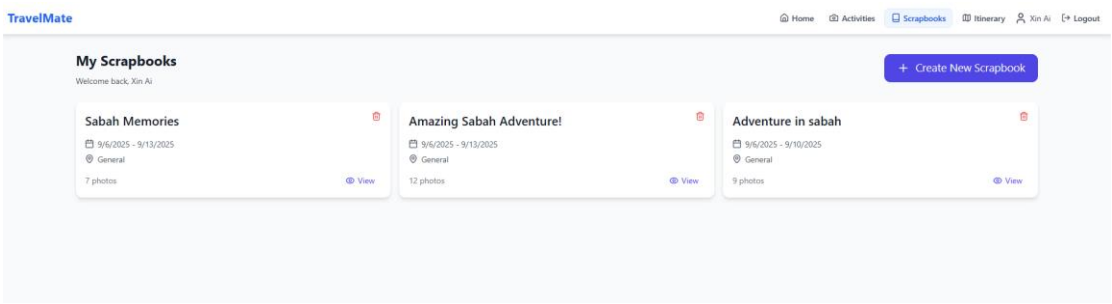

Figure 5.15: Scrapbook List Page

The Scrapbook List Page displays all scrapbooks created by the user. It includes functions to view or delete scrapbooks and provides access to create a new scrapbook. This page acts as the organizational hub for scrapbook-related activities.

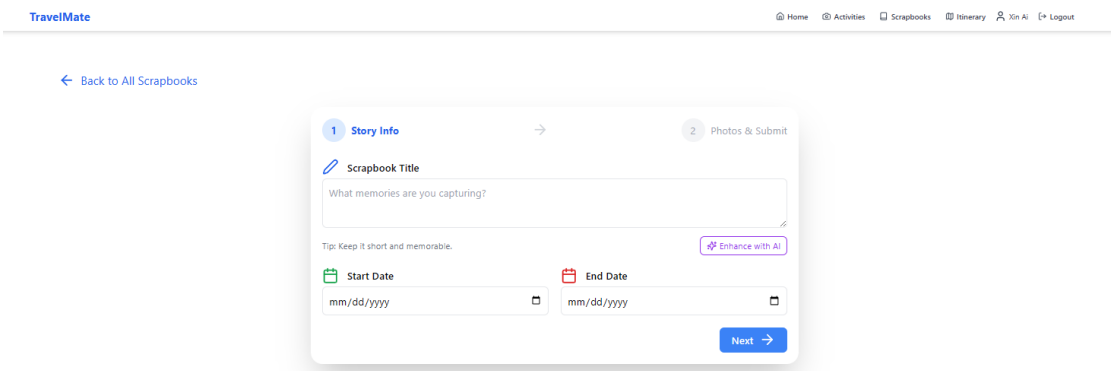

Figure 5.16: User Input Page (Create Scrapbook)

The User Input Page initiates scrapbook creation by prompting users to enter a scrapbook title and select the start and end travel dates. User can choose to enhance their scrapbook title with AI by clicking the Enhance with AI button. This information forms the contextual basis for scrapbook organization.

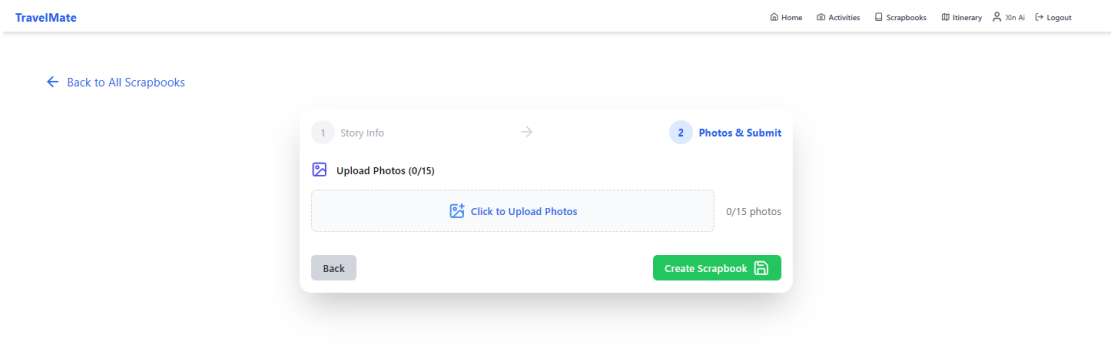

Figure 5.17: Photo Upload Page (Create Scrapbook)

The Photo Upload Page allows users to upload up to 15 photos for inclusion in the scrapbook. This input directly shapes the scrapbook content and provides the primary materials for layout generation.

5.4.3 Scrapbook Module Pages

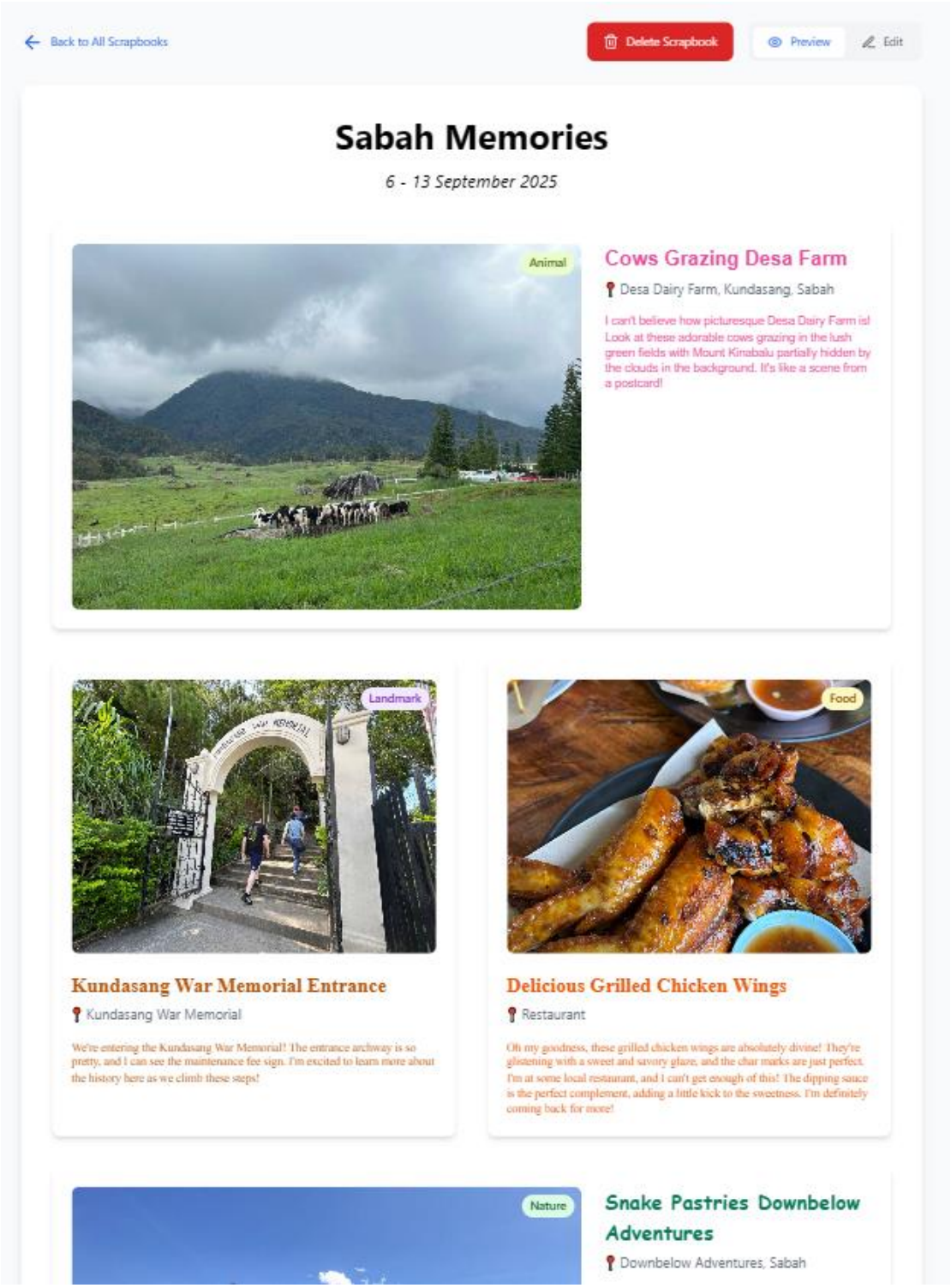

Figure 5.18: Scrapbook Preview Page

The Scrapbook Preview Page presents a near-final view of the scrapbook. At this stage, users can confirm the design and choose to export the scrapbook as a PNG file. This page bridges the creative process with the final output.

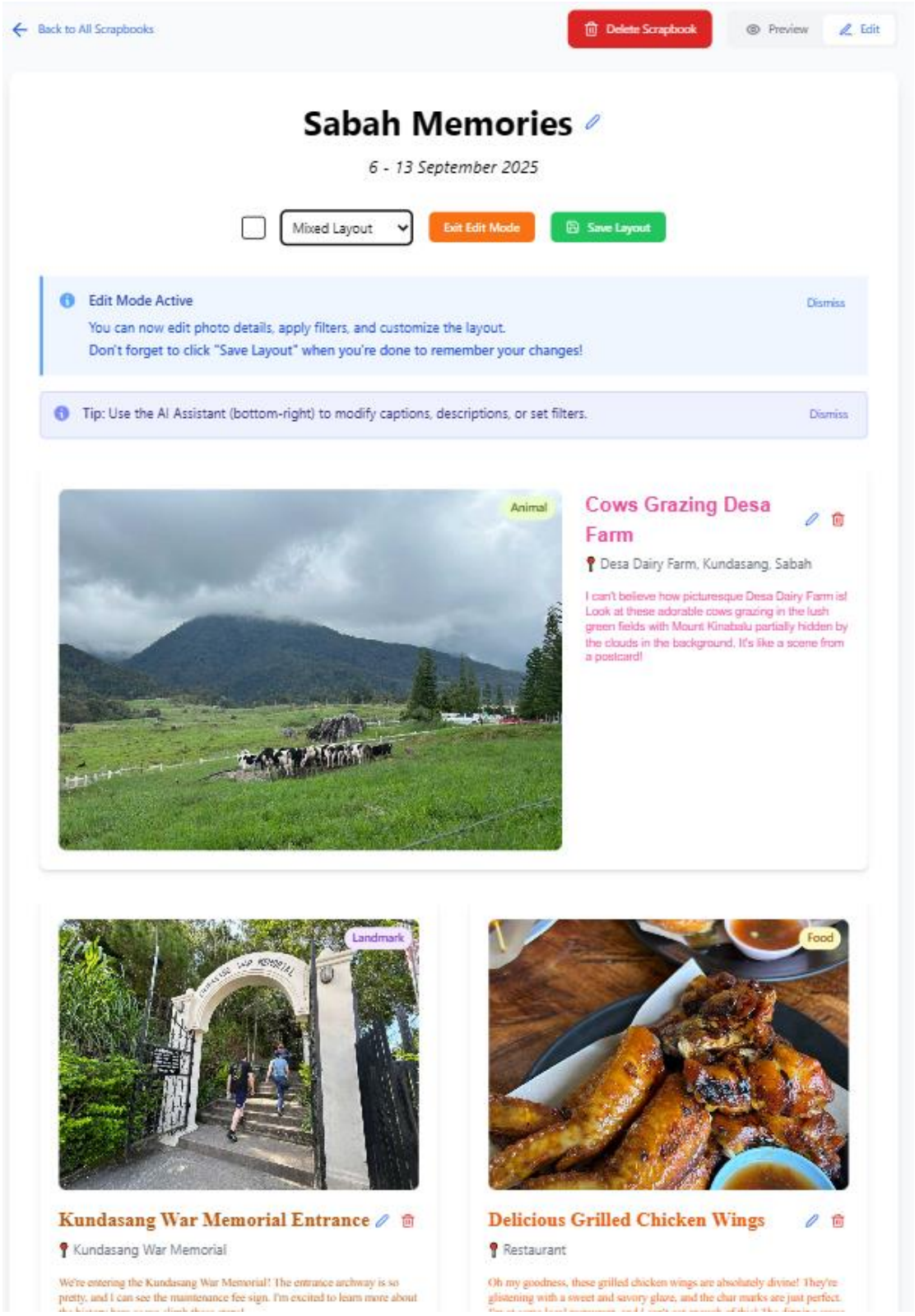

Figure 5.19: Scrapbook Edit Mode Page

The Scrapbook Edit Mode Page provides access to advanced editing functions, enabling users to modify layouts, captions, and visual presentation. It is designed for iterative refinement to achieve a personalized scrapbook.

