UNIVERSITY OF LIVERPOOL

Smart Travel App
Trip Cuts: A Social platform for travellers with influencer curated itineraries

Klinsmann Reddisson

201766073

A Dissertation submitted to

The University of Liverpool

in partially fulfilment of the requirements

for the degree of

MASTER OF SCIENCE

STUDENT DECLARATION

I hereby declare that the work presented in this dissertation is entirely my own and represented original research and findings. Where the ideas, language, or work of others have been used, they are fully acknowledged through proper citation and referenced in accordance with academic convictions. Any direct quotations are clearly identified and attributed to their respective sources.

I confirm that I have neither copied nor reproduced any material from other sources without due acknowledgment, nor have I engaged in plagiarism in any form. Furthermore, I affirm that I have not falsified, fabricated, or misrepresented data or results in this work. No part of this dissertation has been commissioned, improperly proofread, or produced by any third party.

I also certify that this dissertation, in whole or in part, has not been submitted previously to any other institution for the award of any degree or qualification. This work is an original contribution to the field and complies with the academic integrity policies and standards set forth by this institution.

Date: 22.11.2024

Signature:

Klinsmann Reddisson

ACKNOWLEDEMENTS

I would like to express my sincere gratitude to my dissertation supervisor, **Dr. Simona Capponi**, for her invaluable guidance, support, and encouragement throughout this project. Her insightful feedback and expertise have been instrumental in shaping my research and helping me navigate challenges along the way. I am deeply appreciative of her mentorship and the time she has dedicated to ensuring the success of this work.

I would also like to extend my heartfelt thanks to **Mr**. **Keith Dures**, whose assistance and practical advice have been immensely helpful during the course of this study. I would also like to extend my appreciation to my family, friends, and colleagues for their unwavering support and encouragement during this journey. Their belief in me has been a constant source of motivation. Finally, I am grateful to the academic staff and peers at University of Liverpool for their resources and collaborative environment, which have significantly contributed to the successful completion of this dissertation. I offer my sincerest thanks. This work is a reflection of your support, and I am truly grateful for each of your contributions.

Smart Travel App

Trip Cuts: A Social platform for travellers with influencer curated itineraries

Abstract

The need for exploration is a fundamental human desire. However, current travel planning tools rely on generic and static content, often lacking clarity and personalization. This creates a tedious and overwhelming process for travellers.

This project addresses these challenges by developing a social platform tailored for travellers which features influencer-curated itineraries that are showcased to users through engaging short videos, making them easily accessible and a cohesive space for travel inspirations.

Existing tools and research have focused on generic solutions or static itineraries, failing to create a dynamic platform that combines inspiration, adaptability, and user-specific needs. Even with AI-curated travel apps, true travel enthusiasts often find a gap, as these platforms don't fully connect with their specific desires and preferences. The platform leverages advanced algorithms and user preferences to offer dynamic, influencer-curated itineraries and personalized trip recommendations.

This results in a streamlined and intuitive travel planning experience. The platform adapts to individual preferences, simplifies complex processes, and offers an engaging approach to trip planning.

Contents

Introduction

In recent years, the travel industry has witnessed a significant shift towards digital platforms, with social media playing a pivotal role in shaping travel experiences and decisions. As, of 2024, a staggering 74% of travellers use social media during their journeys, highlighting the integral role of digital connectivity in modern tourism (Reali 2024).This trend underscores the need for innovative solutions that cater to the evolving preferences of tech-savvy travellers.

The intersection of social media, influencer marketing, and travel has created a new paradigm in the tourism sector. Social media platforms facilitate quick communication, allowing tourists to exchange and share travel information and rate and comment on the tourism experiences of others (Zhou Y 2022). Contemporary travel technology, despite significant advancements, continues to struggle with a fundamental challenge: capturing the intricate, emotional dimensions of travel experiences (Payne 2024).

To address this gap, a groundbreaking travel platform can shift the focus from traditional booking systems to an immersive, experience-centric approach to trip planning. By offering a personalized, community-driven framework, such a platform would connect travellers with content creators to share authentic, verified travel experiences. This would foster a thriving ecosystem for discovering genuine, well-curated itineraries. Tourists can obtain useful travel information in such an interactive environment to make rational travel decisions (Pudliner 2007).

A crucial factor in the success of mobile tourism applications is their ability to deliver a seamless user experience, driven by well-structured, intuitive, and engaging design (Garry Wei-Han Tan 2017).Effective app design ensures task completion is straightforward and encourages users to experience "flow"—a state of deep engagement and satisfaction while navigating the platform (Smith 2018). By prioritizing user-friendly and playful system architectures, such a platform could significantly enhance adoption rates and revolutionize the way people plan their travels (Kim 2020).

Problem Statement

The current travel planning industry faces several significant challenges that hinder travellers from enjoying a seamless and personalized journey. Travelers typically plan their trips through travel booking apps, offline agents, or social media. As mentioned earlier, travel booking apps primarily focus on bookings rather than the planning process. Offline agents, on the other hand, often provide generic itineraries that fail to meet the specific needs of most travellers, and in many cases, they rely on predefined data sets. Social media appears to be the most promising alternative, offering more personalized insights.

While social media plays a pivotal role in shaping travel decisions, it often lacks the structure needed for efficient, personalized planning, with recommendations typically focusing on the same popular destinations. Platforms like YouTube and Instagram may inspire with glimpses of places, but they fail to offer actionable and well-organized travel plans.

A primary challenge lies in the plethora of travel-related information being created and disseminated across numerous social networking platforms (Jinyi Xu 2023).This makes it difficult to sift through and find relevant, personalized travel inspiration. Another major limitation is the scarcity of authentic user content. While platforms like Instagram and YouTube provide travel-related videos, much of this content tends to be overly polished or promotional, reducing its reliability for genuine traveller insights. However, travellers are increasingly favouring videos when seeking product information online, particularly those that highlight cognitive and emotional value, which are challenging to assess for tourism services (Huertas A 2017).

Additionally, planning a comprehensive travel itinerary remains a time-consuming task, requiring users to navigate various websites and apps to gather information on destinations, activities, and logistics. This process becomes even more cumbersome when seeking unique, off-the-beaten-path experiences. Furthermore, many travel apps overemphasize popular tourist spots, leaving lesser-known, local destinations underrepresented and limiting travellers' exposure to diverse, authentic options.

Motivation

The motivation for developing this travel app is rooted in the growing demand for personalized travel experiences and the significant role social media plays in influencing travel decisions. Social media wields a tremendous influence on tourism today. A remarkable 78% of Americans report being inspired by social media influencers to discover new destinations, restaurants, or attractions. Globally, 35% of consumers—and an even higher 53% among Gen Z—turn to social media for travel inspiration (Woolf 2024). These figures underscore the powerful role social platforms play in shaping travel decisions and preferences.

However, there are concerns regarding the authenticity and trustworthiness of influencer-driven content. Nearly 59% of users have felt misled by a travel influencer at least once, with accuracy, financial misrepresentation, and lack of authenticity being the primary concerns (Woolf 2024). This presents a unique opportunity for the app to offer a more transparent and trustworthy alternative.

Additionally, with growing concerns around sustainable tourism, the app could guide users toward responsible travel practices and less-explored destinations, positively impacting local economies and reducing the environmental footprint of travel.

Approach

In modern times, mobile-based short video apps have emerged as a rapidly growing branch of social media, expanding at an unprecedented pace. These apps represent a transformative medium tailored to align with modern content consumption habits in today's fragmented information landscape (Lu 2019). Unlike traditional social media platforms, short video apps offer distinct technical advantages and a unique user experience. For instance, travel vloggers leverage customizable editing tools and filming templates to craft engaging short videos showcasing diverse travel experiences, such as exhibitions, festivals, and more (Du 2020).

In contrast to static travel guides, this app will deliver users a continuously refreshed stream of authentic content, capturing real-time travel trends and genuine experiences, such as:

1. *User-Curated Travel Planning:* By leveraging user-generated content, the app fosters a dynamic platform where travel planning continually adapts and grows, providing fresh and personalized insights that keep pace with the ever-changing nature of travel.
2. *Verified and Trustworthy Content:* A content verification system will ensure the reliability of shared travel information, confirming the content's connection to real places and events. This oversight creates a trustworthy platform where users can confidently rely on content for trip planning.
3. *Content Sharing and Community Building:* Travelers can create and share their itineraries, videos, and blogs, fostering a vibrant community. This interactive environment encourages users to connect over shared experiences, while the Instagram-like format keeps the platform engaging and easy to navigate. Fresh perspectives and recommendations will continually update the platform.

4. *Safety and Navigation Support:* The platform will include features like safety tips, route tracking, and guidance for solo travellers, providing peace of mind during both the planning and travel stages.

By combining short-form video content with a structured travel plan and community support, this platform will transform how travellers find inspiration and connect over shared adventures, ensuring each journey feels safe, personalized, and well-informed.

Scope

The app targets two main audiences: creators who wish to share their travel stories and inspire others, and users looking for immersive travel ideas to plan their trips. Content curators are expected to possess deep knowledge and expertise, enabling them to create high-quality content that is both functional and highly usable, effectively meeting the needs of their audience (Camilleri 2023).

As the user base grows, the platform will emphasize lesser-known or underexplored destinations, encouraging more local creators to provide insights about their regions. This focus on unique, local content aims to diversify travel inspiration and foster a community with varied perspectives. The functionality of a technology significantly influences customers' perceptions of its usefulness, ultimately shaping their intentions and willingness to adopt and use it (Yu 2017).

Outcome

The app's unique selling point (USP) will be its user-curated itineraries, providing users with a one-stop shop for all their trip details, eliminating the need to juggle between multiple apps. With an Instagram-inspired user interface, the app will be intuitive and easy to navigate, allowing users to quickly adapt without a steep learning curve. The key feature, "Cuts" (short 30-second clips), will enhance user engagement by offering an immersive, quick-access format for exploration. Psychologically proven to boost user engagement, the short-form video format triggers dopamine responses, encouraging users to keep exploring and interacting with the app (Sharif 2022). (Sharif 2022). Travelers share this information across multiple platforms, creating an environment that helps fellow travellers make informed decisions (Y 2014). This increased engagement will enrich the user experience and could reduce dependency on traditional travel agents by providing reliable, user-generated insights. While the app is not intended to fully replace travel agents, it will serve as an accessible alternative, with the potential to empower users to make informed travel decisions directly within the app. Overall, the app aims to bridge the gap between the social media influence on travel and the need for authenticity and personalization, offering users a trustworthy and tailored travel planning tool.

Literature Review

Role of Mobile App Quality in Enhancing User Engagement and Loyalty

The quality of mobile applications plays a vital role in shaping user experiences and fostering engagement. Studies emphasize that attributes such as system reliability, efficient performance, and accurate information delivery significantly contribute to user satisfaction and loyalty (Meltem Caber 2024) (Tahir Albayrak 2023). High-quality applications ensure seamless interactions, reducing user frustration and enabling uninterrupted flow experiences. These qualities not only encourage users to remain active on the platform but also motivate them to become advocates for the service by sharing positive feedback and promoting it within their networks.

These insights underscore the necessity of developing intuitive, user-friendly interfaces that cater to user needs while delivering personalized experiences. Emotional engagement is a key driver of customer loyalty, and apps that successfully evoke a connection with their users are more likely to cultivate long-term relationships. Additionally, reliability and precision in information foster trust, a crucial element for platforms aiming to establish themselves as dependable tools for travel planning. By prioritizing these aspects, platforms can achieve their goals of building collaborative communities and ensuring sustained user participation.

Advanced Features and Personalization

The integration of advanced features into travel applications can revolutionize how users plan and experience their journeys. Studies stress the importance of incorporating elements such as geolocation tracking, real-time personalization, and augmented reality (AR) to cater to modern traveller's demands (Lius Steven Sanjaya 2017), (Phoebe Yueng-Hee Sia 2023). These features provide convenience, enhance interactivity, and make travel planning more engaging. For instance, AR can offer immersive previews of attractions, while geolocation ensures accurate navigation and location-based recommendations.

