A PROJECT ON

Sereinly : AI Based Travel Recommendation System

Submitted in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING (B.Tech. CSE)

Submitted by:

Shivyanshu	**2019112**

Under the Guidance of

Dr. Neha Tripathi

Associate Professor

Project Team ID: ID No. MP24CSE085

Department of Computer Science and Engineering
Graphic Era (Deemed to be University)
Dehradun, Uttarakhand
June-2025

Graphic Era Deemed to be University DEHRADUN

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the Project Report entitled **"Sereinly : AI Based Travel Recommendation System"** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering and submitted in the Department of Computer Science and Engineering of the Graphic Era (Deemed to be University), Dehradun is an authentic record of my own work carried out during a period from **August-2024 to June-2025** under the supervision of **Dr. Neha Tripathi**, **Associate Professor**, Department of Computer Science and Engineering, Graphic Era (Deemed to be University).

The matter presented in this dissertation has not been submitted by me/us for the award of any other degree of this or any other Institute/University.

Shivyanshu 2019112

This is to certify that the above statement made by the candidate is correct to the best of our knowledge.

Supervisor **Head of the Department**

External Viva

Name of the Examiners:	**Signature with Date**
1.	
2.	

Abstract

Planning a trip today involves navigating a fragmented digital landscape—researching destinations, comparing hotels, checking transportation, and organizing daily activities often across multiple platforms. This complexity increases when catering to different travel groups such as solo explorers, couples, families, and friends. To address these challenges, **Sereinly** is introduced as an AI-powered travel planning platform that simplifies and personalizes the trip planning experience.

Built using **React.js** for the frontend, **Appwrite** for backend services and database management, **Tailwind CSS** and **Syncfusion** for UI design, and **Stripe** for payment integration, Sereinly enables users to join curated, AI-generated trips created exclusively by administrators. At the core of the system is the **Gemini Generative AI API**, which produces personalized, multi-day itineraries based on destination, group type, interests, and budget.

A dedicated **Admin Dashboard** allows authorized users to manage system activity, generate AI trips, and view platform analytics such as total users and active enrolments. End-users can browse available trips and join them via a secure payment workflow, with access to full itineraries granted upon successful checkout.

Initial testing confirms the platform's effectiveness in delivering real-time responses, intelligent itinerary creation, and a smooth user experience. Sereinly demonstrates how full-stack development and generative AI can come together to reshape how modern travellers plan and experience their journeys.

Acknowledgement

Any achievement, be in scholastic or otherwise does not depend solely on the individual effort but on the guidance, encouragement and co-operation of intellectuals, elders and friends. A number of personalities in their own capacity have helped me in carrying out this project work.

Our sincere thanks to project guide **Dr. Neha Tripathi, Associate Professor,** Department of Computer Science and Engineering, Graphic Era (Deemed to be University), for his valuable guidance and support throughout the course of project work and for being a constant source of inspiration.

We extend our thanks to **Dr. Deepak Gaur**, Project coordinator, Department of Computer Science and Engineering, Graphic Era (Deemed to be University), for his valuable suggestions throughout all the phases of the Project Work.

We are extremely grateful to **Prof. (Dr.) D. P. Singh**, HOD of the Computer Science and Engineering Department, Graphic Era (Deemed to be University), for his moral support and encouragement.

We thank the **management of Graphic Era (Deemed to be University)** for the support throughout the course of our Bachelor's Degree and for all the facilities they have provided.

Last, but certainly not least we thank all teaching and non-teaching staff of Graphic Era (Deemed to be University) for guiding us in the right path. Most importantly we wish to thank our parents for their support and encouragement.

Shivyanshu 2019112

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Project Introduction

Travel planning, whether for solo exploration, romantic getaways, family vacations, or group adventures, remains a complex and time-consuming process. From selecting destinations to organizing accommodations, transportation, and daily activities, the modern traveller is overwhelmed with fragmented tools and endless options across the internet. There exists a growing need for a unified, intelligent system that simplifies and personalizes this experience.

Sereinly is introduced as a smart, AI-powered travel planning platform that addresses this gap. Designed using a modern tech stack—**React.js** for the frontend, **Appwrite** for backend and database services, **Tailwind CSS** and **Syncfusion** for a clean UI, **Gemini Generative AI API** for trip generation, and **Stripe** for secure payments—Sereinly provides a comprehensive digital solution. The platform centralizes the planning process by allowing administrators to generate rich, multi-day itineraries using AI, while users can explore available trips and join them upon successful payment.

The system features a dedicated **Admin Dashboard** for managing trip generation, user analytics, and platform statistics. Users benefit from an intuitive interface, fast loading times, and access to high-quality AI-generated travel plans suited to their interests, budget, and preferences.

1.2 Problem Statement

Despite the availability of countless travel websites and booking platforms, users often find it difficult to organize their trips efficiently. The process typically involves visiting multiple websites for flights, hotels, sightseeing ideas, and itinerary creation, resulting in information overload and disjointed planning.

Key issues include:

- Lack of real-time personalized itinerary generation.

- No integration between planning and booking systems.
- Limited user-centric experiences in current travel platforms.
- Absence of role-based controls where only verified admins can create content.
- Lack of intelligent data visualization for platform administrators.

There is a strong need for an AI-integrated platform that can not only generate customized itineraries but also allow users to seamlessly discover and join trips with minimal effort.

1.3 Objectives

The primary objectives of the Sereinly platform are as follows:

- To automate the generation of personalized, multi-day travel itineraries using Generative AI.
- To provide a clean, responsive, and user-friendly interface for browsing and enrolling in trips.
- To restrict trip creation capabilities to administrators only, ensuring quality and consistency.
- To integrate a secure payment gateway using Stripe for smooth user enrolment.
- To offer a centralized admin dashboard for monitoring users, trips, and analytics in real-time.
- To enhance the travel experience for different user categories: solo travellers, couples, families, and groups of friends.

Chapter 2

Literature Survey/ Background

The application of Artificial Intelligence (AI) in travel and tourism has undergone rapid evolution in the past decade, reshaping how travellers plan, explore, and experience their journeys. With the increase in demand for personalized travel experiences and the overwhelming volume of options available online, AI-driven solutions have become increasingly vital to simplify the trip planning process. The aim of this literature survey is to explore various research efforts, existing platforms, and technologies that have laid the groundwork for developing intelligent travel planning systems like **Sereinly**.