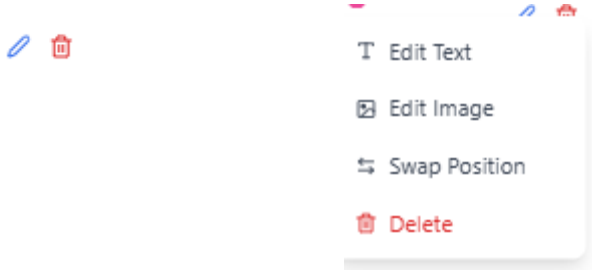

Figure 5.20: Edit Panel

The Edit Panel organizes editing tools into categories: text editing, image editing, swapping image positions, and deleting images. It streamlines the modification process by grouping functionalities in a structured interface.

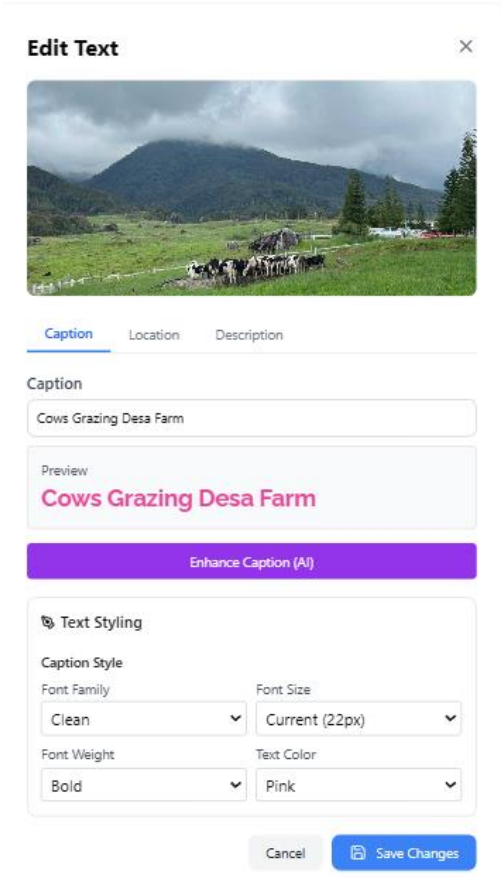

Figure 5.21: Edit Text Panel

The Edit Text Panel allows users to customize textual elements, including captions, location, and descriptions. Options for font family, size, weight, and color are provided via tab-based navigation, supporting personalization and readability. A live preview helps users visualize edits before applying them, ensuring better creative control.

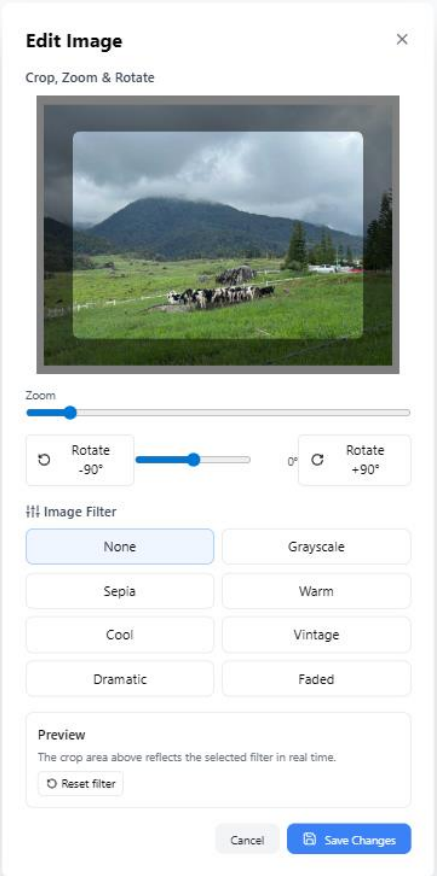

Figure 5.22: Edit Image Panel

The Edit Image Panel offers tools for cropping, zooming, rotating, and applying filters to images. A live preview helps users visualize edits before applying them, ensuring better creative control.

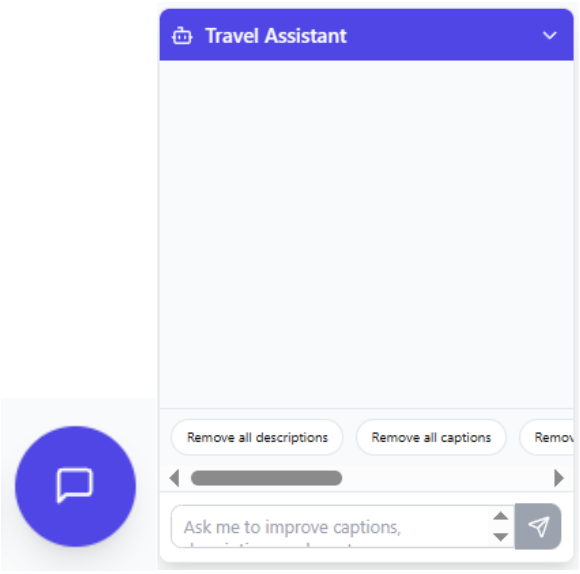

Figure 5.23: AI Assistant Chatbox

The Scrapbook AI Assistant Chatbox provides conversational support during scrapbook creation. User will access this chat box through a floating message button. It helps guide users through editing or layout adjustments, acting as a contextual helper.

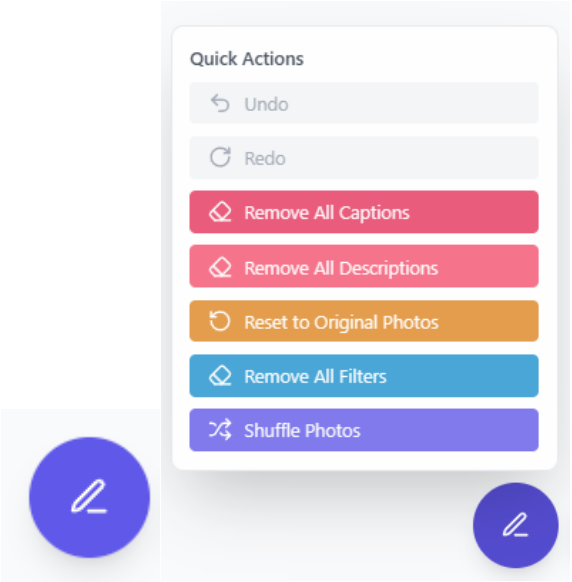

Figure 5.24: Quick Action Panel

The Quick Action Panel contains frequently used commands such as undo, redo, remove all text, reset photos to their original state, remove filters, and shuffle photo placements. Its floating button design ensures quick access without disrupting workflow.

Figure 5.25: Scrapbook Background Option

Figrue 5.26: Background design with Sabah Cultural Elements

This feature allows users to select scrapbook backgrounds, either plain colors or designs incorporating Sabah cultural elements. It enhances the aesthetic value and personalization of scrapbooks.

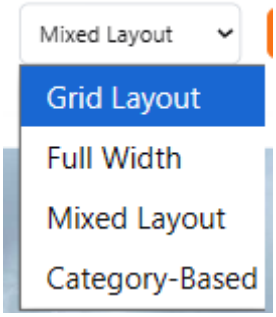

Figure 5.27: Scrapbook Layout Drop Down List

The Layout Drop Down List provides layout options such as Grid, Full Width, Mixed, and Category-Based. This selection directly affects the visual organization of scrapbook content.

5.4.4 Itinerary Module Pages

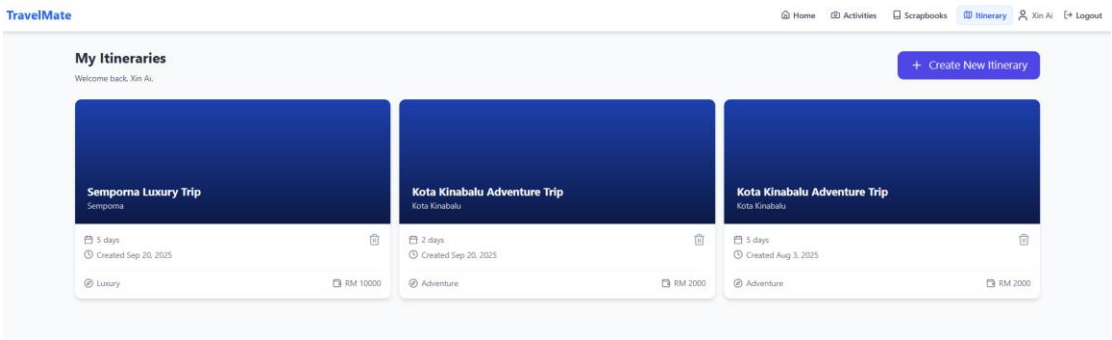

Figure 5.28: Itinerary List Page

The Itinerary List Page displays all itineraries created by the user. Similar to the scrapbook list, it includes options to view, delete, or create a new itinerary. It organizes itinerary management into one accessible hub.

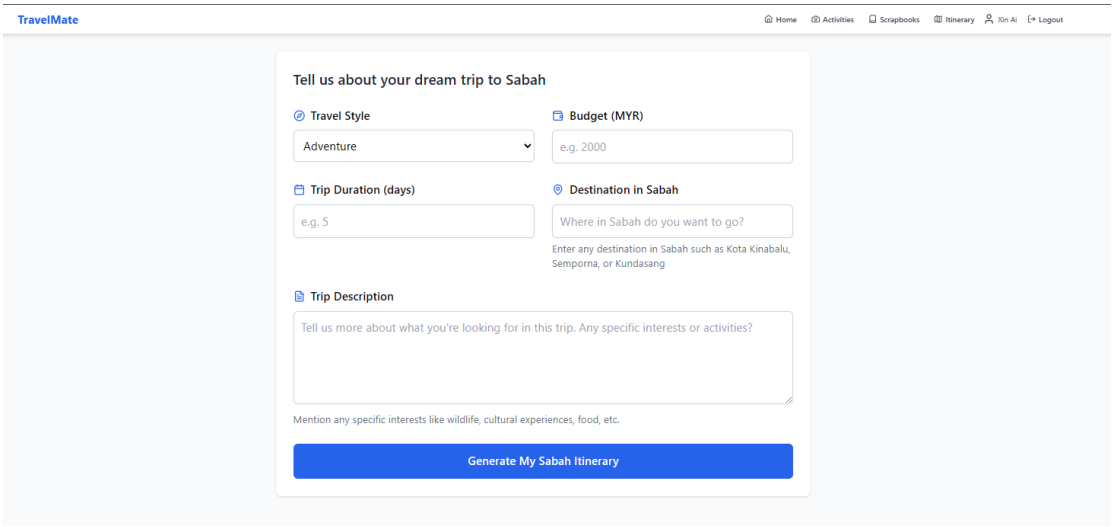

Figure 5.29: User Input Page (Create Itinerary)

The User Input Page for itineraries collects essential trip details such as travel style, budget, duration, destination in Sabah, and trip description. Once completed, the user can generate a personalized itinerary via the “Generate My Sabah Itinerary” button.