However, these innovations come with their own set of challenges, such as high development costs, privacy concerns, and technical limitations in location-based services. Addressing these issues requires a careful balance between innovation and practicality. Personalized recommendations, for example, can significantly improve user satisfaction by offering tailored itineraries, localized suggestions, and interactive trip details. By focusing on these advanced yet user-friendly features, platforms can differentiate themselves and align closely with the evolving expectations of their users.

Usability and Functionality as Key Drivers of User Satisfaction

The usability and functionality of an application are essential for driving user engagement and satisfaction. Studies highlight that users are drawn to applications that offer a seamless experience, from intuitive navigation to reliable, high-quality content (Camilleri 2023) (Dymkov 2021). These factors are not just critical for initial adoption but are also instrumental in promoting frequent usage and behavioural loyalty.

For travel platforms, aligning design and functionality with user expectations is imperative. Features such as streamlined navigation, easy-to-use interfaces, and visually appealing layouts can enhance user experiences significantly. In addition, tools like 3D mapping and adaptive route planning not only improve usability but also address practical concerns such as safety, route efficiency, and environmental sustainability. By embedding these elements into their platforms, travel applications can create a trustworthy, efficient, and enjoyable environment for users.

Impact of Short Video Platforms on Travel Planning

Short video platforms have dramatically influenced how users seek travel information and make decisions. These platforms, powered by features like personalized recommendations, search functionalities, and live streaming, have redefined content consumption in the travel sector (Jinyi Xu 2023). Visual storytelling through short videos provides potential travellers with immersive, engaging, and credible insights into destinations.

Emotionally driven videos stand out as particularly impactful in inspiring travel intentions. Studies show that videos rich in storytelling and emotional resonance are far more effective in encouraging users to act on their aspirations than those solely focused on information delivery (X 2023). This process of inspiration, moving users from being "inspired by" to "inspired to," underscores the transformative potential of high-quality video content. Platforms that prioritize dynamic and interactive media can create an environment that fosters deep engagement, enabling users to visualize their trips and make informed decisions (Zhou Y 2022) (Liu J 2023).

User-Generated Content and Social Influence

User-generated content has emerged as a powerful tool in travel planning, particularly during the pre-trip phase. Videos created by users offer vivid, authentic representations of destinations, helping potential traveller's shape their expectations and make decisions (Phuong Minh Binh Nguyen 2024). The credibility of such content often surpasses that of professionally curated materials, as it provides real-world insights and experiences.

Moreover, the influence of online word of mouth is amplified when combined with engaging User-generated content. Positive reviews and interactions within online communities can significantly enhance the perceived value of a travel platform, building trust and encouraging participation. Platforms that integrate user-generated videos and foster dynamic interactions can position themselves as hubs for authentic and collaborative travel planning (Zhou Y 2022).

Addressing Real-World Travel Planning Complexities

Travel planning is inherently complex, involving a multitude of constraints such as budgets, itineraries, and personal preferences. While advanced tools like AI and natural language models show potential in addressing these challenges, they often fall short in managing the intricacies of real-world scenarios (Jian Xie 2024). Developing intuitive platforms that can accommodate diverse user needs while offering precise and personalized solutions is essential.

Such platforms should incorporate tools that simplify decision-making, such as automated budget calculators, adaptive itinerary planners, and preference-based recommendations. By addressing these practical challenges, travel applications can streamline the planning process, reduce user effort, and enhance satisfaction. This aligns with the broader goal of creating efficient, user-centric solutions for modern travellers.

Innovative Navigation and Sustainable Travel Practices

Navigation is a fundamental aspect of travel planning, and its effectiveness can greatly influence user experience. Studies highlight the importance of route consistency and geographical accuracy in building user trust and ensuring safety (Dymkov 2021). Features such as 3D mapping, real-time tracking, and adaptive planning not only improve navigation but also contribute to sustainability by optimizing routes and reducing environmental impact.

Travel platforms that leverage innovative navigation tools and incorporate community-driven content can significantly enhance usability. By prioritizing accurate, intuitive, and eco-friendly navigation solutions, these platforms can cater to the needs of environmentally conscious traveller's while fostering trust and reliability.

Industry Sources

No.	Author	Title	Research Methodology	Key Findings	Industry Insights
01	(Meltem Caber 2024)	Building customer citizenship behaviour through mobile application quality: the mediating role of flow experience and customer engagement.	A survey-based quantitative analysis using structured questionnaires.	App quality improves user engagement and advocacy.	Focus on reliable, efficient apps to boost satisfaction and loyalty.
02	(Camilleri 2023)	Functionality and usability features of ubiquitous mobile technologies: The acceptance of interactive travel apps.	Analysis of mobile travel app features and engagement metrics.	Usability and functionality drive satisfaction and loyalty.	Create seamless, intuitive designs to enhance trust and frequent use.
03	(Lius Steven Sanjaya 2017)	Mobile Application Business Plan to Assist Travel Planning.	Mixed-method approach using surveys and statistical analysis on app quality.	Personalization enhances user satisfaction.	Offer personalized trip plans and localized content for better engagement.
04	(Phoebe Yueng-Hee Sia 2023)	Systematic review of mobile travel apps and their smart features and challenges.	Analysis of travel apps for multi-day route planning and tracking.	Advanced features improve experiences but have challenges.	Leverage tech like AR while addressing privacy and cost concerns.
05	(Jinyi Xu 2023)	Exploring factors influencing travel information-seeking intention on short video platforms.	Quantitative analysis of user-generated videos and travel intentions.	Short videos redefine travel planning.	Use dynamic content to inspire and engage users.
06	(Tahir Albayrak 2023)	The Use of Mobile Applications for Travel Booking: Impacts of Application	Survey-based analysis of app quality, user perceptions, and behaviours.	Trust and system reliability drive app adoption.	Build trustworthy platforms with accurate information and

		Quality and Brand Trust.			dependable service.
07	(Phuong Minh Binh Nguyen 2024)	A systematic literature review on travel planning through user-generated video.	Analysis of user-generated videos' impact on travel intentions.	User videos inspire and aid trip planning.	Promote community-driven, authentic content to build credibility.
08	(Jian Xie 2024)	TravelPlanner: A Benchmark for Real-World Planning with Language Agents	Evaluation of travel apps for route planning and navigation features.	AI tools face challenges in complex planning.	Develop intuitive systems for diverse travel needs.
09	(Zhou Y 2022)	The effects of perception of video image and online word of mouth on tourist's travel intentions: Based on the behaviours of short video platform users.	Quantitative analysis through structured user surveys.	Reviews and videos influence travel intentions.	Use engaging videos and positive reviews to shape user decisions.
10	(X 2023)	Which type of tourism short video content inspires potential tourists to travel.	Review of travel apps for route planning and user feedback.	Emotionally resonant videos drive travel decisions.	Focus on storytelling to inspire user actions.
11	(Liu J 2023)	How do short videos influence user's tourism intention? A study of key factors	Analysis of tourism videos using cognitive and word-of-mouth metrics.	Interactive videos enhance immersion and engagement.	Invest in high-quality, interactive content.
12	(Dymkov 2021)	Travel route planning and tracking apps.	Case study review of travel route planning and tracking apps.	Accurate routes and navigation improve safety and trust.	Use advanced navigation tools to enhance usability and promote sustainable travel.

Table 1 Industry Sources

Design

To ensure a better understanding of the applications flow and functionality, the design and implementation sections have been separated. This approach helps convey the technology and the underlying concepts behind each aspect of the app in a structured way, avoiding confusion with the flow of the application

Authentication

Splash Screen

Upon launching the application, users are greeted by the splash screen (Figure 1). This initial screen serves as a visually impactful introduction, establishing the app's branding by prominently displaying the logo. It features a smooth transition into the platform, creating an engaging user experience. The purpose of the splash screen is to provide a visually appealing entry point while minimizing load time, ensuring a seamless shift into the main interface.

Pre-Login Welcome Screens

For users who have not yet logged into the platform, a series of welcome screens are presented (Figure 4, Figure 3). These screens convey a brief introduction to the app, setting the tone for the user's journey. The final screen of this sequence prompts users to either register or log in (Figure 2), offering a clear and direct call-to-action that navigates them to the appropriate entry points for account creation or access.

Login and Registration Process

When users are ready to access the platform, they are prompted to log in via the login page (Figure 5), where they can enter their credentials—email and password to proceed. If users have forgotten their password, a conveniently placed "Forgot Password?" link directs them to a page where they can request a password reset. For users without an account, the registration page offers a simple and intuitive setup process, requiring only basic details such as name, email address, and a secure password (Figure 6). In the case of a forgotten password, users can easily navigate to the reset page by selecting the "Forgot Password?" link. After entering their email address, they will receive a secure link via email to reset their password. After either logging in or signing up the user would be redirected to the Main Page of the application (Figure 7).

User Categories After Authentication

While role-based access has been implemented on the backend, its integration into the front-end user interface is not yet complete. Once incorporated, this feature will differentiate user roles, such as administrators and regular users, and provide access to specific areas of the app based on their privileges.

For the current design, the user journey is conceptualized into two distinct paths for better understanding and clarity. These paths are not role-based or restricted; they are accessible to all users. This distinction serves only as a framework to explain the two primary ways users can interact with the platform:

1. *Travelers:* Users who focus on exploring and planning trips by engaging with itineraries, cuts, and travel content. They draw inspiration from the platform to organize their journeys.
2. *Creators:* Users who actively contribute to the platform by uploading cuts, sharing curated itineraries, and inspiring fellow travellers through their personalized content.

This approach highlights the platform's dual functionality, where any user can switch between exploring travel content and contributing to the community, fostering a versatile and engaging experience.