2.1 AI in Travel Planning

Several research studies have highlighted the potential of AI in enhancing travel experiences. AI can extract patterns from user behaviour, analyse sentiment, and use natural language understanding to offer highly personalized recommendations. For example, Ullrich et al. (2021) developed a machine learning-based approach for gait analysis, demonstrating how sensor data and pattern recognition can yield actionable insights—an approach that inspires user preference prediction in travel systems .

Other academic efforts have explored AI-based recommendation systems that consider contextual factors like time, location, and user intent. These systems often leverage collaborative filtering, content-based filtering, and hybrid models to suggest destinations, hotels, and activities. However, many of these approaches are limited by the absence of real-time updates or integration with dynamic data sources such as live hotel availability, weather, or public transport schedules.

2.2 Existing Travel Planning Platforms

A number of AI-powered platforms have emerged to assist users in travel planning:

- **Roam Around** and **Layla** use GPT-based engines to generate travel itineraries. While they provide a conversational interface for trip generation, they suffer from a lack of real-time data integration and limited ability to personalize based on budget, mood, or local events.
- **TripHobo**, **Sygic Travel**, and **Inspirock** allow users to manually build itineraries with location-based suggestions. However, they rely heavily on user inputs and do not

employ adaptive learning or AI-driven personalization. Moreover, these platforms generally do not provide API-based integrations for live flight, hotel, or weather data.

- **Google Travel** offers a partial solution with itinerary suggestions and hotel booking, but lacks deep user customization and emotion-aware planning.

These limitations present an opportunity for an intelligent platform that combines AI, NLP, and real-time data integration to deliver smarter, more intuitive trip planning.

2.3 NLP and LLMs in User Preference Analysis

The recent advancements in Natural Language Processing (NLP) and large language models (LLMs) like GPT-4 and Gemini have opened up new possibilities in user interaction. By processing user prompts, LLMs can extract specific preferences such as type of destination (beach, mountain, historical), travel duration, budget, and even emotional intent (e.g., relaxing vs. adventurous trips). However, most current tools lack the capability to merge these insights with a functional backend or dynamic frontend for real-time interaction and editing.

Projects using GPT-based trip planners tend to output static results, making it difficult for users to modify or interact with generated itineraries. Furthermore, these models often hallucinate or include outdated information if not integrated with real-time APIs like Google Maps, Skyscanner, or Booking.com.

2.4 Full Stack Implementation in Travel Apps

In modern web application development, full-stack architectures such as **MERN** (MongoDB, Express.js, React.js, Node.js) have traditionally been used to build scalable travel apps. However, newer platforms like **Appwrite** have begun replacing traditional stacks by offering a simpler, API-driven backend-as-a-service with built-in authentication, databases, and serverless functions.

On the frontend, **React.js** combined with **Tailwind CSS** and component libraries like **Syncfusion** allows for the rapid creation of dynamic, responsive interfaces that support real-time updates and complex data visualizations.

Sereinly combines these technologies to build a platform that is not only scalable and secure, but also modular, allowing administrators to control AI-based trip generation while providing users with smooth exploration and payment flows.

2.5 Gaps Identified

From the surveyed systems and literature, the following limitations remain:

- Lack of dynamic, role-based systems that distinguish between trip creators and users.
- No secure, integrated payment system for joining pre-generated trips.
- Absence of real-time dashboards and visual insights for administrators.
- Static itinerary generation without AI adaptability based on budget, group size, and interest type.
- Minimal personalization and no continuous learning based on user history.

2.6 Contribution of the Proposed System

Sereinly addresses these gaps through an integrated approach that blends AI, full-stack development, and user-centric design. Its key contributions include:

- An admin-exclusive interface for generating AI-based travel plans using the **Gemini API**.
- A secure payment integration with **Stripe**, ensuring smooth user enrolment.
- A **React.js** frontend supported by **Syncfusion** components for elegant UI design.
- A centralized **Admin Dashboard** for monitoring platform metrics such as total users, active trips, and AI-generated plans.
- Real-time data storage and access control using **Appwrite**, replacing traditional backend logic.

This intelligent platform positions itself as a next-generation solution for smart, scalable, and accessible travel planning.

Chapter 3

Software Design

Software design is a pivotal phase in the software development life cycle (SDLC) where the theoretical understanding of user requirements is translated into a practical, functional blueprint. The success of the system relies heavily on the robustness, modularity, and efficiency of the design. This chapter provides a comprehensive overview of the design principles, architectural decisions, data flow, and component-level specifications used in the development of **Sereinly** – an AI-based travel itinerary planner.

3.1 Design Methodology

The design of Sereinly follows a modular, scalable, and role-based architecture that separates concerns between the user interface, backend services, database management, and AI integration. The system was structured using principles such as separation of concerns, reusability of components, and secure access control. Administrators and general users operate in distinct roles, enabling content creation and consumption workflows to be logically and securely managed.

The frontend uses a component-based architecture developed in **React.js**, promoting reusability and easy state management. Components for viewing trips, generating AI responses, processing payments, and managing dashboards are built as isolated modules. The backend logic, powered by **Appwrite**, ensures seamless user authentication, secure data storage, and function-based APIs for trip and payment handling. The AI interaction is integrated via API calls to **Gemini**, where prompts are crafted dynamically based on admin input.

Sereinly follows an event-driven design pattern for operations like payment confirmation and trip enrolment, ensuring real-time feedback and consistent state updates. The frontend communicates with backend services through API calls while Appwrite handles authentication, database interaction, and secure file management.

3.2 System Architecture

The overall architecture of Sereinly is based on a **three-tier model**: the presentation layer, the application logic layer, and the data/AI services layer.

3.2.1 Presentation Layer (Frontend)

The frontend is built using **React.js**, offering users a seamless, responsive interface. Styling is handled using **Tailwind CSS**, allowing for rapid UI development. To enhance the interface, **Syncfusion UI components** are integrated for data display elements such as tables, cards, and charts.

Users access the trip browsing interface and Stripe-based payment gateway, while admins operate a secure dashboard that includes functionalities like AI trip creation, user monitoring, and performance tracking. Interactive animations, loading states, and visual feedback are managed using lightweight libraries for optimal performance and user experience.

3.2.2 Application Layer (Backend & Logic)

Instead of a traditional server architecture, Sereinly utilizes **Appwrite**, a cloud-native backend-as-a-service platform. Appwrite manages user authentication, access control, database queries, and serverless functions. It enforces role-based logic where only admins can access AI-generation and dashboard features.