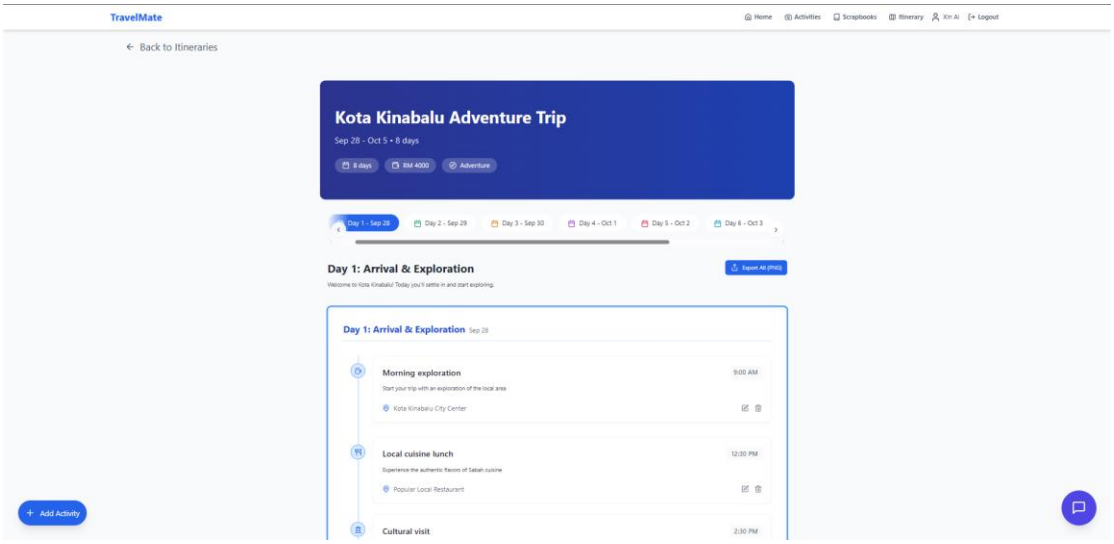

Figure 5.30: itinerary Preview Page

The Itinerary Preview Page allows users to review and edit the generated itinerary. Once satisfied, the itinerary can be exported as a PNG for offline use.

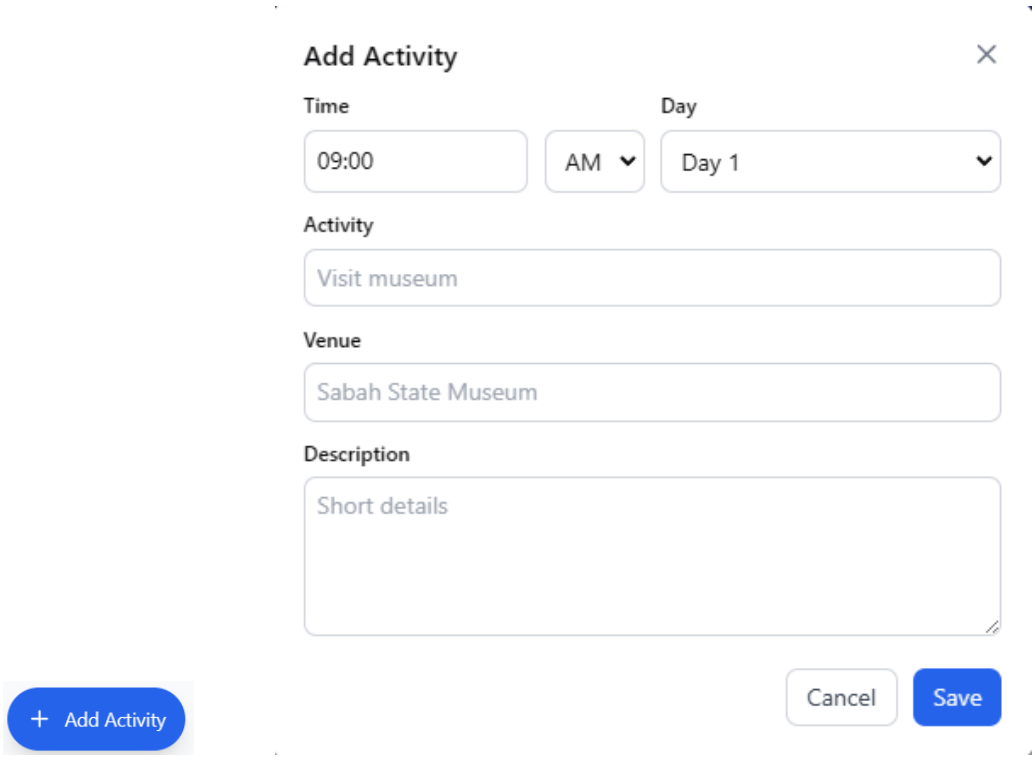

Figure 5.31: Add Activity Button and Card

The Add Activity Card feature lets users add custom activities through the Add Activity Button by specifying the time, day, activity name, venue, and description. This ensures flexibility and adaptability in itinerary design.

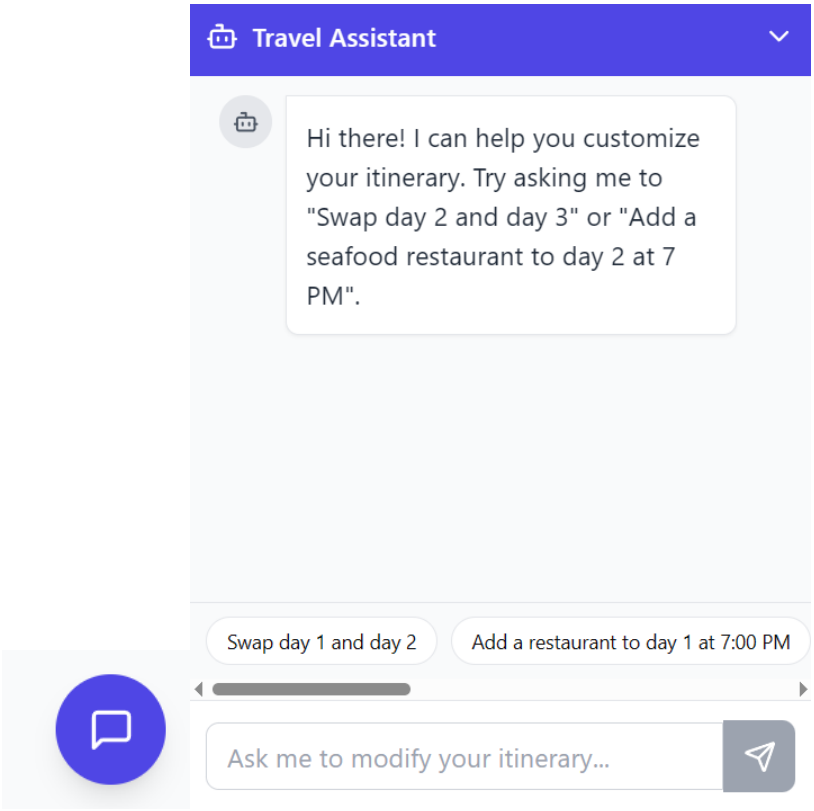

Figure 5.32: AI Assistant Chatbot (Itinerary)

The Itinerary AI Assistant Chatbox supports users in reviewing and adjusting itinerary content. User will access this chat box through a floating message button. It serves as a conversational guide, simplifying interaction with structured itinerary data.

5.4.5 UI for Mobile

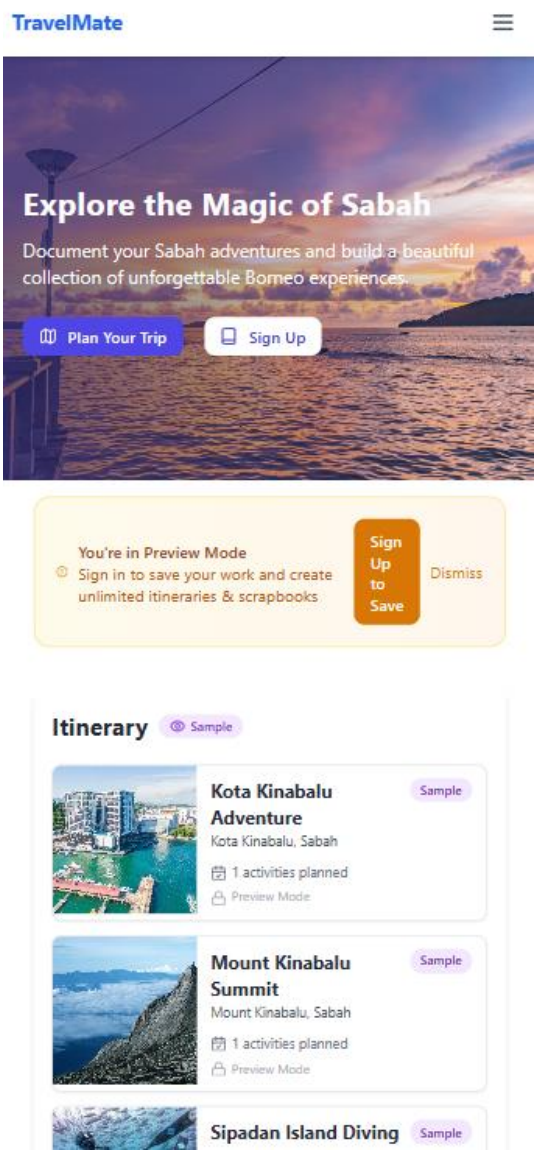

Figure 5.33: Home Page 1 (Guest Mode)

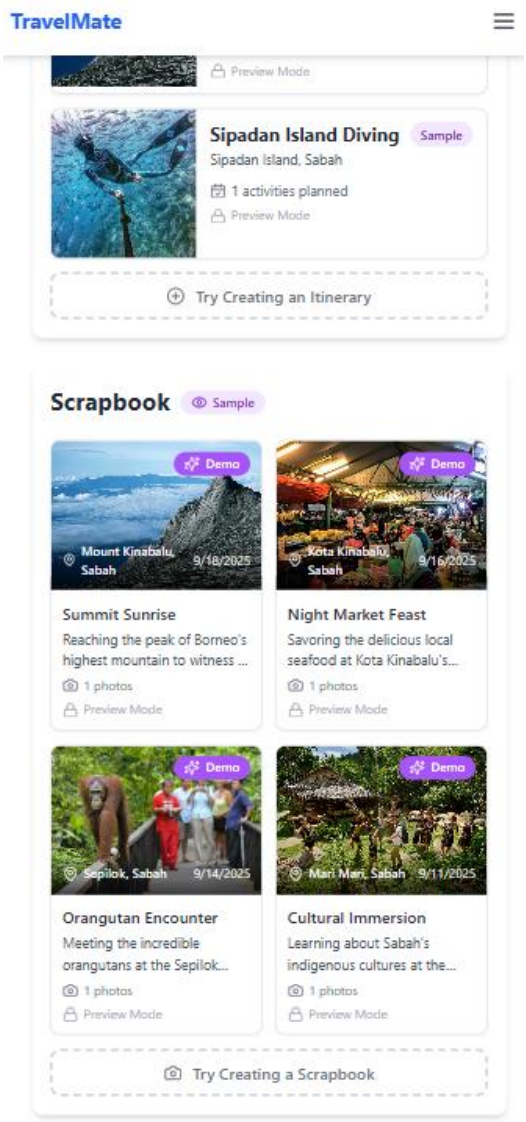

Figure 5.34: Home Page 2 (Guest Mode)