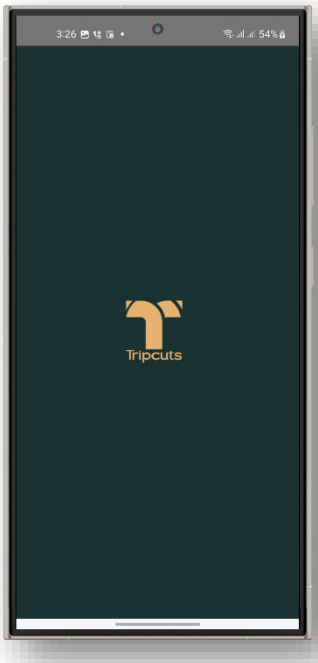

Figure 1 : Splash Screen

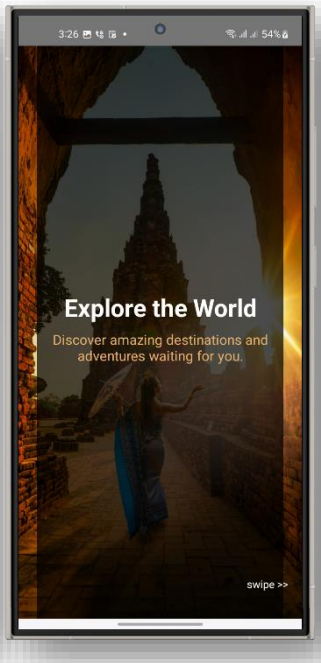

Figure 4 : Welcome Screen 1

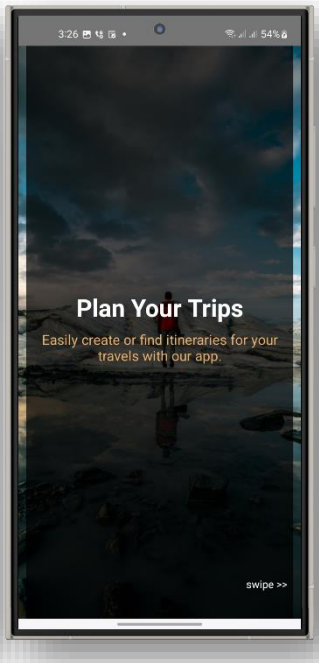

Figure 3 : Welcome Screen 2

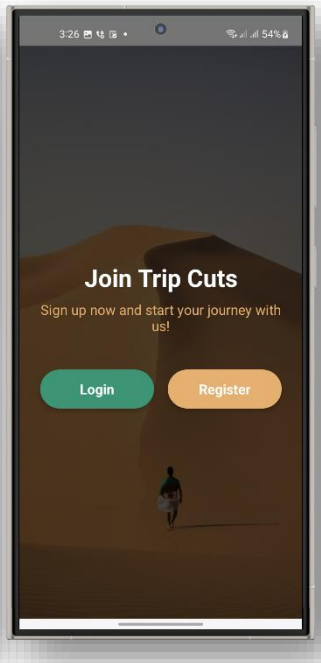

Figure 2 : Authentication

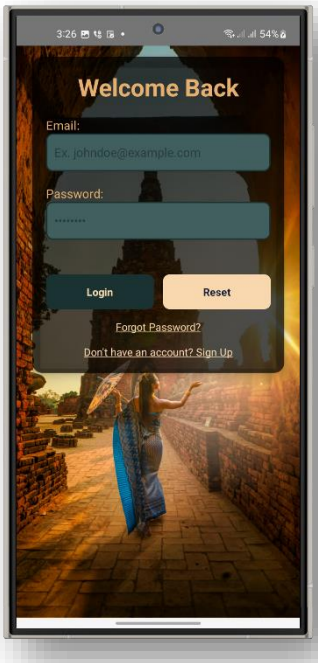

Figure 5 : Login

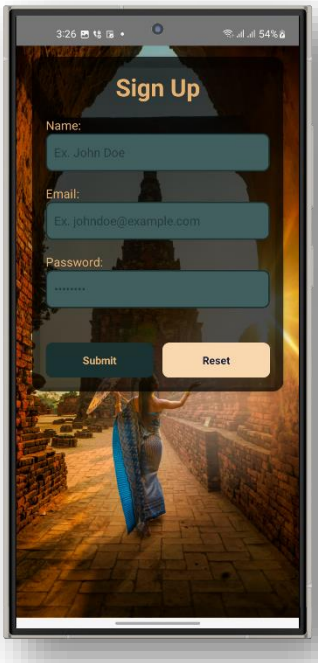

Figure 6 : Sign Up

Figure 7 : Main Screen

Main Screen: Traveller's approach to the application

Cuts Screen

This Screen features "Cuts" (Figure 8). *Cuts* is a stream of 30-second videos, each recommended based on the user's interests, which are gathered and analysed by the platform's algorithm. The videos are designed to switch automatically when the user performs an upward swipe gesture, offering a dynamic, engaging viewing experience. The interface is deliberately minimalistic, with no buttons or distractions. The only interactive element is the video itself. A double-tap on the screen will indicate a "like" action, while tapping once pauses the video, revealing a search bar, map button, and profile button for further user engagement (Figure 9).

Figure 8 : Cuts Playing

Figure 9 : Cuts Paused

Search Functionality

When the search bar is tapped, the page transforms into a search interface (Figure 10). In this interface, users can search for either a specific creator's name or a destination to explore related itineraries and cuts (Figure 11, Figure 12).

Upon searching for a creator, selecting their profile directs the user to their personal profile. On this profile screen, all verified cuts are displayed (Figure 13) and when user taps on a specific cut, the corresponding video opens (Figure 14). Additionally, if the itinerary button is pressed in the following video, the user is taken to a detailed trip guide (Figure 15, Figure 16, Figure 17, Figure 18), which is one of the important features of the application. Wherever there is *Cuts,* just by swiping right anywhere on the screen would do the same job of pressing the itinerary button. This guide includes important travel details such as:

- Vlog of the Travel
- Trip Description

- Daily Plans
- Safety Measures
- Estimated Budget
- Trip Type
- Best Time to Visit
- Language Tips
- Transport Options
- Reviews

Each itinerary is supplemented with multiple media for every location, providing a rich, multimedia experience that supports travel planning.

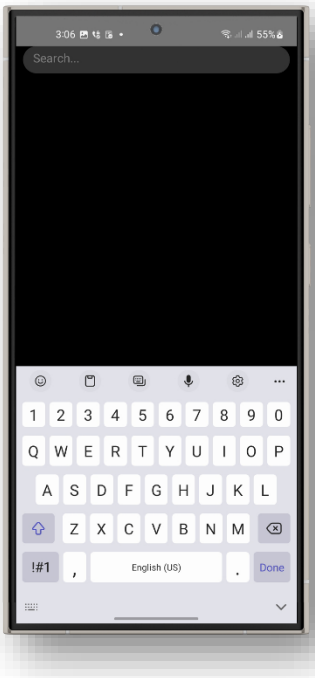

Figure 10 : Search Screen

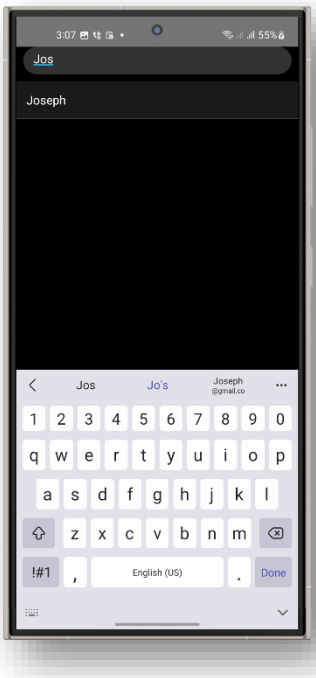

Figure 11 : Profile Search

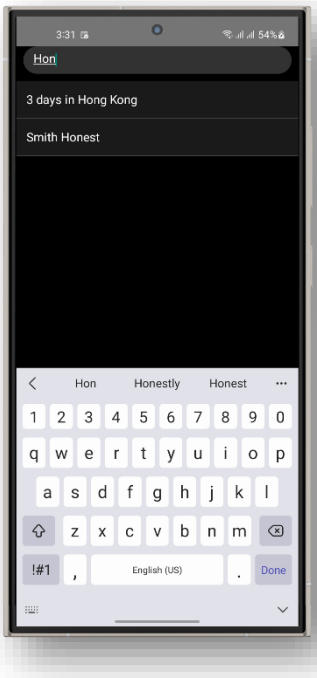

Figure 12 : Trip Search

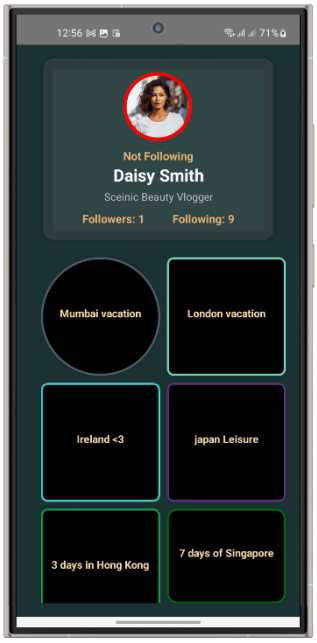

Figure 13 : Creator's Profile

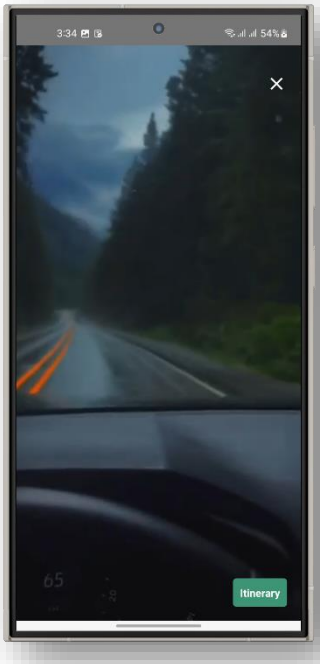

Figure 14 : Cuts

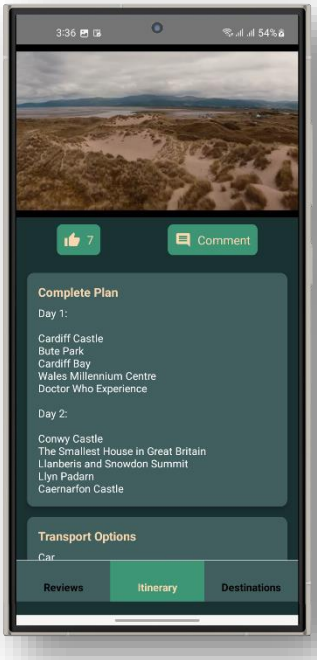

Figure 15 : Complete Itinerary

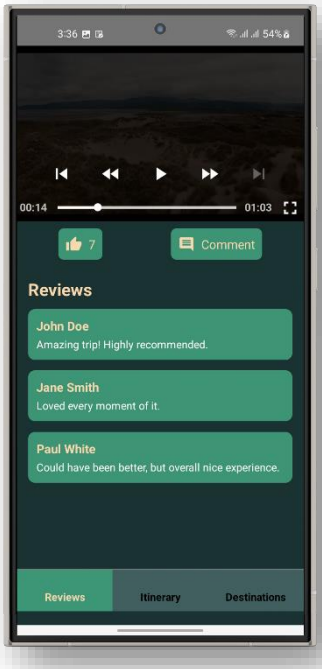

Figure 16 : Reviews

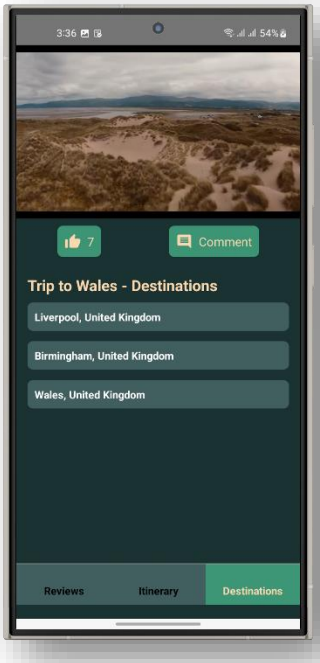

Figure 17 : Destinations

Figure 18 : Media

Map Functionality and Navigation

Returning to the main screen (Figure 10), from here the map button is available for navigation. Upon tapping this button, the app prompts users to enter both a source and destination (Figure 19). Once entered, the app provides turn-by-turn guided navigation, as shown in (Figure 20). The navigation interface includes voice-guided directions, a re-centre option, and relevant trip details, such as the total estimated time of arrival (ETA) and the total distance to be covered. This feature ensures a smooth and practical experience for users traveling to their desired destination.

Profile Page

The final feature on the main screen is the profile button, which takes users to their own profile page (Figure 21). This page mirrors the layout of the other creator profiles but with subtle differences tailored to the current user's personal information. The profile page allows users to manage their content, followers and other preferences, offering a unique and personalized experience that evolves based on their activity within the platform.

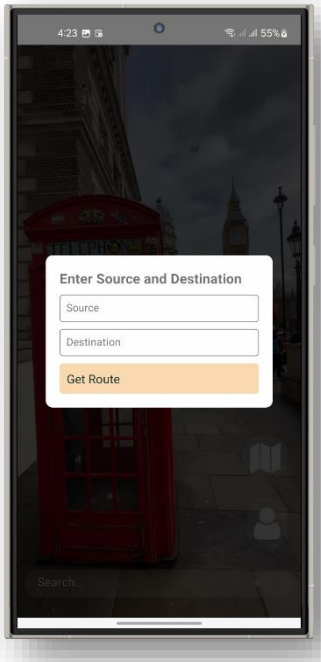

Figure 19 : Navigation Details

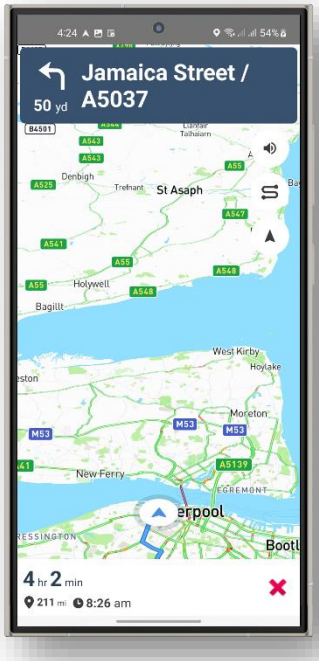

Figure 20 : Map

Create Itinerary: Creator's approach to the application
Initiating Trip Creation: Phase 1

This is a point where the creator starts creating the itinerary before starting the trip. From the creator's profile page, as previously described, verified trips are prominently displayed as cuts. To maintain authenticity, travellers can only upload trip details for public visibility if they have genuinely visited the destinations.