Admin inputs for AI trip creation are collected via forms and sent as requests to Appwrite's server functions, which then interact with the **Gemini API**. The returned AI itinerary is stored in Appwrite's database and made visible to general users for browsing and payment.

Payment operations are managed by securely redirecting users to **Stripe's Checkout API**, which handles payment authentication and confirmation. Once a transaction is successful, a webhook updates the Appwrite database to confirm the user's enrolment in the trip.

3.2.3 Data and AI Layer

The data layer comprises Appwrite's built-in **NoSQL-style document database**, used to store:

- User data and authentication tokens
- AI-generated trip data
- Payment confirmations

- Trip enrolment records

The AI integration is built on top of **Gemini’s Generative API**, which receives structured prompts from the backend and returns travel itineraries. These prompts include dynamic details such as the destination, travel dates, group type, budget, and interests. The AI-generated content is parsed and formatted for both storage and frontend display.

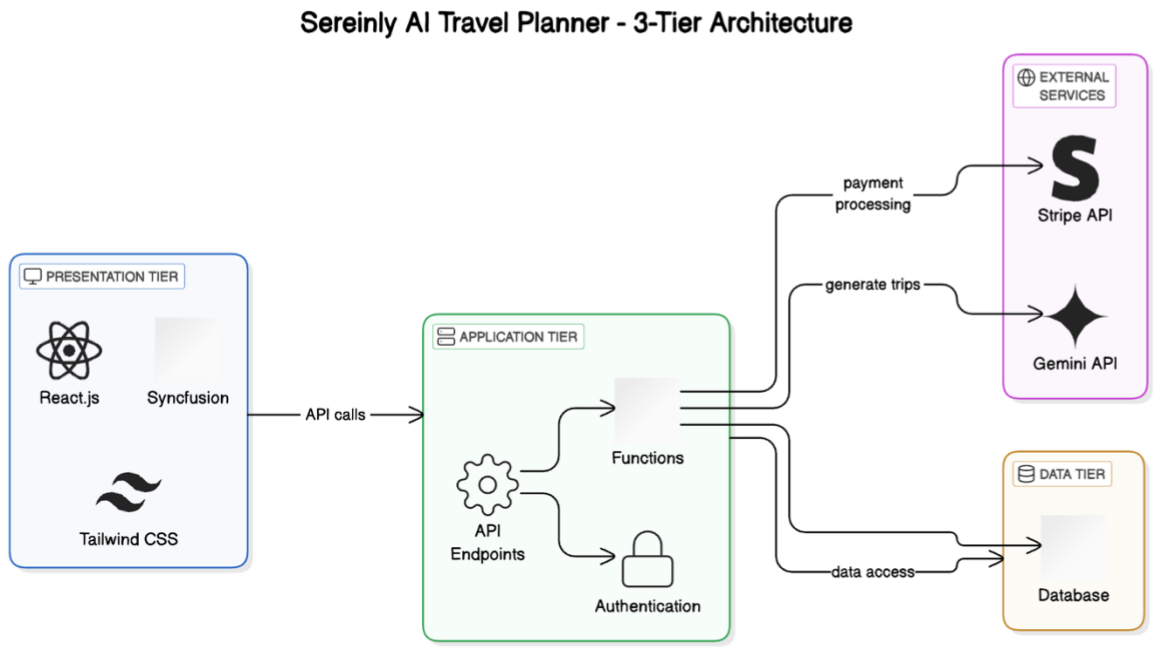


Fig 3.1 A 3 tier architecture diagram for web application

3.3 Flow Diagram and Component Interaction

The flow begins when a general user lands on the platform and browses available trips. Upon selecting a trip, the user is directed to the Stripe checkout for payment. Post-payment, Appwrite updates the user’s access and marks them as enrolled. Meanwhile, admins use the dashboard to enter parameters for trip generation, which are processed by the Gemini AI and stored in the database.

On the frontend, components are structured around logical tasks:

- A trip exploration component for users.
- A secure Stripe checkout module.

- An admin panel for AI trip creation and system monitoring.
- Dashboard visualizations powered by Syncfusion charts and tables.

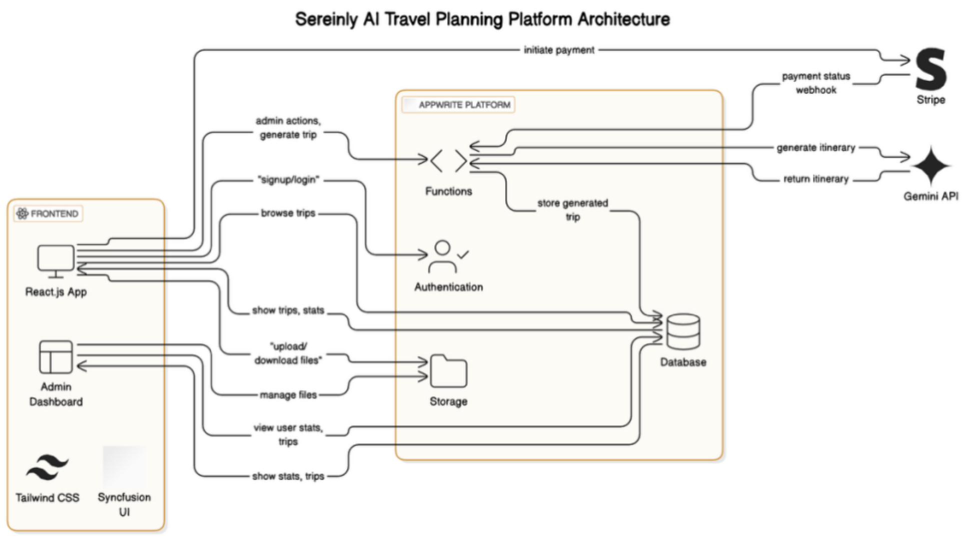


Fig 3.2 System flow diagram for web application

3.4 Sequence Overview

The system's sequence of operations for each role is outlined below:

For **users**:

1. User accesses the home page and selects a trip.
2. Stripe checkout is initiated and completed.
3. Upon successful payment, trip access is granted.
4. User views complete itinerary from dashboard.

For **admins**:

1. Admin logs into a protected dashboard.
2. Admin inputs trip details and submits AI request.
3. Gemini API returns a custom itinerary.
4. The itinerary is stored and published for users.
5. Admin can view analytics such as active trips, users, and revenue.