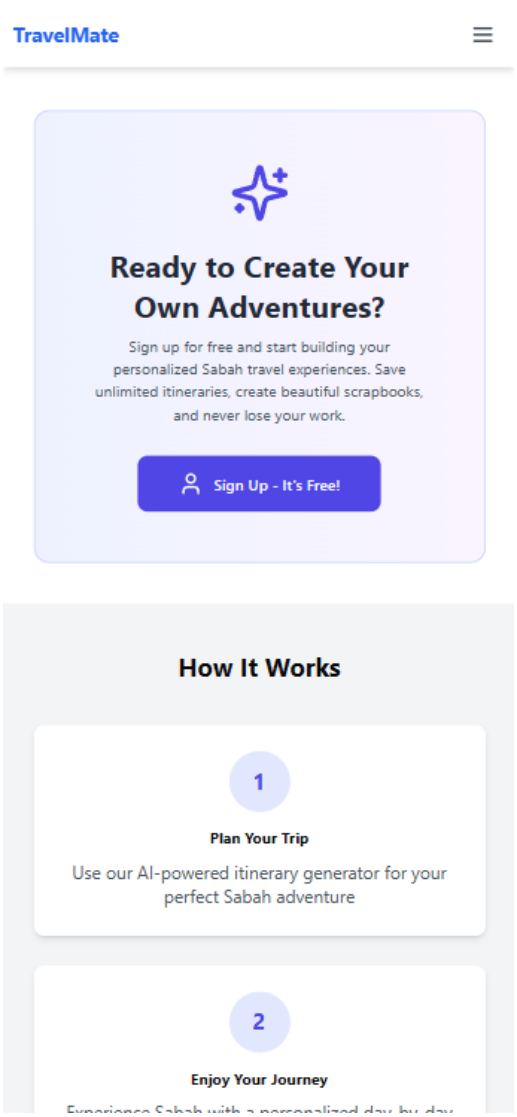

Figure 5.35: Home Page 3

(Guest Mode)