The trip creation process begins when the creator presses the plus icon labelled "Create Trip" (Yellow button on the top right corner in Figure 21). This action leads to the trip creation page (Figure 22). At this stage, creators must fill out an initial form before beginning their trip. The form includes fields such as Trip Name, Type of Trip, Expected Budget, Source, and Destinations. The application assists by suggesting place names as users' type.

The most critical fields in this form are the Source and Destinations (Figure 23, Figure 24), as these will later be verified to ensure the user has actually visited them. If the information entered cannot be validated, the creator will not be allowed to proceed with the trip creation process. After successfully completing the form and pressing "Create Trip," users are redirected to the Trips Created page (Figure 25).

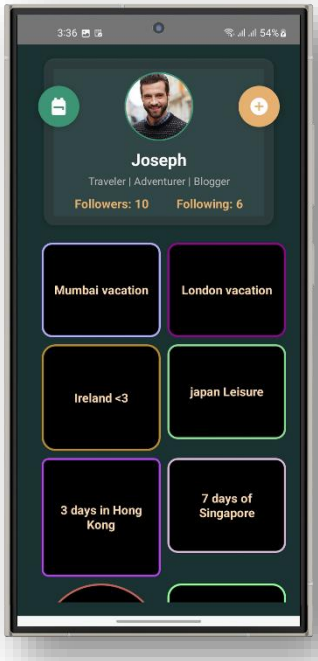

Figure 21 : User Profile

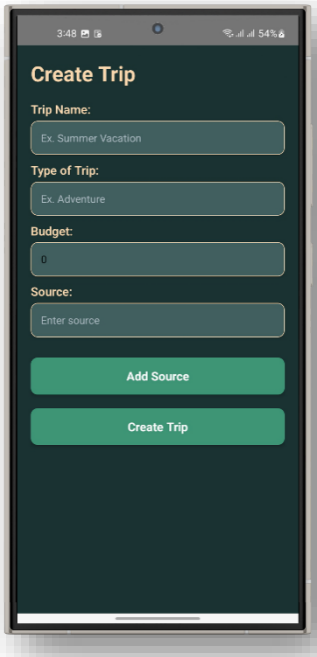

Figure 22 : Initial Destination Plan

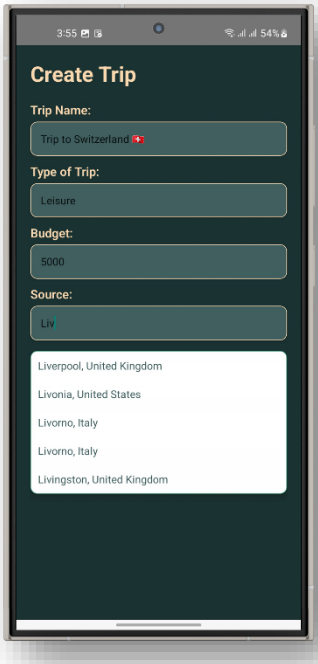

Figure 23 : Auto-Fill 1

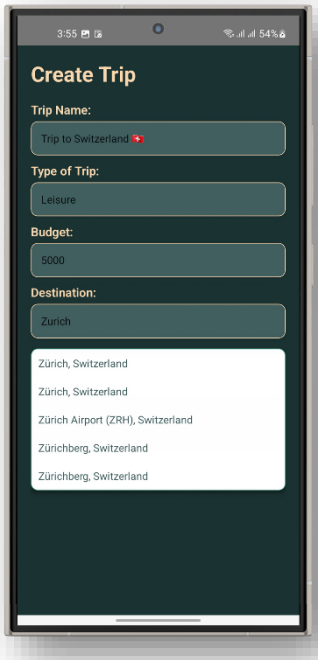

Figure 24 : Auto-Fill 2

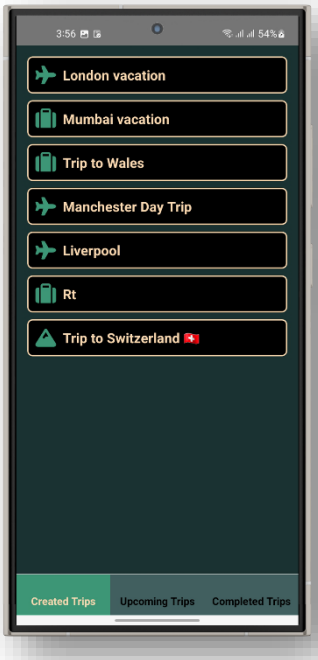

Figure 25 : Created Trips

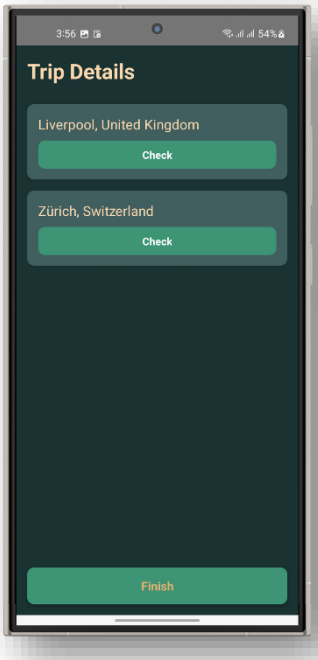

Figure 26 : Verify Location

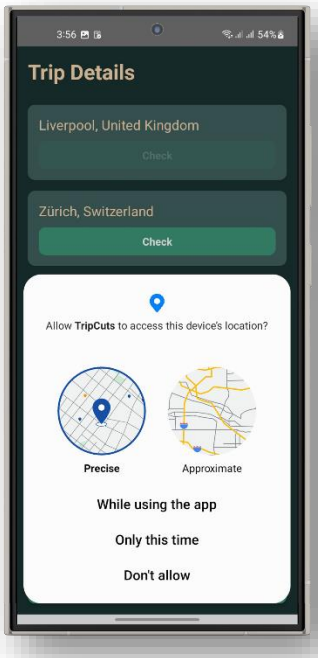

Figure 27 : Location Access Permission

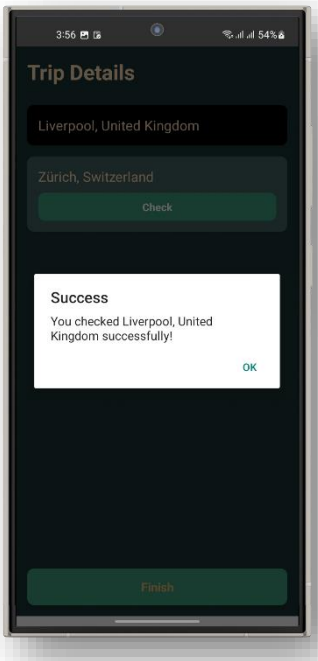

Figure 29 : Location Verified

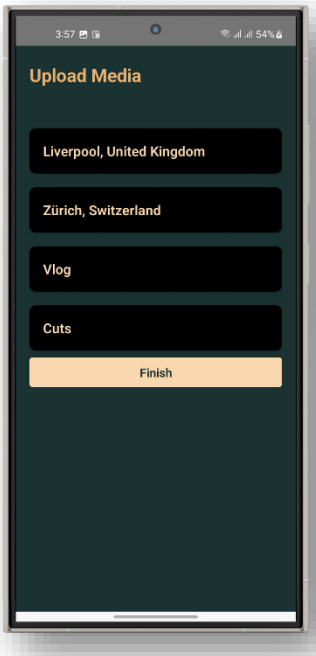

Figure 28 : Media Upload

Figure 30 : Media Select

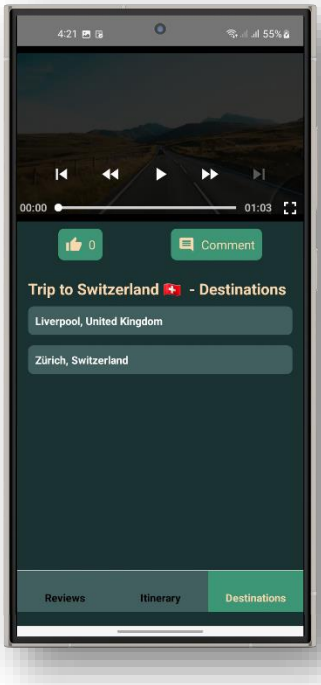

Figure 31 : Trip Created

Figure 32 : Cuts Visible in Profile

Navigating the Trips Created Page: Phase 2

The phase 2 is when the creator actually travels to these places. So, continue the task by validating these destinations the creator has to get into the Trips created page which displays all the trips a creator has initiated (Figure 25). This page is accessible via the bag icon located on the profile page (Figure 21). Once users start their journey, they return to this page to select the active trip they are working on.

When users select a trip, they are directed to the verification page. Here, the entered destinations appear as individual cards. Users must verify their presence at each location by pressing "Check." Verification is achieved only if the user is genuinely present at the specified destination (Figure 29). All destinations must be verified before the user can proceed to the next phase—providing detailed trip information.

Uploading Media and Required Content

After completing the verification phase, creators move to the media upload page (Figure 28, Figure 30). This page allows them to upload media related to each destination. While there are no restrictions on the type or number of images on any other but are allowed to upload only one vlog and one cut per trip.

Uploading media for individual destinations is optional. However, uploading both a vlog and a cut is mandatory to proceed to the next step in the process. Once the required media is uploaded, users press "Finish" to begin building the detailed itinerary.

Creating a Detailed Itinerary

After pressing "Finish," a series of pop-up boxes appears. These prompts collect comprehensive details about the trip. Users may choose to skip certain questions, and skipped questions will not leave blank spaces in the final output; instead, the corresponding sections will be omitted entirely.

The questions include:

- Define your Itinerary
- Transport Options
- Best time to visit
- Safety Tips
- Emergency and Useful Contacts
- Approximate Walking Distance
- Language Tips
- Favourite Place Visited
- Brief Summary

Completing and Publishing the Trip

After answering the questions or skipping them as preferred, users press the final "Finish" button. At this point, the cuts are updated in the user's profile (Figure 32), and the full detailed itinerary page is displayed (Image 31). This is exactly the same as seen in the previous Flow which is verified for the travellers to see (Figure 15,16,17,18).

This structured process ensures that trip details shared on the platform are both authentic and comprehensive, enhancing the value for travellers seeking reliable travel inspiration.