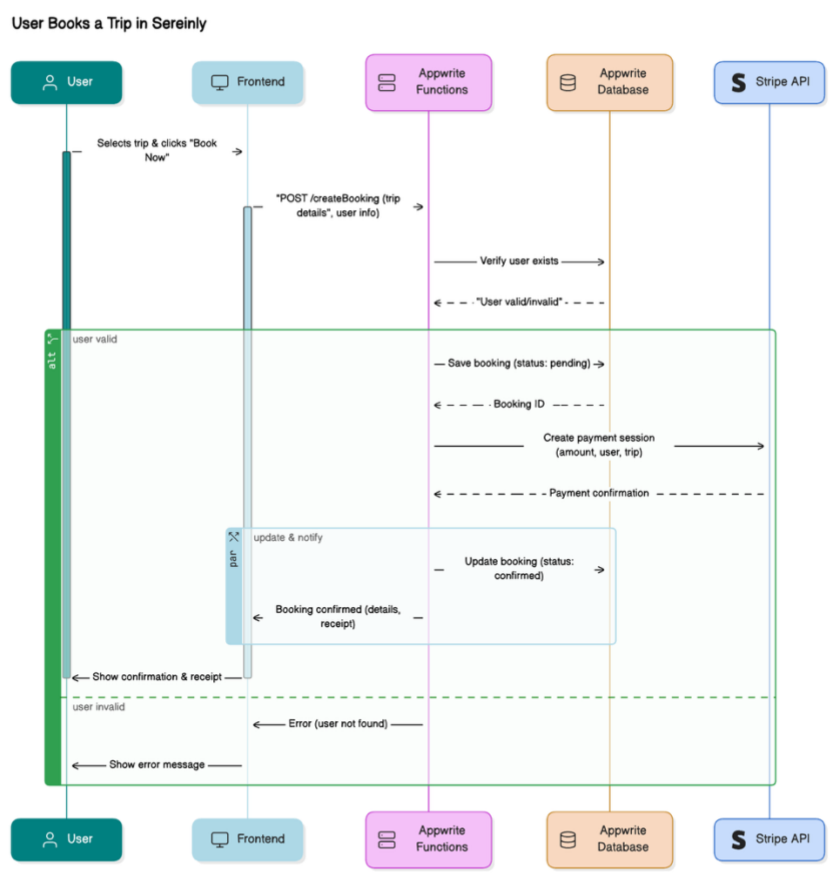

User Books a Trip in Sereinly
User
Frontend
Appwrite Functions
Appwrite Database
Stripe API
Selects trip & clicks "Book Now"
"POST /createBooking (trip details", user info)
Verify user exists
"User valid/invalid"
alt
user valid
Save booking (status: pending)
Booking ID
Create payment session (amount, user, trip)
Payment confirmation
par
update & notify
Update booking (status: confirmed)
Booking confirmed (details, receipt)
Show confirmation & receipt
user invalid
Error (user not found)
Show error message
User
Frontend
Appwrite Functions
Appwrite Database
Stripe API