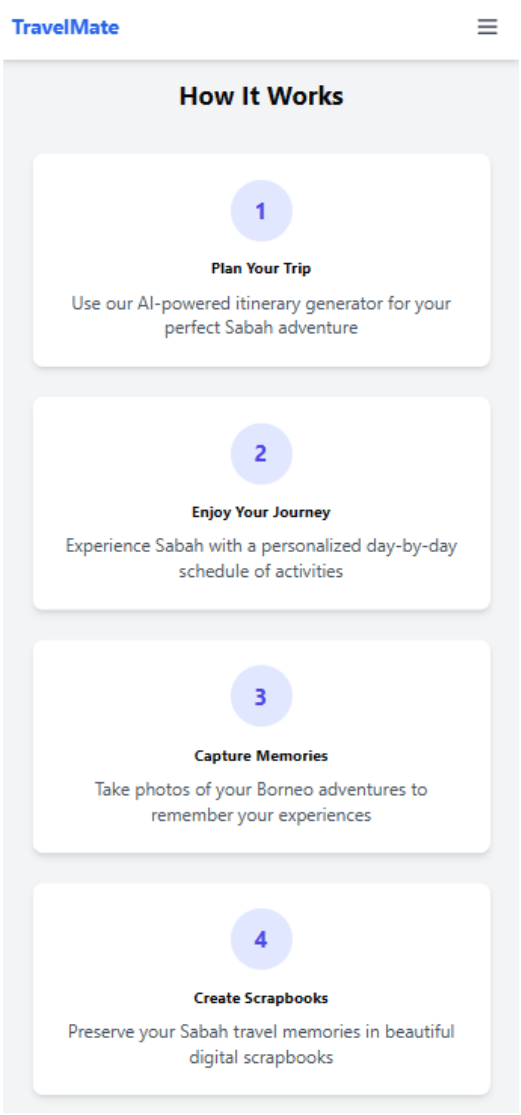

Figure 5.36: Home Page 4

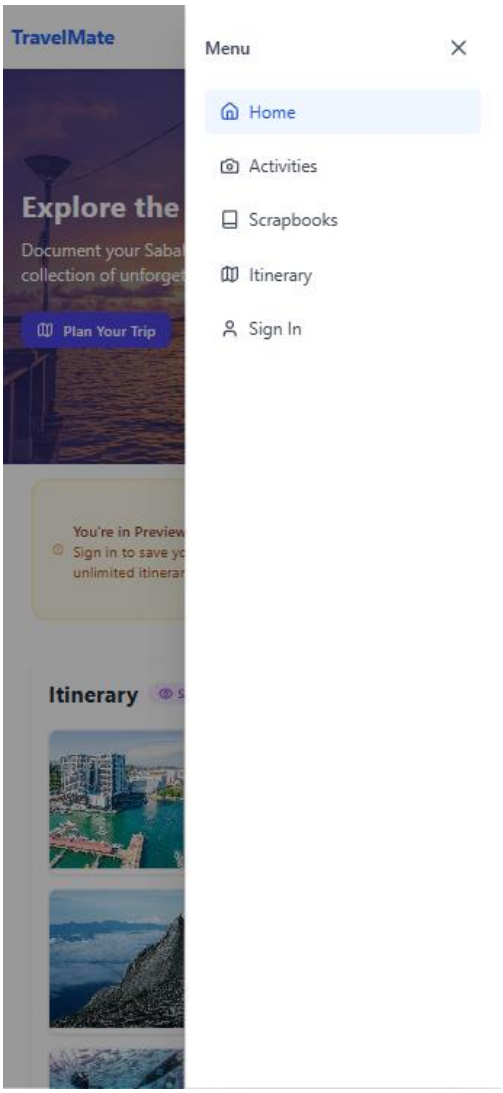

Figure 5.37: Navigation Bar

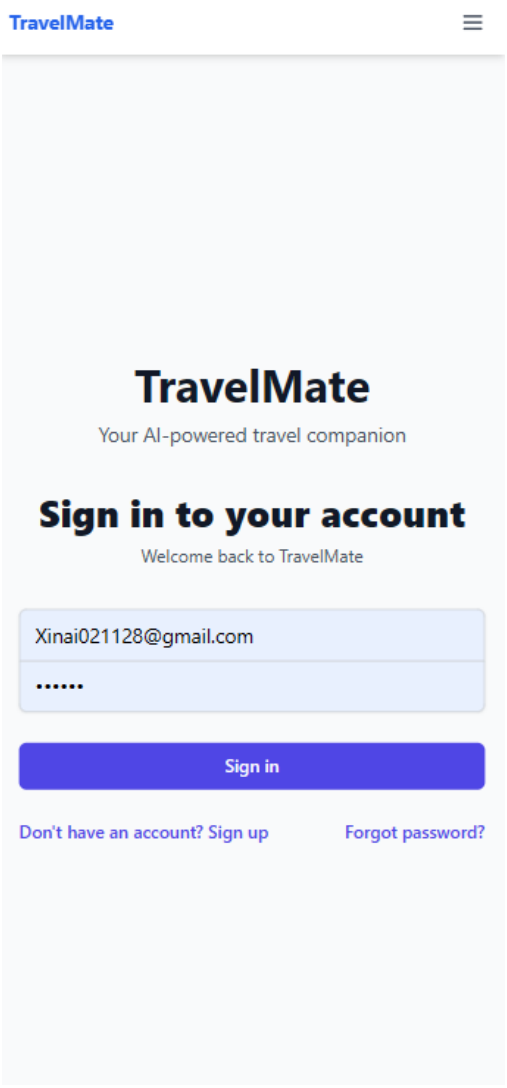

Figure 5.38: Sign In

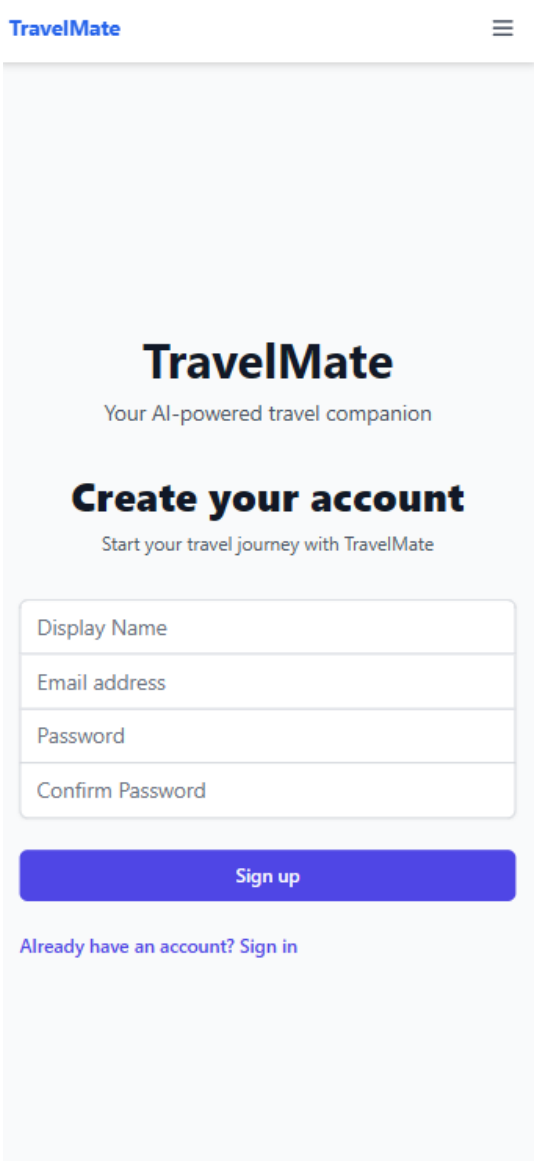

Figure 5.39: Sign Up

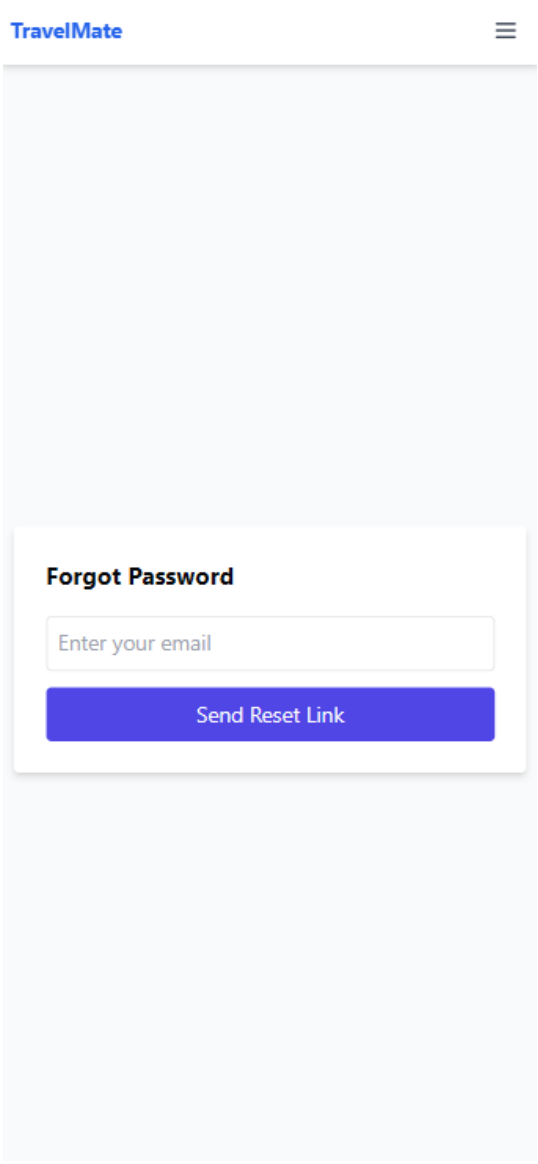

Figure 5.40: Forgot Password

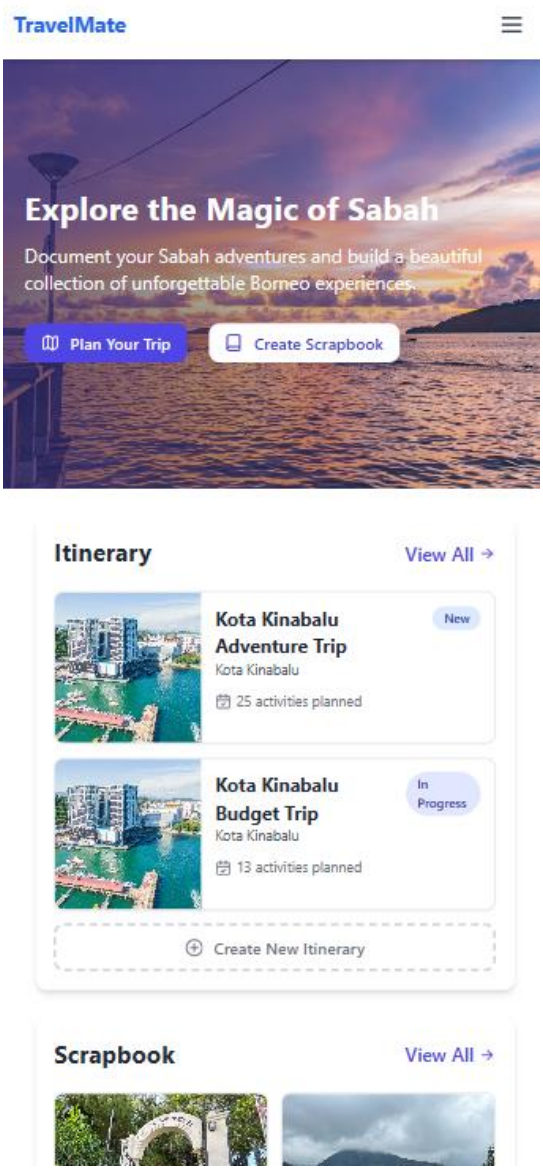

Figure 5.41: Home Page 1

(After user login)

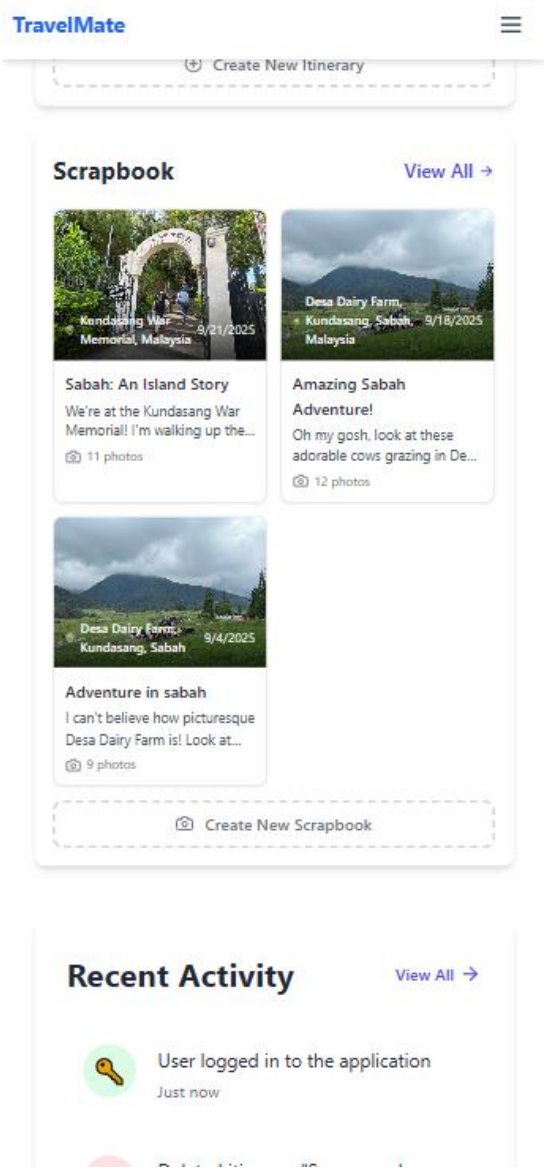

Figure 5.42: Home Page 2

(After user login)

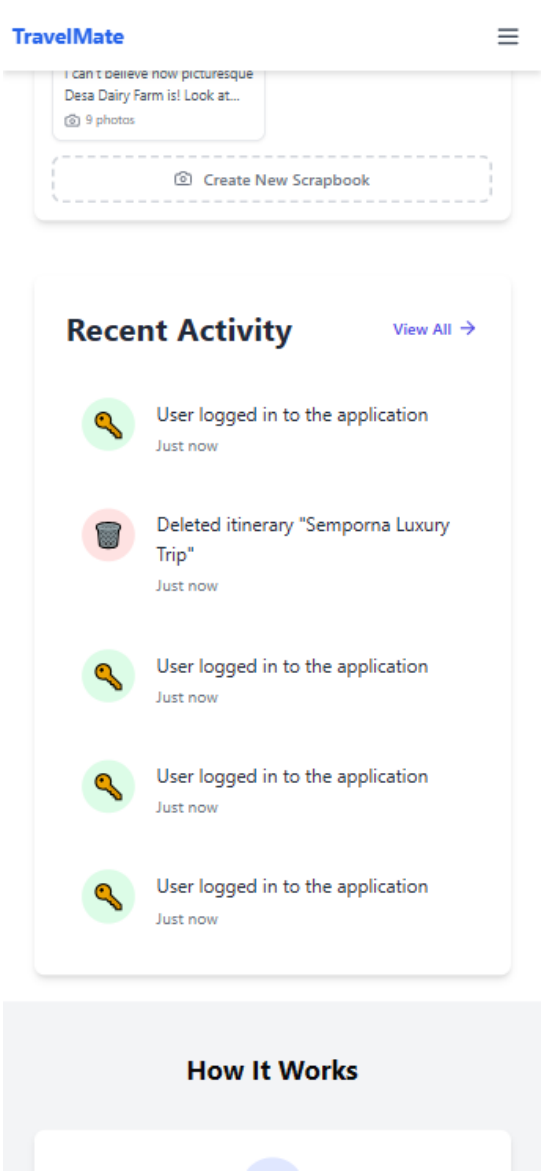

Figure 5.43: Home Page 3 (After user login)