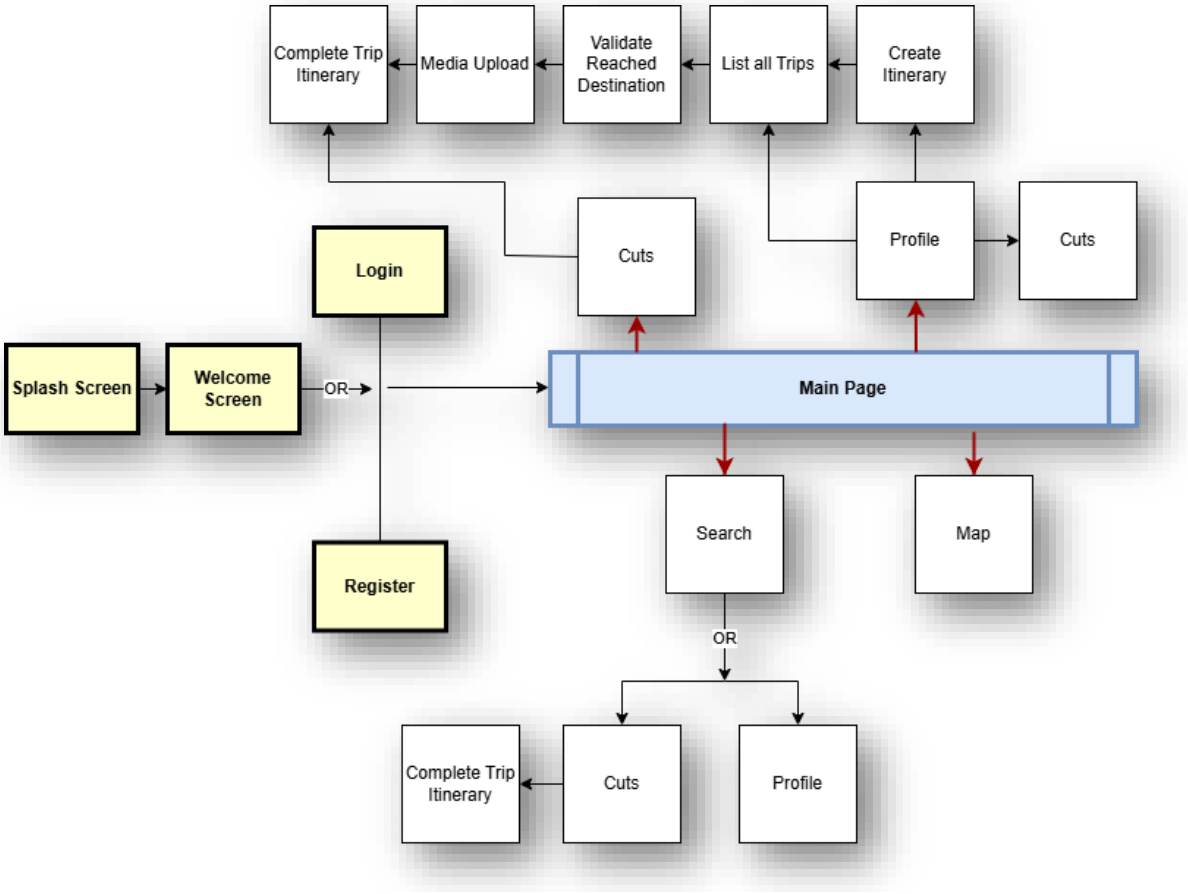

Figure 33 : System Architecture

Implementation

Overview

The project was developed with a strong focus on quality, performance, and maintainability, ensuring compatibility across multiple platforms. Agile methodologies were followed, allowing for iterative development, flexibility, and rapid response to evolving requirements. This approach was essential in meeting both the immediate and long-term needs of the project. A thorough evaluation of both challenges and opportunities before starting the coding process was made. The planning took more time than the actual coding, ensuring a well-defined roadmap that minimized potential roadblocks during development.

System Architecture and Scalability

The system architecture was designed with scalability and cross-platform functionality in mind. The mobile frontend was developed using React Native, ensuring seamless performance on both iOS and Android devices. With scalability for future web platforms in mind, the structure allows for easy adaptation using React Native for Web or Expo. On the frontend, efficient state management, optimized rendering, and reusable components were implemented to guarantee a smooth and responsive user experience across mobile devices.

On the backend, Node.js was used to create a scalable and efficient server environment. As a lightweight JavaScript runtime, it allows for full-stack development using the same language on both the frontend and backend, streamlining the development process and ensuring better code cohesion. Express, a minimal and flexible Node.js framework, was chosen to build the RESTful API. Express simplifies the creation of routes, middleware, and request handling, providing a robust foundation for the backend. MongoDB was selected as the database to handle flexible, document-based storage. Being a NoSQL database, MongoDB supports a dynamic schema, allowing for easier and more flexible data modelling. It is particularly useful for handling unstructured or semi-structured data, which aligns well with the nature of the travel-related content in the application. Its document-based storage also enables faster queries and scalability, which is essential as the user base and data volume grow. This combination of technologies provided the foundation for a robust, high-performance backend that supports secure authentication, seamless data processing, and API interactions.

The app was built to target the latest version of Android and iOS, ensuring maximum compatibility and taking advantage of the newest features and optimizations. By targeting the latest releases, the app provides a smoother, faster, and more secure experience for users. This approach also ensures compatibility with modern devices, enhanced battery performance, and improved security measures. Supporting the latest versions of Android and iOS guarantees that the app remains relevant and functional for years to come, while also ensuring compliance with the latest app store standards and enhancing the overall user experience.

API Versioning

The backend incorporates API versioning, ensuring that future updates to the API will not disrupt existing services or features. This versioning strategy provides a layer of flexibility, allowing new features or improvements to be added while maintaining backward compatibility.

Middleware Extensibility

To enhance flexibility and scalability, the backend includes middleware extensibility. "Middlewares" like "bigPromise" and "error Handler" can easily be extended to accommodate additional features such as input validation or logging, ensuring the application can evolve to meet future needs without a complete overhaul. The middleware structure also supports error handling and centralized management of API responses, which simplifies debugging and enhances maintainability.

Environment Configuration

Environment configuration was streamlined using environment variables, making it easy to manage sensitive data. This approach, enabled through .env files, ensures secure and configurable environment management, reducing the risks of exposing sensitive credentials in the codebase.

Version Control and Deployment

Throughout the development process, GitHub and version control played a key role. By leveraging GitHub for version control, clear tracking of changes was ensured, facilitating smooth branching strategies and enabling continuous integration and deployment (CI/CD) workflows. This provided a stable development environment and streamlined the deployment process.

Algorithm Integration and API Usage

This application effectively integrates algorithms developed in-house, along with carefully selected APIs, to ensure optimal user experience and functionality.

Frontend File Structure

The frontend architecture of the application has been meticulously organized to ensure maintainability, scalability, and clarity. Each folder and file serve a specific purpose, contributing to a seamless development process:

src/: The root directory housing the main application logic.

- *assets/:* Contains all static resources such as images, fonts, and icons. These are used across screens and components to maintain consistency.
- *screens/:* Encapsulates the core screens of the application, such as HomeScreen, ProfileScreen, TripDetailScreen, etc. Each screen is modular, with its styles and logic, promoting code separation.
- *components/:* Houses reusable UI elements like buttons, cards, and headers. This modular approach ensures consistency and reduces code duplication.
- *utils/:* Includes helper functions and utilities such as date formatters, API request handlers, and common constants, ensuring code reusability.
- *authContext/:* used to provide the authentication state (like whether the user is logged in or not) and related functions (such as login, logout, and user data fetching) to any component in your app without having to pass these props manually through each component.

This structure ensures a clear hierarchy, where each component and module have a well-defined role, making the application scalable and maintainable.

Backend File Structure

The backend is organized to support scalability, maintainability, and efficient development practices. The structure reflects a logical separation of concerns:

1. *controllers/:*
   Contains application logic for handling API endpoints.
2. *middlewares/:*
   - *bigPromise.js:* A utility to handle asynchronous errors efficiently in route handlers.
   - *customError.js:* Centralized error-handling logic to provide consistent error responses.
3. *models/:*
   Includes schema definitions using Mongoose for MongoDB. For instance, the User model defines the structure and methods for user-related operations.
4. *routes/:*
   Organizes application endpoints into distinct modules, ensuring a clean separation of concerns.
5. *utils/:*
   - cookieToken.js: A helper function for securely storing tokens in cookies.
   - emailHelper.js: Handles email operations such as sending password reset links.

This modular and layered approach ensures that the backend remains easy to scale and adapt to future requirements, supporting seamless integration with the frontend.

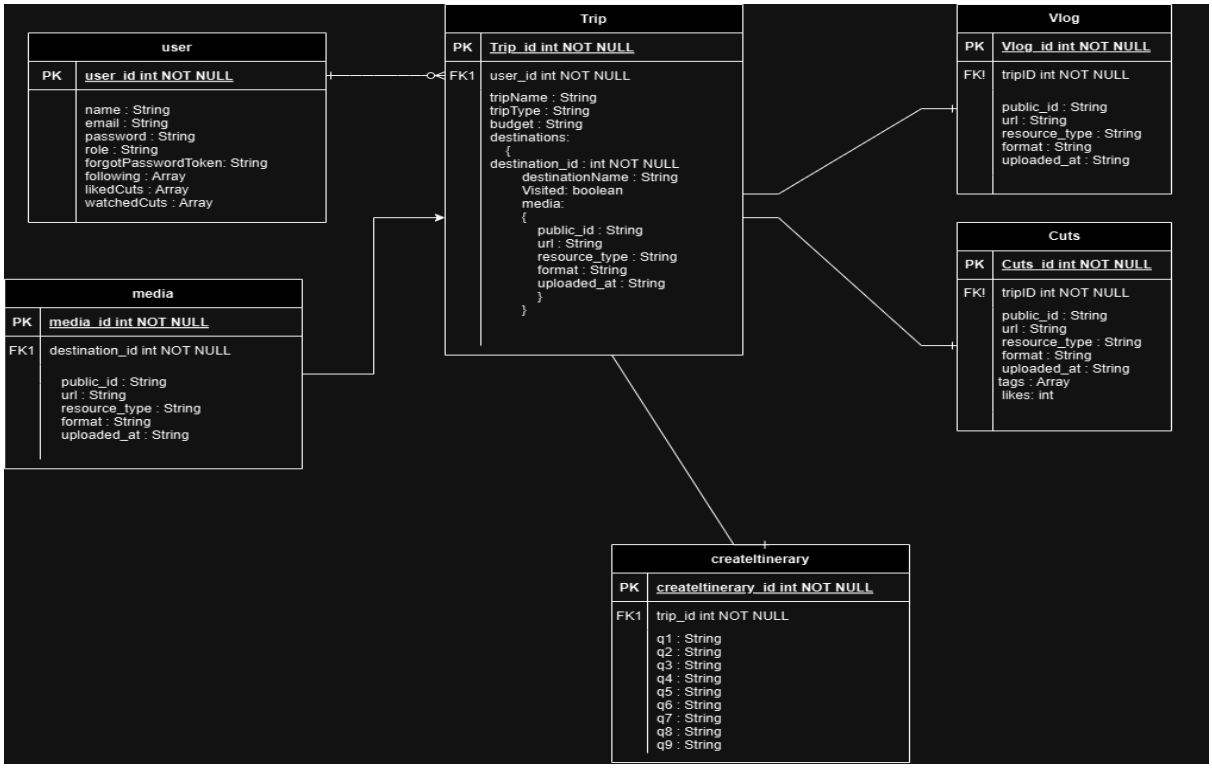

Figure 34 : Database design

User

Authentication

The authentication system for the application was thoughtfully designed to ensure security, reliability, and a smooth user experience. The system integrates a user-friendly frontend with a robust backend to handle essential functionalities like signup, login, and password recovery.

Signup, Login, and Forgot Password

The authentication process combines intuitive design on the frontend with secure data handling and validation on the backend.

1. *Form Validation and Handling*
   The frontend uses Formik and Yup to manage form states and validate inputs for the signup, login, and forgot password forms. These libraries ensure that inputs like email and passwords are correct before submission. Any errors, such as invalid email formats or missing fields, are displayed in real time to guide users.

2. *Backend Processes:*

   - *Signup*: Once validated, the frontend sends the user's name, email, and password to the backend via an API call. The backend verifies the input and creates a new user in the MongoDB database using the User model. Passwords are securely hashed with bcrypt before storage. Upon successful registration, a JWT is generated and sent through body, allowing the user to stay logged in.

   - *Login:* The backend retrieves the user's details based on the email and validates the password using a custom method. If authentication is successful, a secure JWT is issued and sent to establish the session.

   - *Forgot Password:* The backend generates a time-sensitive token when a user requests a password reset. This token is hashed and stored in the database. A password reset link containing the token is sent to the user's email, allowing them to securely reset their password via the frontend.

3. *Error Handling and Feedback*: Both the frontend and backend handle errors consistently. Validation errors are displayed directly beneath input fields, while backend responses, such as "Invalid credentials" or "Email not registered," are presented as clear and user-friendly messages.