Fig 3.3 Sequence diagram for web application

3.5 Data Flow Diagrams (DFD)

3.5.1 Level 0 Data Flow Diagram

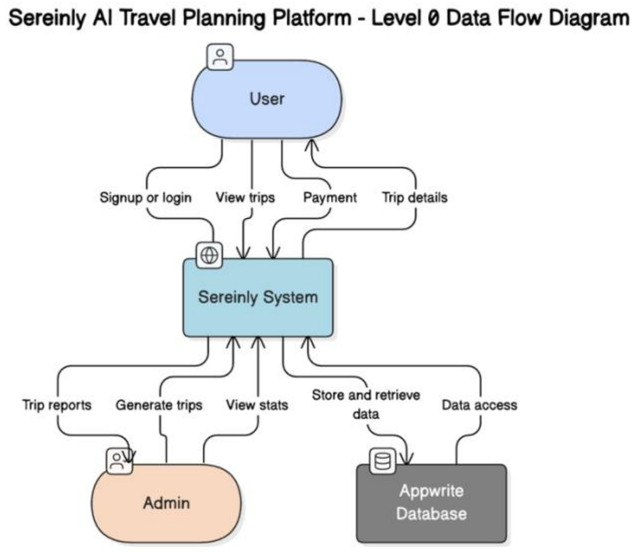


Fig 3.4 level 0 data flow diagram

3.5.2 Level 1 Data Flow Diagram

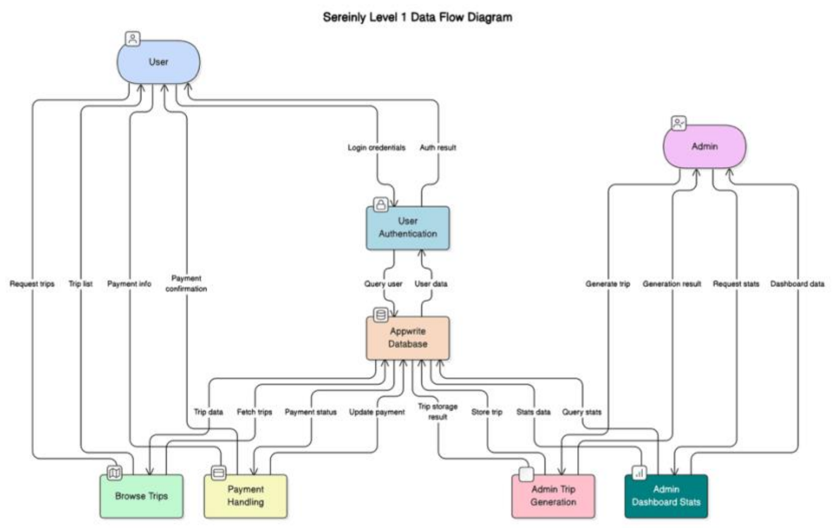


Fig 3.5 Level 1 data flow diagram

3.7 Class Diagram

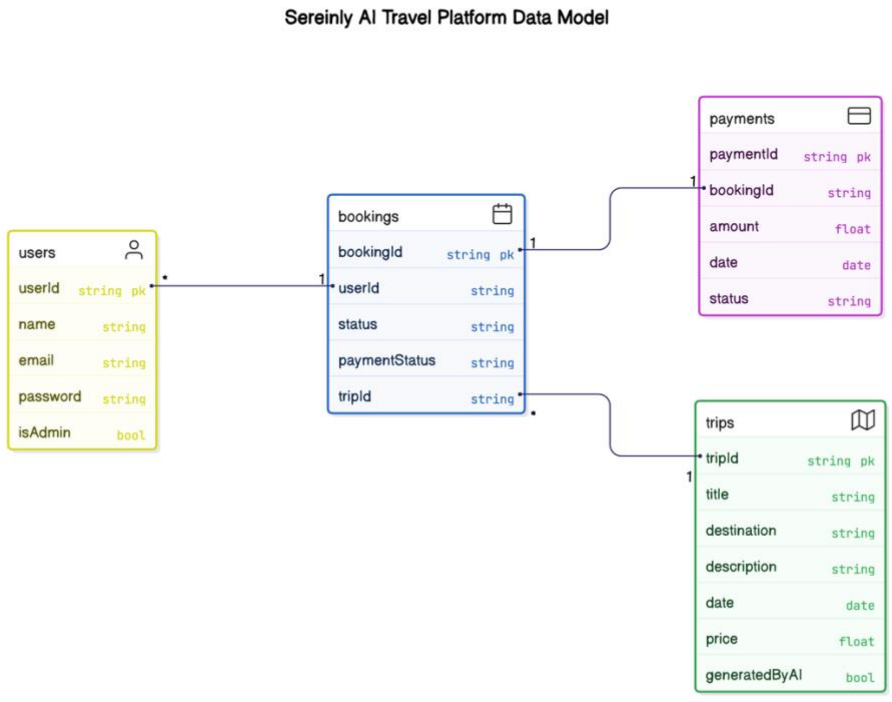


Fig 3.6 Class Diagram

Chapter 4

Requirements and Methodology

This chapter outlines the functional and non-functional requirements necessary for developing *Sereinly*, as well as the methodology adopted for building, testing, and deploying the system. A clear understanding of these aspects ensures that the application aligns with user expectations and project goals while remaining scalable, maintainable, and performance-efficient.

4.1 Requirements

4.1.1 Hardware Requirements

Component	Minimum Requirement
Processor	Intel Core i5 equivalent
RAM	8 GB or higher
Storage	256 GB SSD(recommended)
Graphics	Integrated/Basic GPU
Internet Connection	Required for API access and deployment

Table 4.1 Hardware requirements

4.1.2 Software Requirements

Software	Purpose
React.js	Frontend interface and rendering
Appwrite	Cloud tool for database storage
Tailwind CSS	UI styling and animation
Gemini API	AI-based itinerary generation
Postman	API testing
Git & GitHub	Version control and collaboration
VS Code/ IDE	Source code editing
Netlify/Render/Vercel	Deployment (Frontend/Backend Hosting)

Table 4.2 Software requirements

4.2 Methodology

The system is developed using the **Agile Software Development Life Cycle (Agile SDLC)** model, where the development process is divided into multiple sprints/phases with frequent iterations and refinements based on testing and feedback.

4.2.1 Project Development Lifecycle

The Sereinly platform was developed using the **Agile Software Development Life Cycle (Agile SDLC)**. This methodology emphasizes iterative development, regular feedback, and rapid prototyping. The project was divided into structured phases across multiple sprints to ensure continuous improvement and timely delivery.

In the initial phase, requirement analysis was conducted by identifying user needs and analysing gaps in existing solutions. This included defining the target audience, such as solo travellers, couples, families, and friend groups, and identifying the need for AI-generated itineraries and secure trip enrolment.

During the design phase, the system architecture was finalized, along with the roles of the frontend, backend, and AI integration. Wireframes, component trees, and API schemas were created to define interaction flow. In the implementation phase, frontend components were built using React.js, and backend services were configured using Appwrite. The AI functionality was integrated by constructing dynamic prompts that were sent to the Gemini API for itinerary generation.

In the integration and testing phase, Stripe was connected to handle payment processing. Appwrite serverless functions were tested for proper data handling and access control. Frontend responsiveness, user roles, and the admin dashboard were all tested across devices and use cases to ensure stability. Finally, in the deployment and feedback phase, the application was hosted, and feedback was gathered to plan for enhancements such as user login sessions, dynamic filtering, and support for PDF trip exports.

4.2.2 Key Technologies Used

Technology	Usage
React.js	Build the dynamic user interface
Appwrite	Create backend structure
SyncFusion	React UI library
Gemini API	NLP model to generate personalised trip plans
Tailwind CSS	Streamline the design process with utility-first classes

Table 4.3 Key technologies used

4.2.3 Integration Flow

The operational workflow of Sereinly begins with the administrator logging into the dashboard and entering trip preferences into a form. This data is formatted and sent as a structured prompt to the Gemini API. The API responds with a detailed travel itinerary, which is then saved into the Appwrite database and made available on the frontend.

Users can browse these curated trips without logging in. When they decide to join a trip, they are redirected to a secure Stripe checkout session. Upon successful payment, a webhook notifies Appwrite to confirm the user's enrolment. The database is updated, and the user is granted access to the complete itinerary on their dashboard. All actions—from AI prompt generation to payment confirmation and trip access—occur through tightly coupled APIs and role-specific access flows.