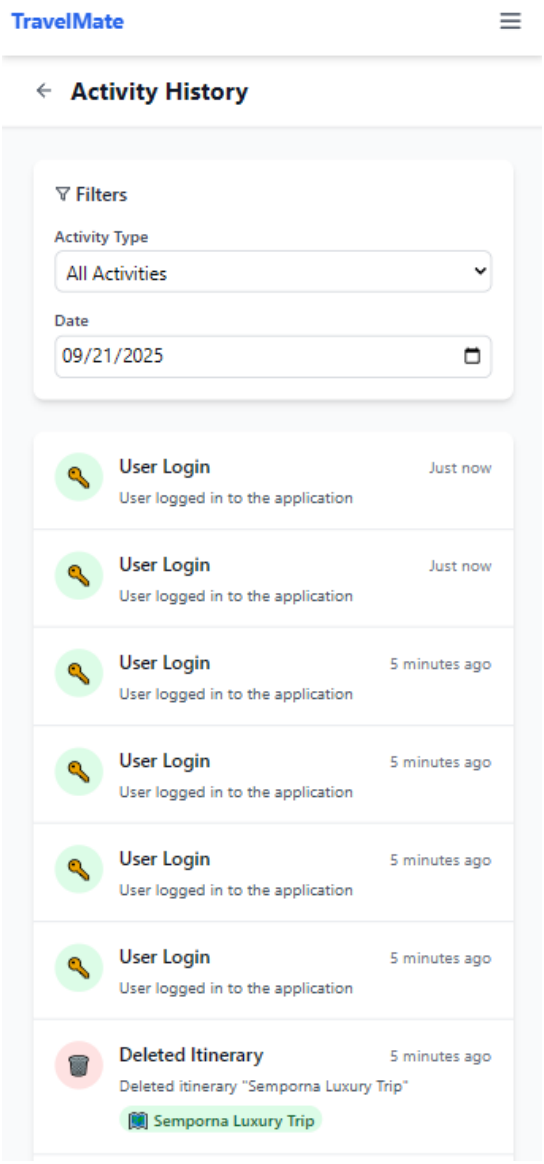

Figure 5.44: Activity History

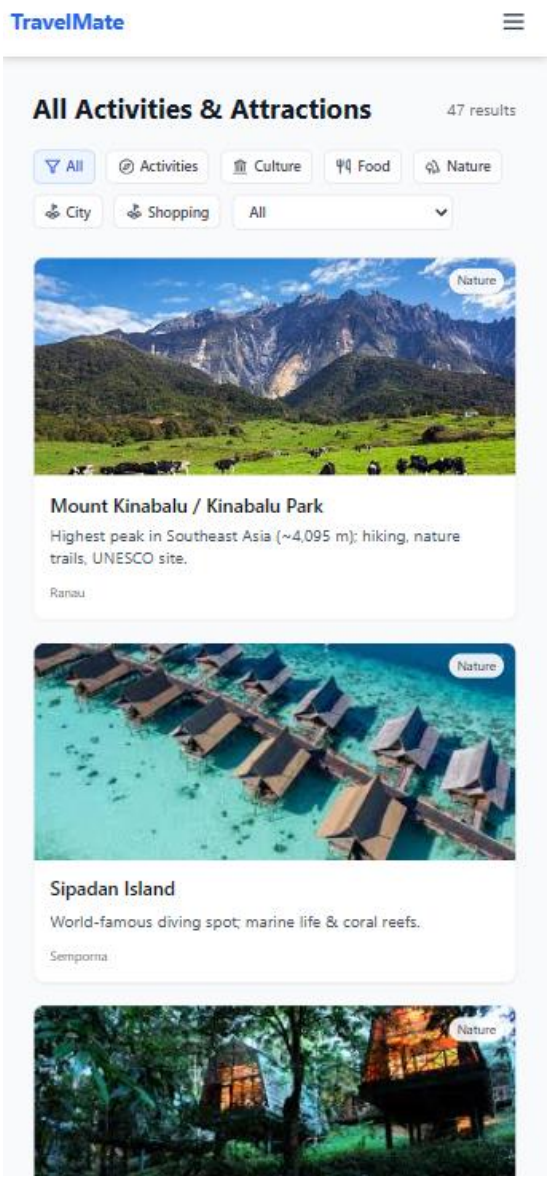

Figure 5.45: Activities Page

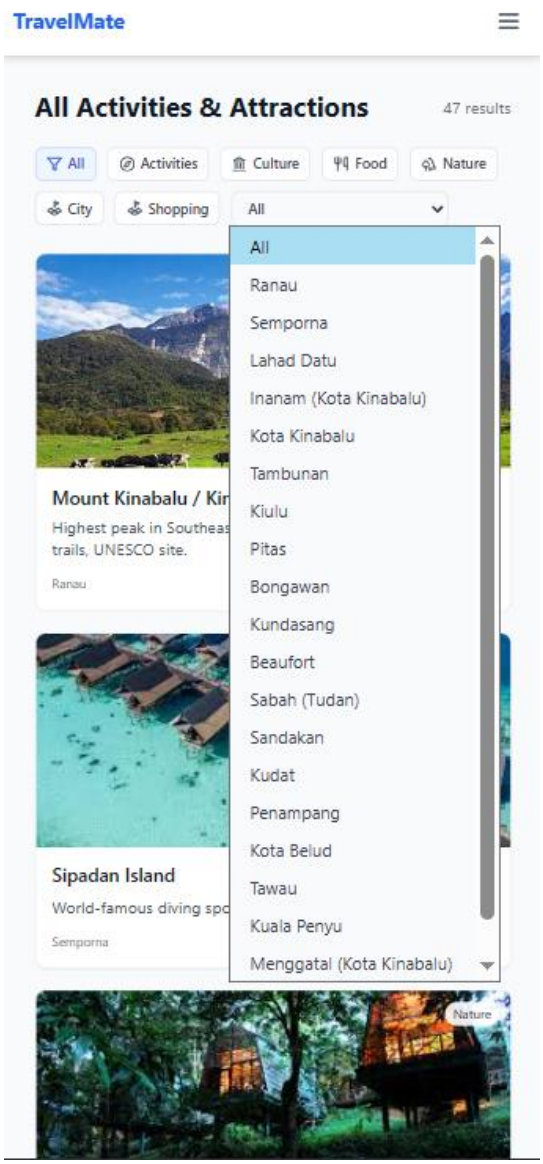

Figure 5.46: Activity Area Filter

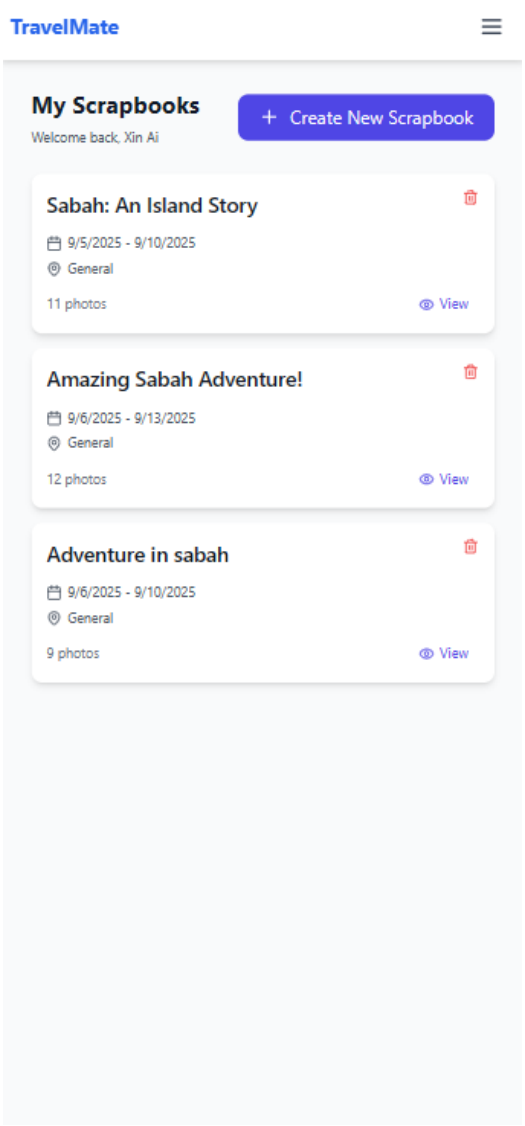

Figure 5.47: Scrapbook List Page

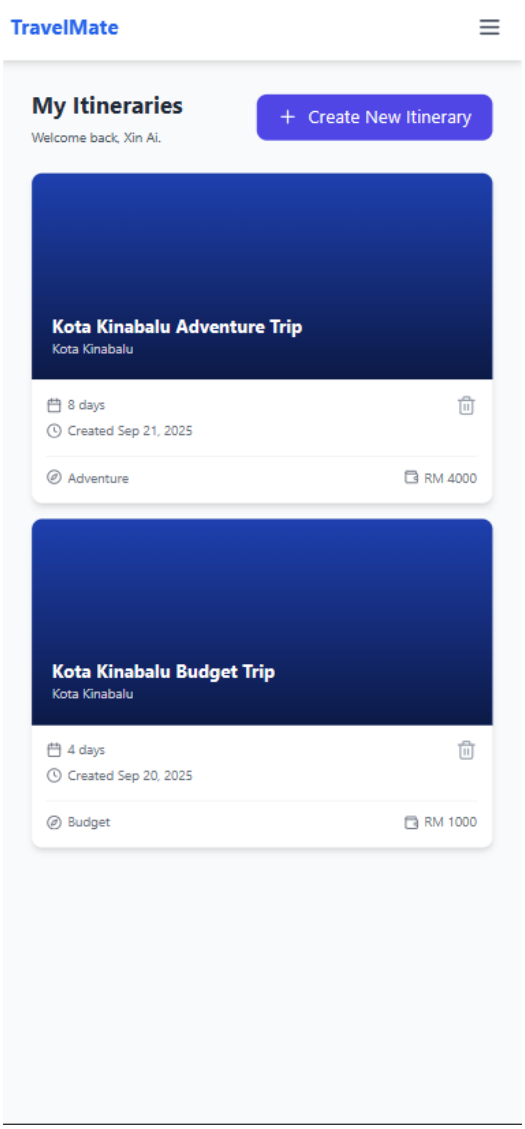

Figure 5.48: Itinerary List Page

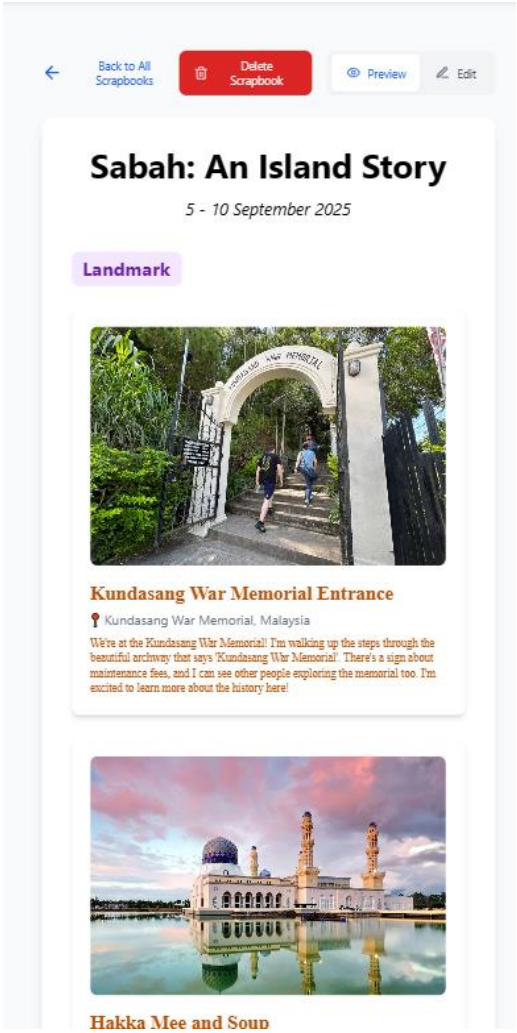

Figure 5.49: Scrapbook Preview Mode

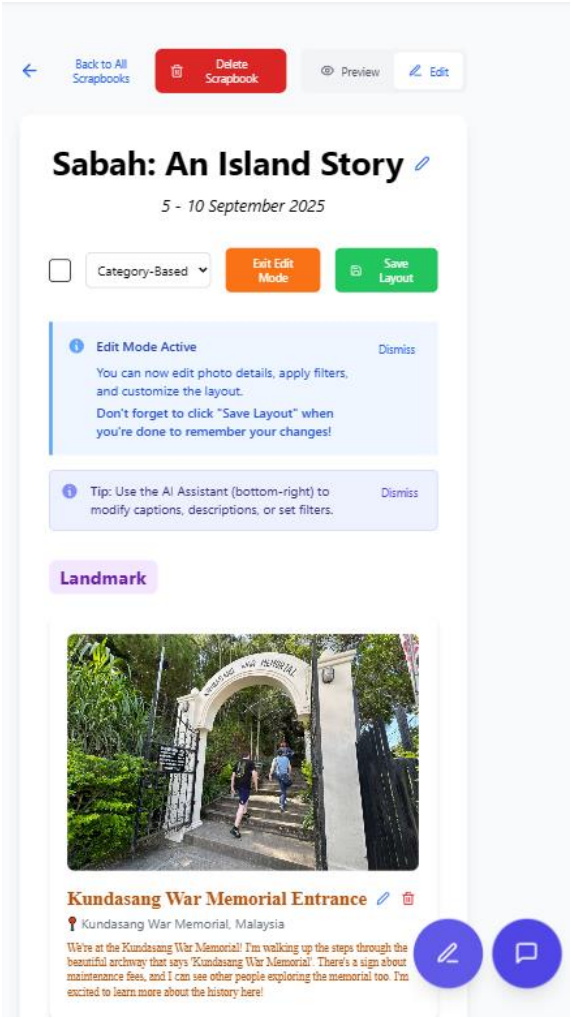

Figure 5.50 Scrapbook Edit Mode

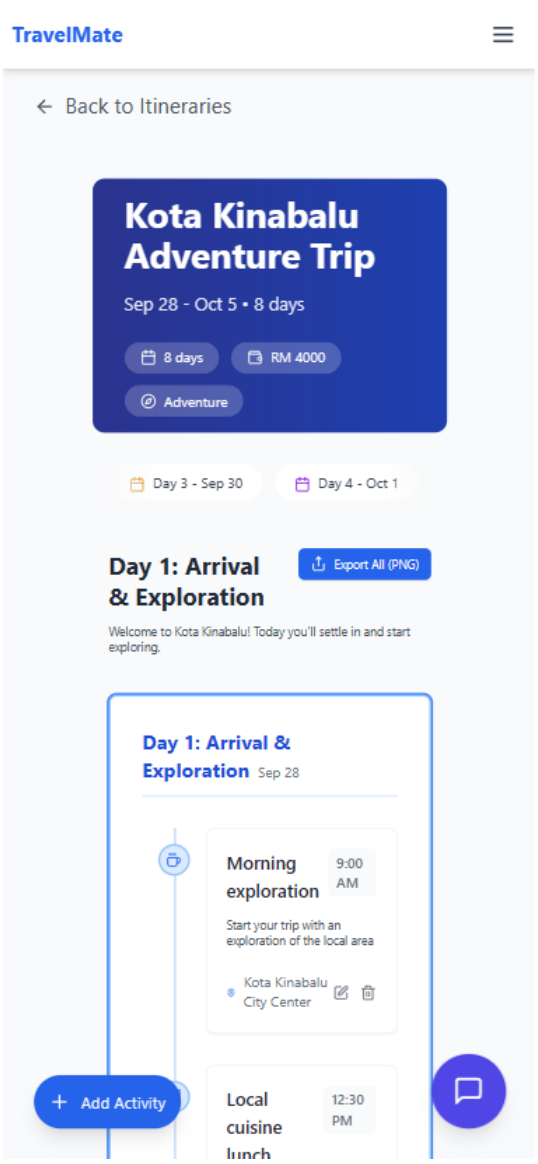

Figure 5.51: Itinerary Detail Page

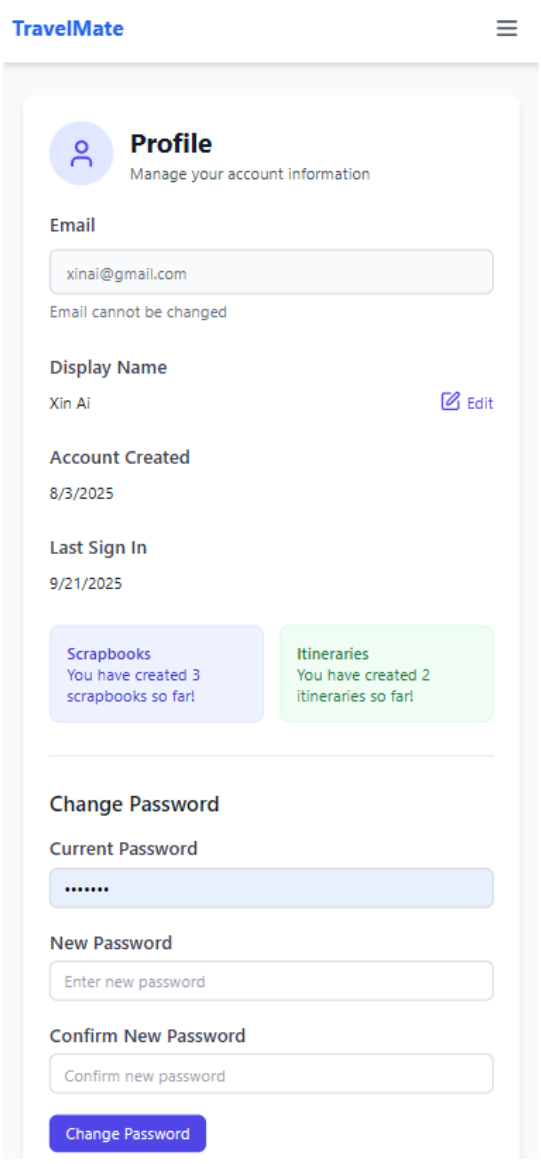

Figure 5.52: User Profile

5.5 Implementation of Issues and Challenges

During the development of the Sabah Smart Travel Companion, several technical and design challenges arose that required careful consideration and workarounds.