4. *Design and Navigation*: The signup, login, and forgot password pages share a cohesive design, ensuring a seamless user experience. Navigation links, such as "Forgot Password?" and "Don't have an account? Sign Up," allow users to move between screens effortlessly.

Account Management Features

Beyond authentication, the system includes features for managing user accounts, such as viewing and updating profile details and changing passwords. These features are designed to empower users while ensuring the highest level of security.

1. *Fetching Logged-In User Details:* The backend processes authenticated requests for user information through the getLoggedInUserDetails endpoint.

- Middleware extracts the user ID from the JWT stored in the HTTP-only cookie.
- Using the user ID, the backend retrieves the corresponding record from the MongoDB database and returns it to the frontend.
  This enables dynamic updates and personalized content based on the logged-in user's profile.

2. *Updating User Details:* Users can modify their profile information, including their name, email, or profile picture.
    - The backend validates the input data to ensure that unauthorized changes are not allowed.
    - If a new profile picture is uploaded, the backend uses Cloudinary to remove the existing image and store the new one. The updated details are saved in the database, and a success message is sent back to the frontend.
      This process ensures that user data remains accurate and secure.
3. *Changing Passwords*: The "changePassword" endpoint allows users to securely update their passwords.
    - The backend verifies the old password by comparing it with the hashed password stored in the database.
    - If the old password matches, the new password is hashed with bcrypt and updated in the database.
    - This process ensures that only the account owner can modify the password, maintaining the integrity of user accounts.

Main Page

Cuts

The Cuts feature is a cornerstone of the platform

Frontend

The frontend of the Cuts feature is developed using React Native, ensuring compatibility across iOS and Android platforms. The design prioritizes user engagement and ease of navigation through various functionalities.

Videos are rendered using the "FlatList" component for efficient scrolling and dynamic loading, with each video occupying the entire screen for an immersive experience. The react-native-video library powers the playback, offering features such as autoplay, pause, and repeat. Gestures are central to the user experience: a single tap pauses or plays the video and a double tap registers a like. Swipe gestures allow seamless navigation, with upward and downward swipes switching videos and rightward swipes redirecting users to trip details.

Backend Processes

The backend lays the foundation for managing video data, delivering personalized recommendations, and supporting user interactions. It uses a series of well-defined APIs to ensure seamless communication between the frontend and backend, enabling efficient data flow.

- *Recommendation Engine:*
  The recommendation engine, accessed via the /getRecommendedVideos endpoint, powers the personalized feed by analyzing the user's profile, including liked videos, watch history, and followed creators, to determine preferences. Tags associated with the content help infer user interests, and the engine applies content-based filtering to prioritize relevant videos. To maintain variety, trending content—determined by metrics like likes and views—is included as fallback recommendations. If neither personalized nor trending options are sufficient, the engine provides a random selection of new videos the user has not interacted with, ensuring the feed remains fresh and engaging. The algorithm adjusts dynamically based on real-time user interactions, leveraging MongoDB's aggregation framework for efficient querying, tag-based scoring, and managing exclusions. This approach ensures the platform remains scalable and responsive as the user base grows.
- *User Engagement Tracking:*
  Interaction endpoints such as /incrementVideoViews and /likeVideo play a key role in tracking user activity. The /incrementVideoViews endpoint updates video view counts in real time while adding the viewed content to the user's watch history. Similarly, the /likeVideo endpoint modifies the likes array for the video and updates the user's likedCuts list. These actions continuously refine the recommendation algorithm, ensuring that the platform consistently aligns with user preferences and delivers a personalized experience.

- *Video Retrieval*: The /getCuts endpoint retrieves videos tailored to a user's trips or preferences while excluding content that has already been liked or watched, ensuring a fresh and engaging experience for the user. Additionally, the /searchUsersAndCuts endpoint enhances discoverability by matching search terms against usernames and trip titles, making it easy for users to find creators or destinations effortlessly.

Profile

The page layout focuses on delivering a personalized experience, displaying the user's profile details, follower counts, and a collection of uploaded cuts.

Key components include:

- *Profile Header:* Displays the user's profile picture, username, bio, and follower/following counts.
- *Video Grid:* Implements a visually appealing masonry grid of uploaded videos. Each video tile is styled with random colours and shapes to make the grid dynamic and engaging.
- *Video Modal:* Opens a full-screen modal for video playback when a video is selected. The modal includes playback controls and an "Itinerary" button that navigates users to the corresponding trip details.

- *Floating Buttons:* Two action buttons – "Bag" and "Create Trip" – are positioned for easy access. These buttons allow users to view their saved trips and create new trips.

The FlatList component efficiently renders the video grid, enabling dynamic loading and smooth scrolling. Videos are managed using the react-native-video library, ensuring smooth playback.

Gestures enhance interactivity:

- A tap opens the video modal.
- Navigation gestures redirect users to trip details or other sections of the app.

Backend Processes

The backend for the Profile Page is powered by a robust architecture that supports efficient data management and seamless communication with the frontend. It employs a series of APIs to handle user data and video content:

- */getCuts Endpoint:* Fetches videos associated with the logged-in user. The backend filters and retrieves video data, ensuring the content is relevant and fresh. This data is displayed dynamically on the frontend.
- */incrementVideoViews Endpoint:* Tracks video views in real-time. Each view updates the video's view count and the user's watch history, enhancing the recommendation engine's ability to deliver personalized content.
- */likeVideo Endpoint:* Updates the likes count of a video and the user's likedCuts list whenever a like action is performed.
- */getUserDetails Endpoint:* The backend retrieves user-specific data such as the profile picture, username, bio, and follower/following counts via this endpoint. This data is dynamically sent to the frontend to populate the Profile Header, ensuring the user sees an accurate and personalized summary of their profile.

Itinerary Creation

This multi-step process integrates advanced APIs, robust backend systems, and a dynamic frontend to provide users with a smooth and interactive experience.

Trip and Destination Suggestions

The first step in creating an itinerary involves entering trip details, including the destination. To enhance user experience:

- *Mapbox API Integration*: Mapbox API is used to provide destination suggestions as users type. This ensures accurate and location-specific inputs.
- *Form Validation*: Formic and Yup libraries validate the input fields for required details such as trip name, type, budget, and destinations. Once validated, the details are sent to the backend through the /createTrip API for secure storage.

Verifying Destinations

Destination verification ensures that users have genuinely visited the listed locations, maintaining the authenticity of shared itineraries:

1. *Fetching Trips:* The "/getTrips" API retrieves all trips associated with the logged-in user, allowing the frontend to display destinations dynamically.
2. *Geolocation Integration:*
    - The app leverages the react-native-geolocation-service library to fetch the user's current GPS coordinates upon tapping the "Check" button next to a destination.
    - Permissions are handled through react-native-permissions, ensuring that user privacy is respected. Alerts guide users if permissions are denied.
3. *Reverse Geocoding:*
    - The Mapbox API converts GPS coordinates into readable location names.
    - The app validates the current location against the destination name, ensuring the user is physically present at the destination.
4. *Database Update:*
    - Successful verification triggers the /updateVisited API to mark the destination as "visited" in the database.
    - This process ensures accurate check-ins while handling potential errors like denied permissions or API issues.

Media Upload

The media upload phase allows users to document their trip with photos and videos for each destination.

Dynamic Frontend for Media Management

- *Expandable Destination Cards:* Each destination card can be expanded to upload and manage media files.
- *Vlogs and Cuts Uploads:* Users can upload a single video for vlogs and cuts in distinct sections.
- *Media Previews:* Uploaded media files are displayed as thumbnails, providing immediate feedback to users.

Media Selection

The react-native-image-picker library enables users to:

- Select multiple media files (images/videos) for each destination.
- Choose a single video for vlogs and cuts.

Selected media is stored locally in the component state until uploaded, ensuring smooth performance.

Cloudinary Integration

The app uses Cloudinary for optimized media storage:

- *Uploading Files:* Media files are uploaded to Cloudinary using APIs that handle file compression and optimization, ensuring quick retrieval during playback.
- *Metadata Storage:* Cloudinary responses, including URLs and metadata, are sent to the backend via the /saveMedia API to associate files with specific destinations.

Interactive Itinerary Questions

To finalize the itinerary, users are guided through a series of interactive questions:

- *Modal Interface:* A modal displays sequential questions designed to capture detailed trip information, such as transport options, best time to visit, and safety tips.
- *State Management:* User responses are stored in a state and submitted to the backend through the /createItinerary API.
- *Dynamic Navigation:* Upon completing the questionnaire, users are redirected to the "TripLandingPage", where the finalized itinerary is displayed.

Backend *Processes*

1. *Fetching Trip Details:* The /getTrips endpoint retrieves user-specific trips, enabling the frontend to dynamically display destinations and statuses.
2. *Uploading Media*:
    - Media files are uploaded to Cloudinary for secure storage and compressed for efficient retrieval.
    - The /saveMedia endpoint associates uploaded files with the respective trip and destination.
3. *Vlogs and Cuts Uploads*: Separate APIs (/createVlog and /createCuts) handle vlogs and short video cuts, storing them securely in the database.
4. *Submitting Itinerary*: The /createItinerary endpoint stores user responses, linking them to the trip for future access and updates.

Search

The SearchPage implementation introduces a search interface integrated with backend APIs to allow users to search for travel-related content, including user profiles and short videos ("cuts").

Search Input and Animation

The search functionality is initiated with an animated slide-in input field, providing a polished and engaging visual experience. The Animated API is used to slide the input field into view when the screen is loaded, and the useRef hook ensures that the search input field is immediately focused for user convenience.

Search Query Handling

As the user types, the "fetchSearchResults" function, debounced using "lodash.debounce", sends the query to the backend API "/searchUsersAndCuts". This ensures minimal API calls by only triggering the search after the user pauses typing. Queries shorter than three characters are ignored to reduce unnecessary backend calls.

Dynamic Search Results

The results, fetched from the backend, include two categories:

- *Cuts:* Short travel videos associated with trips.
- *Users:* Profiles of content creators or travellers.

These results are combined into a single list and displayed dynamically using a FlatList component. If no results are found, a user-friendly message appears, such as "No matches found."

Interactive Result Display

Each search result is rendered as a clickable item:

- *Cuts:* Clicking a cut opens a modal to preview the video using the react-native-video component. The modal also includes an "Itinerary" button that navigates the user to the detailed itinerary associated with the cut.
- *Users:* Clicking a user redirect to the corresponding traveller's profile using their unique ID.

Modal for Video Preview

For selected cuts, a full-screen modal displays the video with playback controls. A floating "Itinerary" button within the modal allows users to navigate to the itinerary page associated with the video, enhancing content discoverability.

Backend *Processes*

The search functionality is tightly integrated with the backend API:

- *Search Endpoint:* The "/searchUsersAndCuts" endpoint accepts the search query and returns matching users and cuts. The backend processes the query with filters for names (for users) and trip names (for cuts), ensuring relevant and accurate results.
- *Error Handling:* Robust error handling ensures that users are notified of any connectivity or server issues, maintaining a smooth user experience.

Map Integration

The map feature leverages the Mapbox Navigation SDK to provide users with real-time turn-by-turn navigation. I have also used a third-party API to make the UI in action. Using the Mapbox Places API, the application suggests source and destination based on inputs. Once the user selects their desired locations, the corresponding latitude and longitude are retrieved, ensuring accurate routing tailored to the user's needs.

Deployment and Hosting

Backend Deployment on Heroku

To get the backend up and running, I decided to go with Render as the hosting platform. Render makes deploying and scaling applications super easy, so it was the perfect choice for this project. I started by creating a new Render service and linking it to the Node.js backend repository. Once the service was live, I made sure to securely store sensitive information, like API keys and the database connection string, using Render's environment variables. This kept everything safe and ensured that no confidential details were exposed.

To make life even easier, I set up continuous deployment. This means that whenever I push updates to the GitHub repository, Render automatically deploys the changes, keeping everything up to date with no extra effort on my part. To handle different levels of traffic, I enabled Render's auto-scaling feature, which means the app can scale up or down as needed, without me having to worry about it. Plus, Render's monitoring tools give me a clear view of how the app is performing, making it easy to spot and fix any issues quickly.