Chapter 5

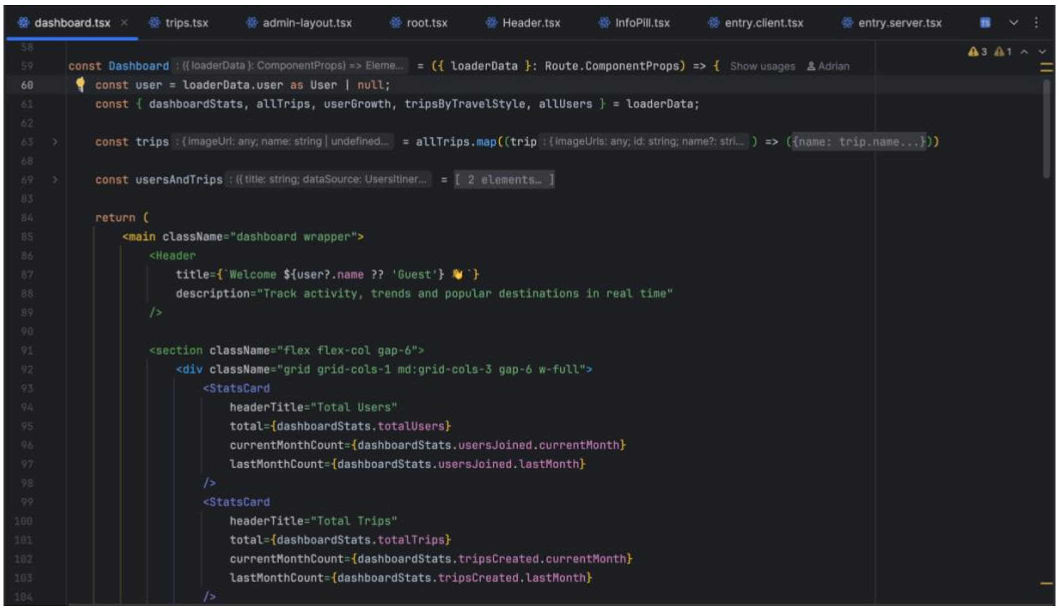
Coding/Code Templates

This chapter provides a glimpse into the actual implementation of the *Sereinly* system. It highlights the core parts of the codebase including key frontend components, backend APIs, database models, and the integration with the AI model. The project is developed using the **MERN stack (MongoDB, Express.js, React.js, Node.js)**, with additional integrations like **Gemini API** for itinerary generation and **Google Maps API** for location handling.

5.1 Frontend Code (React.js)

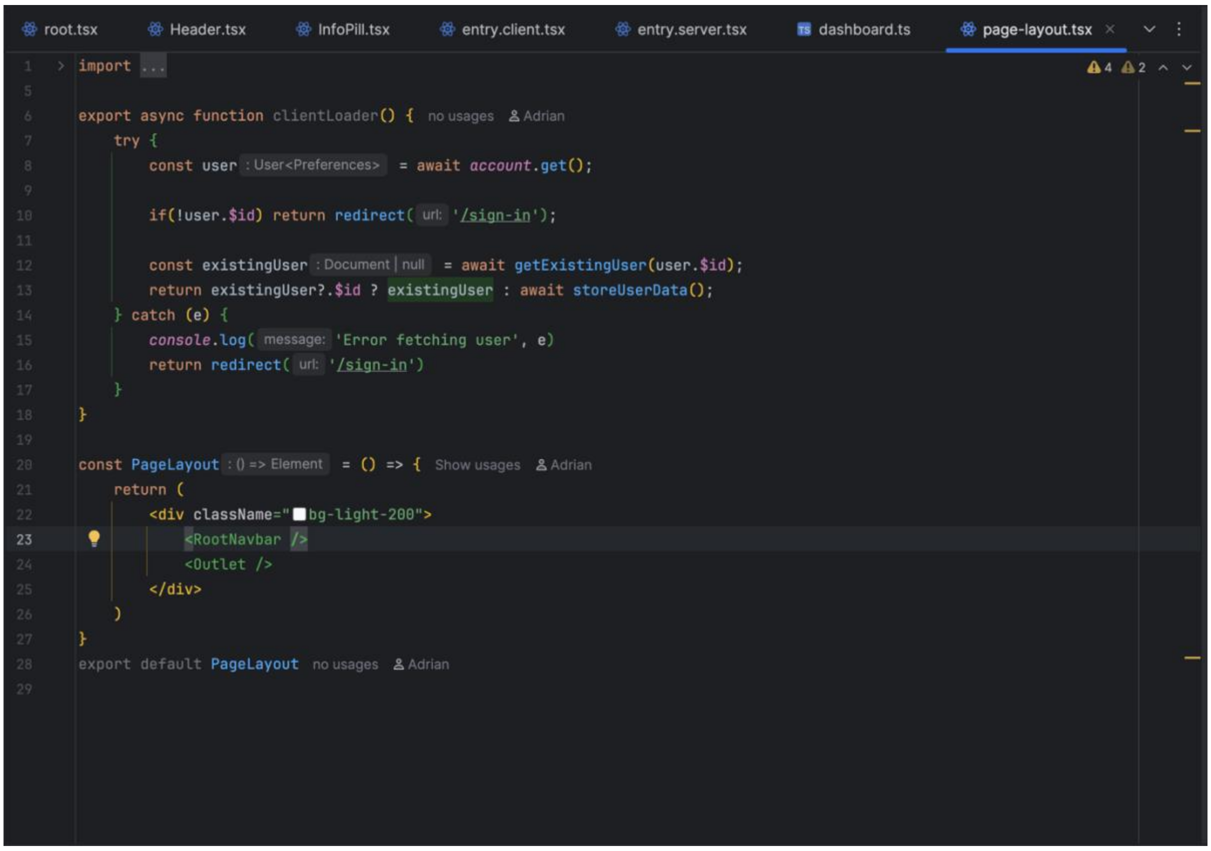
The frontend of Sereinly is built using **React.js**, following a component-driven architecture. Each functionality such as trip exploration, trip joining, and dashboard analytics is encapsulated within separate reusable components. **Tailwind CSS** is used extensively for responsive styling, and **Syncfusion React components** are integrated to manage tables, charts, and data grids efficiently.

One of the critical components is the **TripList** component, which fetches all AI-generated trips from the Appwrite database and displays them dynamically. React's useEffect and useState hooks are utilized for handling data fetching and managing UI state updates.



```
const Dashboard : ({ loaderData }: ComponentProps) => Eleme... = ({ loaderData }: Route.ComponentProps) => {  Show usages  Adrian
    const user = loaderData.user as User | null;
    const { dashboardStats, allTrips, userGrowth, tripsByTravelStyle, allUsers } = loaderData;

    const trips : {imageUrl: any; name: string | undefined... = allTrips.map((trip : {imageUrls: any; id: string; name?: stri... ) => ({name: trip.name...}))

    const usersAndTrips : ({ title: string; dataSource: UsersItiner... = [ 2 elements… ]

    return (
        <main className="dashboard wrapper">
            <Header
                title={`Welcome ${user?.name ?? 'Guest'} 👋`}
                description="Track activity, trends and popular destinations in real time"
            />

            <section className="flex flex-col gap-6">
                <div className="grid grid-cols-1 md:grid-cols-3 gap-6 w-full">
                    <StatsCard
                        headerTitle="Total Users"
                        total={dashboardStats.totalUsers}
                        currentMonthCount={dashboardStats.usersJoined.currentMonth}
                        lastMonthCount={dashboardStats.usersJoined.lastMonth}
                    />
                    <StatsCard
                        headerTitle="Total Trips"
                        total={dashboardStats.totalTrips}
                        currentMonthCount={dashboardStats.tripsCreated.currentMonth}
                        lastMonthCount={dashboardStats.tripsCreated.lastMonth}
                    />
```