One of the most prominent challenges involved the integration of the Gemini API. While the API enabled features such as itinerary creation, image analysis, scrapbook captioning, and location extraction, its usage limits became a recurring issue. Heavy workloads particularly when analyzing multiple photos, and producing detailed itineraries, led to frequent rate-limit errors. This highlighted the need to optimize requests by minimizing redundant calls and prioritizing essential outputs.

Handling large batches of images for scrapbook generation was another significant challenge. Processing multiple photos simultaneously triggered API rate limits and slowed system responsiveness. To mitigate this, the system restricted scrapbook uploads to a maximum of 15 photos per session and enforced sequential processing. While this added a minor step for users, it maintained stability and ensured smoother performance during image analysis and caption generation.

Another notable limitation was the inability to fully integrate social media APIs, such as Instagram and Facebook, for direct publishing of scrapbook pages. While this feature would have streamlined the user experience by allowing one-click sharing, the technical and administrative requirements of these APIs posed a barrier. These platforms require review processes and verified business accounts for publishing permissions, which prevented direct integration in the prototype. As a workaround, scrapbook pages can be exported as PNG files, allowing users to manually share their travel memories online. While functional, this highlighted a gap in seamless social media integration that could be addressed in future iterations.

Overall, the challenges encountered highlight the complexity of balancing functionality, security, and scalability in the development of an AI-driven travel companion. While effective workarounds were applied to deliver a functional and user-friendly prototype, these issues provide a clear roadmap for future improvements, particularly in secure API management, optimized media handling, and seamless third-party integration.

5.6 Concluding Remark

The development of the Sabah Smart Travel Companion successfully integrated both hardware and software components to deliver a functional AI-driven travel application. Through careful configuration of tools, dependencies, and Firebase services, the system achieved a stable and secure environment for scrapbook creation, itinerary planning, and real-time data synchronization. Despite encountering challenges such as API usage limits, handling large image payloads, and social media integration constraints, effective workarounds ensured the prototype remained responsive and user-friendly. Overall, the project demonstrates the feasibility of combining advanced AI features with mobile-friendly travel applications, while also highlighting key areas for future improvements in scalability, API management, and seamless integration with external services.

CHAPTER 6
System Evaluation and Discussion

6.1 System Testing and Performance Metrics

System testing was conducted to validate the reliability, functionality, and efficiency of the scrapbook and itinerary system. Both verification (unit and integration testing) and validation (user acceptance and performance testing) approaches were applied to ensure each module behaves as expected.

The verification plan tested each module independently (unit testing), then integrated modules together to verify smooth data flow between scrapbook creation, itinerary generation, and Firebase storage (integration testing). Validation testing was conducted with simulated user scenarios, measuring responsiveness and correctness of results. Performance metrics were also collected, focusing on time-sensitive operations such as photo uploads, layout generation, and cross-platform accessibility.

Functionality	Target Performance	Observed Result	Status
User Login/Sign Up	< 3 seconds	Avg. 1–2 seconds	Passed
Password Reset (email OTP)	< 1 minute	30–40 seconds	Passed
Photo Upload and Analyze by Gemini API (max 15 photos)	< 1 minute	Avg. 45–55 seconds	Passed
Layout Generation	< 30 seconds	Avg. 18–25 seconds	Passed
Export Scrapbook (PNG)	< 15 seconds	Avg. 10–12 seconds	Passed
Itinerary Generation (Gemini API)	< 40 seconds	Avg. 25–30 seconds	Passed

Table 6.1: Key Performance Matrics

These results confirm the system meets its performance benchmarks, with scrapbook and itinerary generation remaining within acceptable timeframes even under maximum test loads.

6.2 Testing Setup and Result

Testing was performed across both mobile devices (Android smartphones) and laptops (web browsers) to validate cross-platform compatibility. The Firebase backend was connected in both environments, ensuring synchronization and persistence of user data.

6.2.1 Testing of Authentication

Test Case	Description	Expected Outcome	Tested Outcome	Evaluation
Register account	User enters display name, email, password, confirm password, and clicks register	New account is created, saved in database, and user is redirected to Home Page	New account is created, saved in database, and user is redirected to Home Page	Pass
Forgot password	User clicks "Forgot Password" and enters registered email	System sends reset email/OTP link within 1 min; user can reset password successfully	System sends reset email/OTP link within 1 min; user can reset password successfully	Pass
Sign in	User enters correct email & password	User authenticated and redirected to Home Page with access to full features	User authenticated and redirected to Home Page with access to full features	Pass
Sign out	User clicks "Sign Out" button	Session is terminated and user is redirected to Guest Home Page	Session is terminated and user is redirected to Guest Home Page	Pass

Invalid login	User enters wrong email/password	Error message "Invalid credentials" is displayed, access denied	Error message "Invalid credentials" is displayed, access denied	Pass

Table 6.2: Testing of Authentication

6.2.2 Testing of User Profile

Test Case	**Description**	**Expected Outcome**	**Tested Outcome**	**Evaluation**
Change display name	User updates display name in profile settings	New display name saved in database and updated across the system	New display name saved in database and updated across the system	Pass
Reset password	User changes password from profile page	New password is updated, and user must use it for next login	New password is updated, and user must use it for next login	Pass
View stats	User opens profile to check total scrapbooks and itineraries	System displays total count of scrapbooks & itineraries created	System displays total count of scrapbooks & itineraries created	Pass
Invalid input check	User leaves profile fields blank	Error message displayed, changes not saved	Error message displayed, changes not saved	Pass

Table 6.3: Testing of User Profile

6.2.3 Testing of Scrapbook Module

Test Case	Description	Expected Outcome	Tested Outcome	Evaluation
Create Scrapbook	User enters title, start & end date	Scrapbook is created and added to scrapbook list	Scrapbook is created and added to scrapbook list	Pass
Delete Scrapbook	User deletes an existing scrapbook	Scrapbook removed from list and database	Scrapbook removed from list and database	Pass
Preview Scrapbook	User clicks scrapbook to preview	Scrapbook preview page displayed	Scrapbook preview page displayed	Pass
Export Scrapbook to PNG	User exports scrapbook	Scrapbook downloaded as PNG	Scrapbook downloaded as PNG	Pass
Text Edit	User edits text (font, size, color, style)	Text updates instantly in scrapbook preview	Text updates instantly in scrapbook preview	Pass
Image Edit	User edits image (crop, zoom, rotate, filter)	Image updates instantly and preview updated	Image updates instantly and preview updated	Pass
Swap Image	User drags/swaps two images	Images successfully switch positions	Images successfully switch positions	Pass
Change layout	User selects Grid/FullWidth/Mixed/ Category-based	Scrapbook re-arranged into chosen layout	Scrapbook re-arranged into chosen layout	Pass

Change background	User selects plain or Sabah-themed background	Scrapbook background updated	Scrapbook background updated	Pass
AI Assistant to edit Scrapbook	User interacts with chatbot to auto-edit scrapbook	AI suggestions applied (e.g., auto captions, layout advice)	AI suggestions applied (e.g., auto captions, layout advice)	Pass
Quick Action to edit scrapbook	User uses undo, redo, reset, shuffle, or remove all captions	Corresponding quick action executed correctly	Corresponding quick action executed correctly	Pass
Validation check (dates)	User enters invalid travel dates (end date < start date)	Error message "End date must be after start date" displayed, scrapbook not saved	Error message "End date must be after start date" displayed, scrapbook not saved	Pass

Table 6.4: Testing of Scrapbook Module

6.2.4 Testing of Itinerary Module

Test Case	Description	Expected Outcome	Tested Outcome	Evaluation
Create Itinerary	User enters travel style, budget, duration, destination	New itinerary is generated and saved	New itinerary is generated and saved	Pass

Delete Itinerary	User deletes itinerary	Itinerary removed from list and database	Itinerary removed from list and database	Pass
View Itinerary	User previews itinerary	Itinerary details displayed	Itinerary details displayed	Pass
Add activity into itinerary	User adds new activity with day, time, venue, description	Activity saved and shown in itinerary list	Activity saved and shown in itinerary list	Pass
Edit activity	User updates existing activity details	Changes saved and reflected in itinerary	Changes saved and reflected in itinerary	Pass
Delete Activity	User removes an activity	Activity deleted from itinerary	Activity deleted from itinerary	Pass
AI Assistant to edit Itinerary	User interacts with chatbot to auto-suggest itinerary edits	AI updates applied (e.g., reordering activities, adding suggestions)	AI updates applied (e.g., reordering activities, adding suggestions)	Pass
Export Itinerary	User exports itinerary as PNG	PNG downloaded successfully	PNG downloaded successfully	Pass

Table 6.5: Testing of Itinerary Module

6.2.5 Testing of Sabah Activities (Activities & Attractions)

Test Case	Description	Expected Outcome	Tested Outcome	Evaluation
Filter by category type	User filters activities by category (e.g., adventure, culture)	Only activities matching category displayed	Only activities matching category displayed	Pass
Filter by Sabah region	User filters by location/region in Sabah	Only activities in selected region displayed	Only activities in selected region displayed	Pass
Combined filter	User filters by category + region	Results match both filters	Results match both filters	Pass

Table 6.6: Testing of Sabah Activities Module

6.2.6 Testing of Preference Weight Adjustment Model

Test Case	Description	Expected Outcome	Tested Outcome	Evaluation
Initial probability distribution	System initializes weights for layouts and fonts (Grid=3, FullWidth=2, Mixed=3, CategoryBased=2).	Probabilities correctly calculated as Grid=30%, FullWidth=20%, Mixed=30%, CategoryBased=20%.	When scrapbook generation is triggered multiple times, different layouts are produced (not always the same one), roughly aligning with expected probabilities.	Pass
Single update after save	User selects **Grid** layout and saves scrapbook.	Grid weight increases by +1 → new probabilities: Grid=36%,	Next scrapbook generations show Grid more frequently than	Pass

		FullWidth=18%, Mixed=27%, CategoryBased=18%.	before, appearing in several consecutive trials.	
Multiple updates across saves	User alternates between saving **Grid** and **Mixed**.	Weights adjust proportionally, probabilities reflect reinforcement (e.g., Grid=33%, Mixed=33%).	Generated layouts alternate more often between Grid and Mixed, while FullWidth and CategoryBased appear less frequently.	Pass
Skewed probability scenario	CategoryBased layout is boosted to hold ~70% probability.	CategoryBased layout should appear about 7 out of 10 times in repeated scrapbook generations.	In 10 scrapbook generations, CategoryBased appeared 6 times, while other layouts appeared 4 times combined — close to expected probability distribution.	Pass
Edge case – equal weights	All layouts start with equal weights (e.g., 5 each).	Each layout has equal chance (25%) of being selected.	Over repeated generations, layouts appear in a balanced distribution, with no single layout dominating.	Pass

Table 6.7: Testing of Preference Weight Adjustment Model

6.3 Project Challenges

Several challenges emerged during the development and evaluation stages of the project, requiring iterative problem-solving and careful design decisions.