.

Database Deployment on MongoDB Atlas

For storing user data, itineraries, and other important travel information, MongoDB Atlas was used as the cloud database solution. MongoDB Atlas is a powerful, secure, and scalable option for managing databases. The connection to the database was securely set up using a connection string, which was stored as an environment variable on Heroku to keep it safe.

MongoDB Atlas also offered automatic backups, encryption, and strong security features like role-based access control, ensuring that data was always protected. With features like IP whitelisting, access to the database was limited to trusted sources, preventing unauthorized users from gaining access to sensitive information.

Frontend Transformed into SDK

For the frontend, instead of directly hosting the React Native code, the app was turned into an SDK (Software Development Kit). This approach ensured that the source code stayed secure and protected from misuse. By bundling the frontend into an SDK, it could be safely distributed to trusted partners or internal teams who needed it, but without exposing the app's code to the public. This is over usage of the application might lead to billing in the backend hosting.

Testing

To ensure the travel platform app delivered a smooth and reliable user experience, a simple testing process covering functionality, performance, and security was done. Since the hosted platforms were paid and was being hosted in a free-tier tests were not made extensively.

The testing process began with **unit testing** to verify individual components, such as itinerary creation, user profiles, and trip recommendations. Tools like Jest and React Testing Library were used to ensure that the core features worked as expected before being integrated into the broader app. This approach helped identify and resolve issues early, establishing a solid foundation for the app.

Next, **integration testing** was conducted to evaluate how well the app's various components worked together. For example, the connection between the user profile and the backend API

was tested to ensure that data synced correctly. Mock data was used to simulate real-time interactions, enabling testing of data flow without relying on live information.

User testing was a key part of the process, providing valuable insights into the app's usability. Real users tested the app, offering feedback on design and functionality. Based on this feedback, adjustments were made to improve elements like the layout of the profile page and the navigation of itineraries, ensuring a user-friendly experience.

For **performance testing**, tools like Neoload were used to simulate multiple users interacting with the app simultaneously. This helped identify potential slowdowns, especially in media-heavy features such as video cuts. Stress-testing the app ensured that it could handle high traffic while maintaining optimal performance. But this was done in a very low level of only 1 user per minute.

Security was prioritized, with **security testing** carried out to ensure user data remained protected. Key features, including login authentication and data encryption, were thoroughly tested to adhere to best practices for safeguarding sensitive information.

Conclusion

This platform does more than just facilitate travel planning, it fosters a global community of like-minded travellers, helping them find inspiration, share experiences, and create memories. The success of the platform lies in its ability to merge technology with human experience, offering a rich, immersive, and personalized approach to travel that aligns with modern-day needs.

Future updates will include enhanced collaboration and travel group features, allowing users to invite collaborators to edit or view itineraries with permissions managed by roles such as "editor" and "viewer." Users will also be able to create and manage travel groups, simplifying the process of planning and hosting trips together.

Offline support will be added, enabling users to access and edit itineraries without an internet connection. Once reconnected, data will automatically sync. In addition, multi-language support will be introduced to make the platform more accessible globally, catering to a wider range of users.

A paid query feature will allow travellers to receive personalized advice, providing a tailored experience. The introduction of GraphQL will also enable more flexible and efficient querying, further enhancing personalization options for users.

Live location sharing will also be added, allowing users to share their real-time location with their travel group for safety and coordination. Additionally, fall detection and alert functionality will be introduced to help ensure the safety of travellers, especially in unfamiliar or risky locations. If the system detects a fall, it will send an immediate alert to emergency contacts.

Another feature will notify users automatically when they enter an area that has not been highly rated by other users. This user-generated rating system will help provide real-time insights into potentially unsafe or undesirable areas, helping travellers make informed decisions on where to go.

To improve the overall experience, the focus will be on detecting memory leaks, testing scalability using tools like Neoload, and continuously refining the user experience (UX) to ensure a smooth and intuitive interface. These performance improvements will help maintain efficiency as the platform grows.

Lastly, API documentation will be enhanced with tools like Swagger to streamline integration for developers, making it easier to interact with and utilize the platform's API for building new features and applications.

Bringing this flow to life was a challenging yet rewarding process. It involved integrating various elements such as the algorithm, video cuts at multiple touchpoints, itinerary creation, media, and essential information—all intricately connected in a way where each component depended on the others to function properly. At times, it felt overwhelming as the success of the entire system hinged on how well these parts worked together. However, seeing the final product come to life not only validated the effort but also sparked excitement and optimism for the future enhancements that can make the application even more dynamic and a one stop application for any kind of travellers.

References

Camilleri, M.A., Troise, C. & Kozak, M. 2023. "Functionality and usability features of ubiquitous mobile technologies: The acceptance of interactive travel apps." *Journal of Hospitality and Tourism Technology.* doi:https://doi.org/10.1108/JHTT-12-2021-0345.

Du, Xin, Toni Liechty, Carla A. Santos, and Jeongeun Park. 2020. ""'I Want to Record and Share My Wonderful Journey': Chinese Millennials' Production and Sharing of Short-Form Travel Videos on TikTok or Douyin."." *Current Issues in Tourism* 25 (21): 3412–24. doi:10.1080/13683500.2020.1810212.

Dymkov, Artem D. 2021. "TRAVEL ROUTE PLANNING AND TRACKING APPS." *SYNCHROINFO JOURNAL* (IRIS Association, Vienna, Austria) 22-31. doi:10.36724/2664-066X-2021-7-2-22-31.

Garry Wei-Han Tan, Voon Hsien Lee. 2017. "Mobile applications in tourism: the future of the tourism industry?" *Industrial Management & Data Systems* Vol. 117 No. 3, pp. 560-581. doi:https://doi.org/10.1108/IMDS-12-2015-0490.

Huertas A, Míguez-González MI and LozanoMonterrubio N. 2017. "YouTube usage by Spanish tourist destinations as a tool to communicate their identities and brands." *Journal of Brand Management* 24(3): 211–229. doi:10.1057/s41262-017-0031-y.

Jian Xie, Kai Zhang, Jiangjie Chen, Tinghui Zhu, Renze Lou,Yuandong Tian◊ Yanghua Xiao, Yu Su. 2024. *TravelPlanner: A Benchmark for Real-World Planning with Language Agents.* The Pennsylvania State University. https://osu-nlp-group.github.io/TravelPlanner.

Jinyi Xu, Guanghui Qiao & Songhe Hou. 2023. "Exploring factors influencing travel information-seeking intention on short video platforms." *Current Issues in Tourism* 3985-4000. doi:10.1080/13683500.2022.2154197.

Kim, B., Yoo, M., & Yang, W. 2020. "Online engagement among restaurant customers: The importance of enhancing flow for social media users." *Journal of Hospitality & Tourism Research* 44(2), 252–277. doi:https://doi.org/10.1177/1096348019887202.

Liu J, Wang Y and Chang L. 2023. "How do short videos influence users' tourism intention? A study of key factors." *Frontier in Psychology* 13. doi:doi.org/10.3389/fpsyg.2022.1036570.

Lius Steven Sanjaya, Ferdianto, Titan, Johan. 2017. "Mobile Application Business Plan to Assist Travel Planning." *International Conference on Information Management and Technology (ICIMTech).* Melia Purosani Hotel, Yogyakarta, Indonesia: Bina Nusantara University. 144-149.

Lu, X., & Lu, Z. 2019. "Fifteen seconds of fame: A qualitative study of Douyin, a short video sharing mobile application in China." *International Conference on Human-Computer Interaction* 233–244. doi:10.1007/978-3-030-21902-4_17.

Meltem Caber, Tahir Albayrak, Sezer Karasakal & Maria Rosario González Rodríguez. 2024. "Building customer citizenship behaviour through mobile application quality: the mediating role of flow experience and customer engagement." *Current Issues in Tourism* 2918-2933. doi:10.1080/13683500.2023.2241606.

Payne, Will. 2024. *Cruise Trade News.* 01 11. Accessed 11 2024, 20. https://www.cruisetradenews.com/consumers-unperturbed-by-introduction-of-ai-in-travel-study-finds/.

Phoebe Yueng-Hee Sia, Siti Salina Saidin,Yulita Hanum P. Iskandar. 2023. "Systematic review of mobile travel apps and their smart features and challenges." *Journal of Hospitality and Tourism Insights* 2115-2138. doi:10.1108/JHTI-02-2022-0087.

Phuong Minh Binh Nguyen, Lan Xuan Pham, Dang Khoa Tran and Giang Nu. 2024. "A systematic literature review on travel planning through user-generated video." *Journal of Vacation Marketing* Vol. 30(3) 553–581. doi:10.1177/13567667231152935.

Pudliner, B. A. 2007. "Alternative Literature and Tourist Experience: Travel and Tourist Weblogs." *Journal of Tourism and Cultural Change* 5(1), 46–59. doi:10.2167/jtcc051.0.

Reali, Cristóbal. 2024. *Top 5 Social Media Platforms for Travel Agencies and How to Make the Most of Them.* 18 11. Accessed 11 20, 2024. https://mize.tech/blog/top-social-media-platforms-for-travel-agencies/.

Sharif, Kaitlin Woolley and Marissa A. 2022. *Harvard Buisness Review.* 31 Jan. Accessed Nov 14, 2024. https://hbr.org/2022/01/the-psychology-of-your-scrolling-addiction.

Smith, D. N., & Chen, X. 2018. "Brand experience, flow and brand app loyalty: Examining consumer decision making within branded mobile apps." *Marketing Management Journal* 28(2), 145–160. http://www.acrwebsite.org/volumes/12575/volumes/v34/NA-34.

Tahir Albayrak, M. Rosario González-Rodríguez,Meltem Caber,Sezer Karasaka. 2023. "The Use of Mobile Applications for Travel Booking: Impacts of Application Quality and Brand Trust." *Journal of Vacation Marketing* Vol. 29(1) 3–21. doi:10.1177/13567667211066544.

Woolf, Max. 2024. *photoAiD.* 16 October. Accessed November 14, 2024. https://photoaid.com/blog/how-social-media-affects-travel-statistics/?srsltid=AfmBOop4jhYoBEktdQ3mzJHf4Zkc1c0jSNBICOc6vlSYz2pQIrz-kzXe.

X, Wu G and Ding. 2023. "Which type of tourism short video content inspires potential tourists to travel." *Frontiers in Psychology* 14:1086516 ,. doi:10.3389/fpsyg.2023.1086516.

Y, Li H and Liu. 2014. "Understanding post-adoption behaviours of e-service users in the context of online travel services." *Information and Management* 51(8): 1043–1052.

Yu, J., Lee, H., Ha, I. and Zo, H. 2017. "User acceptance of media tablets: An empirical examination of perceived value." *Telematics and Informatics* Vol. 34 No. 4, pp. 206-223.

Zhou Y, Liu L and Sun X. 2022. "The effects of perception of video image and online word of mouth on tourists' travel intentions: Based on the behaviors of short video platform users." *Frontier in Psychology* 13. doi:10.3389/fpsyg.2022.984240.