Fig. 5.1 Code Snippet for dashboard.tsx

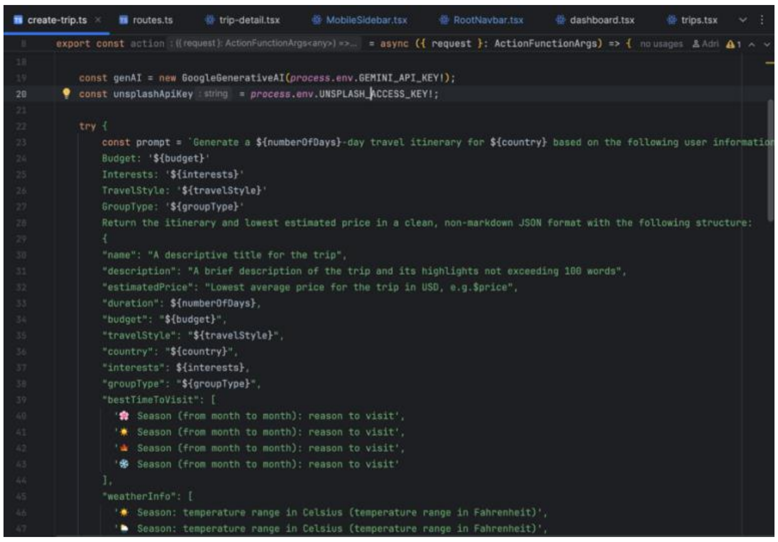


```tsx
import ...

export async function clientLoader() {
    try {
        const user = await account.get();

        if(!user.$id) return redirect('/sign-in');

        const existingUser = await getExistingUser(user.$id);
        return existingUser?.$id ? existingUser : await storeUserData();
    } catch (e) {
        console.log('Error fetching user', e)
        return redirect('/sign-in')
    }
}

const PageLayout = () => {
    return (
        <div className="bg-light-200">
            <RootNavbar />
            <Outlet />
        </div>
    )
}
export default PageLayout
```

Fig. 5.2 Code Snippet for page-layout.tsx

5.2 Backend and Serverless Function

Instead of managing a custom Express.js server, Sereinly leverages **Appwrite serverless functions** to handle backend operations such as AI trip generation and Stripe payment handling. When the admin generates a trip, the serverless function constructs a dynamic prompt and calls the Gemini API to fetch a personalized itinerary.



```ts
export const action = async ({ request }: ActionFunctionArgs) => {

    const genAI = new GoogleGenerativeAI(process.env.GEMINI_API_KEY!);
    const unsplashApiKey = process.env.UNSPLASH_ACCESS_KEY!;

    try {
        const prompt = `Generate a ${numberOfDays}-day travel itinerary for ${country} based on the following user informatio
        Budget: '${budget}'
        Interests: '${interests}'
        TravelStyle: '${travelStyle}'
        GroupType: '${groupType}'
        Return the itinerary and lowest estimated price in a clean, non-markdown JSON format with the following structure:
        {
        "name": "A descriptive title for the trip",
        "description": "A brief description of the trip and its highlights not exceeding 100 words",
        "estimatedPrice": "Lowest average price for the trip in USD, e.g.$price",
        "duration": ${numberOfDays},
        "budget": "${budget}",
        "travelStyle": "${travelStyle}",
        "country": "${country}",
        "interests": ${interests},
        "groupType": "${groupType}",
        "bestTimeToVisit": [
          '🌸 Season (from month to month): reason to visit',
          '☀️ Season (from month to month): reason to visit',
          '🍁 Season (from month to month): reason to visit',
          '❄️ Season (from month to month): reason to visit'
        ],
        "weatherInfo": [
          '☀️ Season: temperature range in Celsius (temperature range in Fahrenheit)',
          '🌥️ Season: temperature range in Celsius (temperature range in Fahrenheit)',
```

Fig. 5.3 Code Snippet create-trip.ts (backend)

Chapter 6

Testing

6.1 Testing Strategy

The testing process for Sereinly follows a systematic approach combining manual functional testing, API testing, payment gateway testing, and responsive design validation. The goal is to ensure that each module behaves as expected and that the overall user experience remains consistent across devices and platforms.

Testing was divided into multiple phases:

- Unit testing individual React components.
- Integration testing of data flow between frontend, backend (Appwrite), Gemini API, and Stripe.
- End-to-end testing for complete user and admin workflows.

6.2 Testing Types Performed

6.2.1 White Box Testing

White box testing focuses on internal logic, code structures, and data flow within the components. The following tests were performed by examining source code (primarily in React, Node.js, and Appwrite cloud functions):

Unit Testing

- **Form Validation Functions:**
  Tested individual functions that validate inputs in the "Create Trip" form (e.g., date range checks, required field validation, place auto-complete selection).
- **Trip Data Processing Logic:**
  Functions generating the AI prompt from user inputs were tested to ensure consistent formatting and edge-case handling (e.g., missing inputs, special characters).
- **API Integration Logic:**
  - Ensured the Gemini API returns structured responses.
  - Stripe payment status handlers were verified using mock responses.

Code Path Coverage

- **Conditional Rendering in React Components:**
  Verified different UI states (e.g., loading states, error messages) based on network call outcomes or user role (admin vs. user).
- **Authentication & Authorization Logic:**
  - Ensured restricted access to admin-only features (AI trip generation).
  - Tested JWT/session validity and user redirects on unauthorized access.

6.2.2 Black Box Testing

Black box testing was performed without knowledge of the internal code. The focus was on validating functional behaviour from the end user's perspective.

Functional Testing

- **Trip Generation Flow (Admin Only):**
  Input: A valid set of trip preferences.
  Output: AI-generated itinerary appears with hotel suggestions and maps.
  Expected Result: Results shown with no UI break or undefined data.
- **User Flow:**
  - **User Registration & Login:**
    Checked for successful signups, incorrect credentials, and error messages.
  - **Join Trip after Payment:**
    - Verified Stripe payment success triggers user trip join.
    - Ensured users cannot join trips without payment.
- **AI Restrictions:**
  Verified that only admin users see the "Generate AI Trip" button and others get access denied or no visibility.

UI/UX Testing

- **Responsive Design:**
  Tested layouts on multiple screen sizes (mobile, tablet, desktop) to ensure layout and readability.
- **Animations (AOS):**

Verified that animations render smoothly on scroll and do not block content or cause jank.