The first major challenge was layout generation and edit flexibility within the scrapbook module. While the scrapbook engine successfully generated layouts dynamically using JSON rules, integrating user edits such as image repositioning, text modifications, and background switching without breaking layout consistency proved complex. To resolve this, a two-phase process was introduced: the system initially generated a static layout, after which users could enter an edit mode to manually adjust elements before saving the final scrapbook page.

Another significant challenge was AI consistency in photo analysis and itinerary generation. Since the system relied on the Gemini API to analyze images and produce structured JSON data, the outputs occasionally varied in formatting or caption accuracy. To mitigate this, a post-processing validation layer was developed to enforce standardized fields such as title, category, and location, ensuring uniformity and reliability across all AI-generated outputs.

Limitation of the Gemini API quota which restricted the number of available free requests during testing and development was another restriction. As the system depended heavily on the API for scrapbook layout generation and itinerary planning, this constraint necessitated careful optimization strategies, including caching frequently used responses, batching requests, and limiting redundant queries to remain within the allocated quota.

Despite these challenges, the application of targeted solutions and optimizations significantly enhanced the system's overall stability, reliability, and usability, ensuring that the core objectives of the project could still be achieved effectively.

6.4 Objectives Evaluation

The main objective of this project was to develop an end-to-end AI-powered travel companion that unifies itinerary planning with intelligent scrapbook generation. This objective was successfully achieved through the development of a web-based system that integrates AI-driven photo analysis, adaptive personalization, and seamless user interaction within a single platform. By centering the scrapbook module while providing itinerary support, the project ensured that travel documentation and planning complemented each other in a coherent workflow.

Each sub-objective was also addressed. First, the AI-powered scrapbook generation module allowed users to upload photos, apply captions, and transform them into structured scrapbook pages through content categorization and captioning. Second, personalization and preference learning were achieved through the Preference Weight Adjustment Model, which continuously refined layouts, fonts, and styles to align with user behavior over time. Third, a lightweight itinerary planner was implemented using the Gemini API, producing pre-itineraries tailored to user preferences such as budget, travel style, and duration. While secondary to the scrapbook, this feature provided structured guidance to enrich the overall experience. Finally, the project ensured a seamless and accessible user experience by adopting a responsive React-based frontend with Firebase authentication and local storage for handling image uploads and export. Collectively, these outcomes demonstrate that the system successfully met its goal of delivering a unified, AI-enhanced platform for both travel planning and documentation.

6.5 Concluding Remark

Overall, the project can be considered a success as it achieves its main goal of providing an interactive AI-powered travel planning system with an integrated scrapbook module. The system allows users to plan itineraries, generate digital scrapbooks, and store travel experiences in a seamless and visually appealing way.

Through the integration of AI (Gemini API), Firebase, and dynamic layout generation, the application not only supports itinerary creation but also enhances user creativity by enabling scrapbook personalization. The combination of cloud storage, export functionality, and quick edit features ensures that the system is practical and adaptable for real travel use cases.

Although the project faced challenges such as AI output consistency, layout handling, and storage limitations, appropriate solutions and design compromises were implemented to ensure system stability. The project also opens opportunities for future enhancements, such as advanced personalization using preference learning, collaborative scrapbook editing, and real-time itinerary updates.

In conclusion, the project demonstrates the application of AI, cloud services, and user-centered design principles in developing an innovative travel system. It provides both practical value for travellers and academic contribution in exploring AI-assisted multimedia content generation.

CHAPTER 7

Conclusion and Recommendation

7.1 Conclusion

The Sabah Smart Travel Companion project set out to address a key challenge in modern tourism: how to combine itinerary planning and memory preservation into a single, AI-driven application. Traditional travel apps often separate utility from storytelling, leaving users with one tool for planning and another for documenting their journey. This project bridges that gap by integrating an AI-based itinerary planner with a smart scrapbook generator, providing a seamless and engaging travel companion. Through the implementation of scrapbook generation, personalized itinerary creation, and cultural activity suggestions, the system not only supports functional travel logistics but also enhances the emotional and cultural dimensions of travel.

The core contribution is the scrapbook module, which balances structured automation with extensive personalization. Users can upload photos, generate scrapbooks, edit content, and experiment with layouts, fonts, and colors. Manual editing ensures flexibility, while the Preference Weight Adjustment Model, using a roulette-wheel style selection algorithm, adapts scrapbook layouts and font choices based on user interactions. Over the time, this probabilistic approach reinforces preferred options while still allowing less frequently chosen layouts to appear, producing scrapbook outputs that evolve according to each traveller's unique style. Exporting completed scrapbooks as high-quality images further supports long-term memory preservation.

The itinerary module complements the scrapbook by generating editable travel plans based on user preferences and Sabah's attractions, enhancing pre-trip planning without overshadowing the scrapbook's primary focus. The prototype demonstrated the feasibility of combining a modern web stack—Next.js, TypeScript, and Tailwind CSS—with Firebase and Gemini API integration, enabling dynamic photo handling, caption generation, and layout creation.

The project successfully met its objectives of enabling travel journaling, itinerary planning, and AI integration, validating the concept of an AI-powered smart travel companion. The scrapbook module delivered dynamic layouts with editable flexibility, while the Gemini API produced reliable photo analysis and itinerary outputs, reinforced by validation for consistency. Challenges such as API quota limits and large

data handling were addressed through caching, batching, and structured processing, enhancing efficiency without compromising usability. Overall, the prototype demonstrated robustness, practicality, and scalability potential within the defined scope.

Ultimately, this project highlights the potential of artificial intelligence in transforming the tourism experience. Beyond mere planning tools, AI can add cultural depth, improve memory preservation, and enhance user engagement in ways that traditional applications cannot. The Sabah Smart Travel Companion provides a foundation that can be further expanded into a production-ready platform, setting the stage for innovation in digital tourism not just for Sabah but as a model for other destinations.

7.2 Recommendation

While the current prototype achieved its objectives, several areas can be improved in future iterations. From a personalization perspective, future development should extend beyond the existing scrapbook editing tools and weight-based personalization model. Incorporating richer feedback mechanisms, such as explicit preference settings or adaptive style recommendations, would allow the system to refine scrapbook generation more accurately. Besides, extending export options such as PDF compilations, interactive HTML scrapbooks, or travel diaries in video form would also increase user flexibility and align with diverse digital preservation practices.

The application would further benefit from deeper localization. Integrating Sabah-specific cultural datasets (e.g., indigenous traditions, eco-tourism trails, and conservation initiatives) would enrich itineraries and scrapbook content, strengthening the system's identity as a Sabah-focused digital companion. Such enrichment could be achieved through collaborations with tourism boards, cultural institutions, and local businesses.

Finally, scalability and collaboration should guide future enhancements. Implementing advanced Firestore indexing, pagination, and support for multi-user environments would enable real-time collaborative scrapbook editing and group itinerary planning. This would extend the system from an individual tool to a community-oriented platform, encouraging shared cultural documentation.

In summary, the Sabah Smart Travel Companion provides a strong foundation for an integrated travel experience. By addressing personalization, scalability, export options, and localized enrichment, future iterations can evolve from a prototype into a comprehensive, innovative solution that redefines how travellers plan, document, and preserve their journeys.

REFERENCES

[1] T. Promsaka and N. Sakolnakorn, “Important Factors and Policies That Contributed to Tourism in Malaysia Between 1991 and 2018,” International Journal of Innovation, Creativity and Change. www.ijicc.net, vol. 14, p. 2020, [Online].

Available: https://www.ijicc.net/images/Vol_14/Iss_12/15210_Sakolnakorn_2020_E1_R.pdf. (accessed Sep. 3, 2024).

[2] MIDA. “Smart tourism: future of tourism in Malaysia.” MIDA.

Available:

https://www.mida.gov.my/smart-tourism-future-of-tourism-inmalaysia/#:~:text=

Tourism%20is%20the%20third%2Dlargest,cent%20to%20the %20total%20GDP

[3] D. Gursoy, Y. Li, and H. Song, “Chatgpt and the hospitality and Tourism Industry: An overview of current trends and Future Research Directions,” *Journal of Hospitality Marketing & Management*, vol. 32, no. 5, pp. 579–592, May 2023. doi:10.1080/19368623.2023.2211993

[4] Sunshine Silva. “5 Reasons why having a Travel Itinerary is Important”.

Available:

https://www.northsouthtravel.com/5-reasons-why-having-a-travel-itinerary-is

important (Accessed 1 April 2024)

[5] Zheng, Yuxiang, and Yue Wu. “An Investigation of How Perceived Smart Tourism Technologies Affect Tourists’ Well-Being in Marine Tourism.” *PLOS ONE*, vol. 18, no. 8, 25 Aug. 2023, pp. e0290539–e0290539.

[6] Torabi, Zabih-Allah, et al. “Smart Tourism Technologies, Revisit Intention, and Word-of-Mouth in Emerging and Smart Rural Destinations.” *Sustainability*, vol. 15, no. 14, 1 Jan. 2023, p. 10911.

[7] Chen, Aili, et al. “TravelAgent: An AI Assistant for Personalized Travel Planning.” *Ar5iv*, 2024, ar5iv.labs.arxiv.org/html/2409.08069.

[8] Yang, Shuran. “Storytelling and User Experience in the Cultural Metaverse.” *Heliyon*, vol. 9, no. 4, Apr. 2023, p. e14759.

[9] Dionisio, Mara, and Valentina Nisi. “Leveraging Transmedia Storytelling to Engage Tourists in the Understanding of the Destination’s Local Heritage.” *Multimedia Tools and Applications*, vol. 80, 7 June 2021.

[10] Zhu, Di, et al. “Digital Storytelling Intervention for Enhancing the Social Participation of People with Mild Cognitive Impairment: Co-Design and Usability Study.” *JMIR Aging*, vol. 7, 17 Jan. 2024, pp. e54138–e54138, https://doi.org/10.2196/54138.

[11] Wan, Bruce. *Exploring a Scrapbook That Promotes Wellbeing – Synergy between Oral and Visual Narratives of Memorable and Meaningful Experiences*. 1 Jan. 2019.

[12] IBM, “Content-based filtering,” Ibm.com, Mar. 21, 2024. Available: https://www.ibm.com/think/topics/content-based-filtering.

[13] Z. Zhang, Y. Liu, G. Xu, and H. Chen, “A weighted adaptation method on learning user preference profile,” Knowledge-Based Systems, vol. 112, pp. 114–126, Nov. 2016.

[14] “Useful algorithms to select random samples based on weight | Zyxware,” Zyxware Technologies, 2022. Available: https://www.zyxware.com/article/6505/useful-algorithms-to-select-random-samples-based-on-weight

[15] D. Gaspar-Figueiredo, “Learning from Interaction: User Interface Adaptation using Reinforcement Learning,” arXiv (Cornell University), Jan. 2023.

[16] Scrapbook “Shou Zhang”.

Available:

https://www.shouzhang.com/.

[17] Traveldiariesapp.

Available:

https://www.traveldiariesapp.com/en

APPENDIX

Poster