APPENDIX

Important Codding Snippets

Video Recommendation Algorithm

```
exports.getRecommendedVideos = BigPromise(async (user_id) => {
  try {
      const user = await User.findById(user_id)
          .populate('likedCuts')
          .populate('watchedCuts._id')
          .populate('following');

      if (!user) {
          throw new CustomError('User not found', 404);
      }

      const tags = [];
      user.likedCuts.forEach(video => tags.push(...video.tags));

      user.watchedCuts.forEach(watched => {
          if (watched._id && Array.isArray(watched._id.tags)) {
              tags.push(...watched._id.tags);
          }
      });

      // Get content-based recommendations
      let contentBasedCuts = await Cuts.find({
          tags: { $in: tags },
          _id: { $nin: user.likedCuts.map(v => v._id).concat(user.watchedCuts.map(v => v._id)) }
      }).limit(10);

      // If not enough content-based recommendations, fetch trending
      if (contentBasedCuts.length < 10) {
          const trendingCuts = await Cuts.find({
              _id: { $nin: user.likedCuts.map(v => v._id).concat(user.watchedCuts.map(v => v._id)) }
          }).sort({ likes: -1, views: -1 }).limit(10 - contentBasedCuts.length);

          contentBasedCuts = [...contentBasedCuts, ...trendingCuts];
      }

      // If still not enough, show random videos that the user hasn't watched
      if (contentBasedCuts.length < 10) {
          const remainingVideos = await Cuts.find({
              _id: { $nin: user.likedCuts.map(v => v._id).concat(user.watchedCuts.map(v => v._id)) }
          }).limit(10 - contentBasedCuts.length);

          // Shuffle remaining videos to add some randomness
          const shuffledRemainingVideos = remainingVideos.sort(() => 0.5 - Math.random()).slice(0, 10 - contentBasedCuts.length);
          contentBasedCuts = [...contentBasedCuts, ...shuffledRemainingVideos];
      }

      return contentBasedCuts;
  } catch (error) {
      console.error('Error getting recommended Cuts:', error);
      throw new CustomError('Failed to get recommended Cuts', 500);
  }
});
```

FlatList for Dynamic Viewing

```
<FlatList
  ref={flatListRef}
  data={videos}
  renderItem={({item, index}) =>
    item ? (
      <PanGestureHandler onGestureEvent={handlePanGesture}>
        <TapGestureHandler
          waitFor={doubleTapRef}
          onActivated={handleSingleTap}>
          <TapGestureHandler
            ref={doubleTapRef}
            numberOfTaps={2}
            onActivated={handleDoubleTap}>
            <LongPressGestureHandler
              ref={longPressRef}
              onHandlerStateChange={({nativeEvent}) => {
                if (nativeEvent.state === State.ACTIVE) {
                  handleLongPress();
                }
              }}>
              <View style={styles.videoContainer}>
                <Video
                  source={{uri: item.uri}}
                  style={styles.video}
                  resizeMode="cover"
                  paused={paused || activeIndex !== index}
                  repeat
                />
              </View>
            </LongPressGestureHandler>
          </TapGestureHandler>
        </TapGestureHandler>
      </PanGestureHandler>
    ) : null
  }
  keyExtractor={(item, index) => `${item.id}-${index}`}
  snapToAlignment="center"
  snapToInterval={height}
  decelerationRate="fast"
  showsVerticalScrollIndicator={false}
  onViewableItemsChanged={handleViewableItemsChanged}
  viewabilityConfig={{
    itemVisiblePercentThreshold: 50,
  }}
/>
```

Detailed Authentication Endpoints

```
Pieces: Comment | Pieces: Explain
exports.signup = BigPromise(async (req, res, next) => {···
});

Pieces: Comment | Pieces: Explain
exports.login = BigPromise(async (req, res, next) => {···
});

Pieces: Comment | Pieces: Explain
exports.logout = BigPromise(async (req, res, next) => {···
});

Pieces: Comment | Pieces: Explain
exports.forgotPassword = BigPromise(async (req, res, next) => {···
});

Pieces: Comment | Pieces: Explain
exports.passwordReset = BigPromise(async (req, res, next) => {···
});

Pieces: Comment | Pieces: Explain
exports.getLoggedInUserDetails = BigPromise(async (req, res, next) => {···
});

Pieces: Comment | Pieces: Explain
exports.changePassword = BigPromise(async (req, res, next) => {···
});

Pieces: Comment | Pieces: Explain
exports.updateUserDetails = BigPromise(async (req, res, next) => {···
});

Pieces: Comment | Pieces: Explain
exports.adminAllUser = BigPromise(async (req, res, next) => {···
});

Pieces: Comment | Pieces: Explain
exports.admingetOneUser = BigPromise(async (req, res, next) => {···
});

Pieces: Comment | Pieces: Explain
exports.adminUpdateOneUserDetails = BigPromise(async (req, res, next) => {···
});

Pieces: Comment | Pieces: Explain
exports.adminDeleteOneUser = BigPromise(async (req, res, next) => {···
});

Pieces: Comment | Pieces: Explain
exports.managerAllUser = BigPromise(async (req, res, next) => {···
});
```

Efficient use of Cloudinary as Media Database

```
const uploadVlogOrCutsToCloudinary = async (
  media: {uri: string; type: string} | null,
  mediaType: 'Vlog' | 'Cuts',
) => {
  if (!media || !media.uri) {
    Alert.alert(
      'Error',
      `Please select a ${mediaType.toLowerCase()} video to upload`,
    );
    return;
  }

  try {
    const formData = new FormData();
    formData.append('file', {
      uri: media.uri,
      type: media.type,
      name: `${mediaType.toLowerCase()}.mp4`, // Naming convention for media
    });
    formData.append('upload_preset', CLOUDINARY_UPLOAD_PRESET);
    formData.append('folder', mediaType); // Upload to the corresponding folder: "Vlog" or "Cuts"

    const response = await axios.post(
      `https://api.cloudinary.com/v1_1/${CLOUDINARY_CLOUD_NAME}/upload`,
      formData,
      {
        headers: {
          'Content-Type': 'multipart/form-data',
        },
      },
    );

    const url = response.data.url;
    const public_id = response.data.public_id;
    const format = response.data.format;
    const asset_id = response.data.asset_id;
    const resource_type = response.data.resource_type;
    const tripName = tripDetails?.tripName || 'Default Trip Name';

    if (mediaType === 'Vlog') {
      await updateVlogInDB(
        url,
        public_id,
        format,
        asset_id,
        resource_type,
        tripId,
      );
    } else if (mediaType === 'Cuts') {
      await updateCutsInDB(
        url,
        public_id,
        format,
        asset_id,
        resource_type,
        tripName,
        tripId,
      );
    }
```

Search Algorithm Front End

```
const fetchSearchResults = async (query: string) => {
  if (query.length < 3) {
    setResults([]);
    setNoResultsMessage('');
    return;
  }

  try {
    const response = await fetch(
      '${BASE_URL}/api/v1/searchUsersAndCuts',
      {
        method: 'POST',
        headers: {
          'Content-Type': 'application/json',
        },
        body: JSON.stringify({query}),
      },
    );

    const data = await response.json();
    if (data.success) {
      const filteredCuts = data.cuts.filter((cut: Cut) =>
        cut.tripName.toLowerCase().includes(query.toLowerCase()),
      );
      const filteredUsers = data.users.filter((user: User) =>
        user.name.toLowerCase().includes(query.toLowerCase()),
      );

      const combinedResults = [...filteredCuts, ...filteredUsers];
      setResults(combinedResults);
      setNoResultsMessage(
        combinedResults.length === 0 ? 'No matches found.' : '',
      );
    } else {
      setResults([]);
      setNoResultsMessage('No matches found.');
    }
  } catch (error) {
    console.error(error);
    setResults([]);
    setNoResultsMessage('No matches found.');
  }
};
```

Search Algorithm Back End

```
exports.searchUsersAndCuts = BigPromise(async (req, res, next) => {
  // Log the request body for debugging
  console.log(req.body);

  // Destructure search term from request body
  const { query } = req.body;

  // Check if query term is provided
  if (!query) {
      return next(new CustomError("Search term required", 400));
  }

  try {
      // Find users whose usernames match the query, case-insensitive
      const users = await User.find({ name: { $regex: query, $options: 'i' } });

      // Find cuts whose titles match the query, case-insensitive
      const cuts = await getCuts.find({ tripName: { $regex: query, $options: 'i' } });

      // If no users or cuts are found, return a 404 status with a message
      if (!users.length && !cuts.length) {
          return res.status(404).json({
              success: false,
              message: "No results found for this search term",
          });
      }

      return res.status(200).json({
          success: true,
          users,
          cuts,
      });
  } catch (error) {
      return res.status(500).json({
          success: false,
          message: error.message,
      });
  }
});
```

Location suggestions by MapBox API

```
// Fetch location suggestions using Mapbox API
const handleSearch = async (text: string) => {
  setQuery(text);
  if (text.length > 2) {
    try {
      const response = await axios.get(
        `https://api.mapbox.com/geocoding/v5/mapbox.places/${encodeURIComponent(text)}.json`,
        {
          params: {
            access_token: MAPBOX_ACCESS_TOKEN, // Access token from environment variable
            autocomplete: true,
            limit: 5,
          },
        }
      );

      const formattedSuggestions = response.data.features.map(
        (feature: {place_name: string}) => {
          const parts = feature.place_name.split(',');
          const city = parts[0];
          const country = parts[parts.length - 1].trim();
          return `${city}, ${country}`;
        },
      );
      setSuggestions(formattedSuggestions);
    } catch (error) {
      console.error(error);
    }
  } else {
    setSuggestions([]);
  }
};
```

Verify Destination

```
// Request location permission from the user
const requestLocationPermission = async () => {
  try {
    const result = await request(
      Platform.OS === 'ios'
        ? PERMISSIONS.IOS.LOCATION_WHEN_IN_USE
        : PERMISSIONS.ANDROID.ACCESS_FINE_LOCATION,
    );

    if (result === RESULTS.GRANTED) {
      console.log('Location permission granted.');
      return true;
    } else {
      console.log('Location permission denied.');
      Alert.alert(
        'Permission Denied',
        'Location permission is required to access your location.',
      );
      return false;
    }
  } catch (error) {
    console.error('Failed to request location permission:', error);
    Alert.alert(
      'Error',
      'Failed to request location permission. Please try again later.',
    );
    return false;
  }
};
```

```
// Fetch trip details and initialize state
useEffect(() => {
  const fetchTripDetails = async () => {
    const user_id = await getUserID();

    try {
      const response = await axios.post(
        '${BASE_URL}/api/v1/getTrips',
        {user_id},
      );
      const trip = response.data.trips.find((t: any) => t._id === tripId);
      setTripDetails(trip);

      if (trip.destinations.every((dest: Destination) => dest.visited)) {
        navigation.goBack(); // Go back once
        navigation.navigate('MediaUpload', {tripId});
      } else {
        setLoading(false);
      }
    } catch (error) {
      console.error('Error fetching trip details:', error);
      setLoading(false);
    }
  };

  fetchTripDetails();
}, [tripId, navigation]);
```

```
const handleCheck = async (destinationName: string) => {
  const hasPermission = await requestLocationPermission();
  if (hasPermission) {
    console.log('Fetching current position...');
    Geolocation.getCurrentPosition(
      async position => {
        const {latitude, longitude} = position.coords;
        const currentPlaceName = await reverseGeocode(latitude, longitude);

        if (currentPlaceName.includes(destinationName.split(',')[0].trim())) {
          try {
            await axios.post(
              '${BASE_URL}/api/v1/updateVisited',
              {
                tripId,
                destinationName,
              },
            );

            setTripDetails((prevDetails: TripDetails | null) => ({
              ...prevDetails!,
              destinations: prevDetails!.destinations.map(
                (dest: Destination) =>
                  dest.destinationName === destinationName
                    ? {...dest, visited: true}
                    : dest,
              ),
            }));

            Alert.alert(
              'Success',
              `You checked ${destinationName} successfully!`,
            );
          } catch (error) {
            console.error('Error updating destination:', error);
            Alert.alert('Error', 'Failed to update destination status.');
          }
        } else {
          Alert.alert('Failed', `You are not at the destination yet.`);
        }
      },
      error => {
        let errorMessage =
          'Unable to retrieve location. Please check your device settings.';

        switch (error.code) {
          case 1:
            errorMessage =
              'Location permission denied. Please allow access to your location.';
            break;
          case 2:
            errorMessage =
              'Location position unavailable. Please try again later.';
            break;
          case 3:
            errorMessage = 'Location request timed out. Please try again.';
            break;
          default:
            errorMessage = 'An unknown error occurred. Please try again.';
            break;
        }

        Alert.alert('Error', errorMessage);
      },
      {enableHighAccuracy: true, timeout: 3000, maximumAge: 1000},
    );
  }
};
```