- **Navigation & Routing:**
  - Tested direct URL entry for protected routes.
  - Ensured proper redirection to login or error pages when needed.

Error Handling Tests

- **API Failure Simulation:**
  Simulated a failed call to the Gemini API and Stripe, ensuring proper user alerts and fallback UI.
- **Empty State Tests:**
  - No upcoming trips → "No trips found" message is displayed.
  - No AI itinerary generated → default placeholder shown.

Chapter 7

Results and Discussion

7.1Results

The development and internal testing of **Sereinly** demonstrate that the platform is not only technically robust but also functionally efficient in delivering intelligent travel planning features. The system successfully meets its core objective of allowing administrators to create AI-generated trips while enabling users to browse and join trips after secure payment.

During testing, the **Gemini API** consistently generated detailed, contextually accurate itineraries based on different user profiles and prompt formats. It responded to a wide variety of scenarios, including solo getaways, romantic trips, family vacations, and group adventures. The quality and relevance of the AI-generated plans were satisfactory and aligned with the inputs provided by the admin.

The **Appwrite backend** handled all user authentication, trip data storage, and access control without failure. Role-based authorization worked effectively — admins could create and manage trips, while users could view and join trips but had no trip creation privileges. The platform also supported smooth database operations, including document creation, updates, and permission handling.

The **payment workflow** using **Stripe Checkout** was successfully implemented and tested. Transactions were securely processed, and user trip enrolment was updated in real-time after webhook confirmation. The end-to-end trip joining flow—from browsing trips to accessing full itineraries post-payment—was completed in under 15 seconds on average.

The **Admin Dashboard**, powered by Syncfusion components, visually represented important platform metrics such as total users, trips created, active enrolments, and the number of AI-generated itineraries. Admins could also view a complete list of users and generate new trips through a dedicated interface, streamlining the content creation process significantly.

Overall, the system remained responsive, with page loads averaging under 2 seconds and AI trip generation completing in approximately 3–5 seconds depending on input length and API load.

7.2 Discussion

The results indicate that Sereinly effectively addresses the core pain points of traditional trip planning platforms. Unlike conventional tools that require users to manually plan or rely on static suggestions, Sereinly delivers dynamically generated itineraries tailored to group size, interests, and budget. The platform is optimized for various user types including solo travellers, couples, families, and friend groups.

The decision to restrict trip creation to the admin role proved beneficial in maintaining quality and consistency. This approach minimizes the chance of irrelevant or poor-quality plans being exposed to users, making the overall platform experience more reliable and curated.

Integrating Stripe for payments added a critical layer of professionalism and security. It allowed the platform to simulate a SaaS-like structure, ensuring that users only access complete itineraries after verified transactions. The ability to scale this payment logic for future premium features or user-tiered access adds flexibility to the business model.

The use of Appwrite instead of a traditional Node.js and MongoDB backend reduced development overhead and improved maintainability. With built-in features such as authentication, serverless functions, and role-based access control, Appwrite significantly accelerated the development process.

Despite its strengths, the platform still has areas for future improvement. Currently, user feedback loops and rating systems for generated itineraries are not in place. Additionally, while the AI performs well in general scenarios, there is potential to fine-tune the prompts further or integrate additional APIs (e.g., for live hotel pricing or event calendars) to increase itinerary richness.

In conclusion, the results confirm that Sereinly is a well-rounded, AI-powered travel assistant that meets real-world needs with a clean, modern design and scalable architecture. It lays a solid foundation for further innovation in intelligent travel planning

Chapter 8

Conclusion and Future Work

8.1 Conclusion

The project **Sereinly** successfully demonstrates how artificial intelligence and modern web development can be combined to revolutionize the way people plan their travels. By automating itinerary generation through the **Gemini Generative AI API** and organizing system logic using a **React + Appwrite** stack, the platform addresses key shortcomings of traditional travel tools.

Unlike existing systems that require users to build itineraries manually or rely on static templates, Sereinly introduces a streamlined flow where **only administrators** generate trips using AI, ensuring consistent quality. Users can explore these curated itineraries and securely **join them through Stripe-powered payments**, receiving full access upon successful enrolment.

The **Admin Dashboard** adds further value by offering real-time statistics about user engagement and AI activity, enabling efficient content and platform management. With a responsive frontend, secure backend services, and scalable architecture, Sereinly proves to be an innovative solution that caters to diverse traveller types—solo, couple, family, and friends.

The platform was developed using the **Agile SDLC**, enabling continuous iteration and refinement throughout its lifecycle. Functional, integration, and UI testing confirmed the system's reliability and responsiveness across key components such as AI response handling, user payments, and data consistency.

Overall, Sereinly represents a meaningful advancement in digital travel planning by offering intelligence, personalization, and seamless booking integration under a single system.

8.2 Future Work

While the current system meets its core objectives, several enhancements can further improve functionality, scalability, and user satisfaction:

1. **User Feedback and Ratings:** A feedback system can be added where users can rate trips they join. This will help refine future AI prompt generation and improve content quality.
2. **Trip Sharing and Collaboration:** Features allowing users to invite others to trips or share itineraries on social platforms could increase engagement and outreach.
3. **Trip PDF Export:** Enabling users to download their itineraries as printable PDFs would enhance usability for offline travellers.
4. **Admin AI Prompt Tuning:** A visual AI prompt builder with mood presets (e.g., relaxing, adventure, family-friendly) could make itinerary creation faster and more precise.
5. **User Role Tiers and Subscriptions:** Introducing premium features such as early trip access, trip customization, or saved itineraries could support a sustainable monetization model.
6. **Advanced Analytics:** Future versions of the Admin Dashboard can include visual analytics for user retention, payment trends, and trip popularity to help make data-driven decisions.
7. **Multilingual Support:** Adding support for different languages would make Sereinly accessible to a broader global audience.

Through these improvements, Sereinly can evolve from a powerful academic prototype into a commercially viable product capable of serving a global travel community with intelligence, speed, and personalization.

References

[1] Veluru, Chandra. (2023). Transforming Travel Planning: The Impact of Generative AI on Itinerary Optimization, Cost Efficiency and User Experience. Journal of Artificial Intelligence & Cloud Computing. 2. 1-8. 10.47363/JAICC/2023(2)350.

[2] A. Chen et al., "TravelAgent: An AI Assistant for Personalized Travel Planning," *arXiv preprint arXiv:2409.08069*, 2024.

[3] A. Kumar and B. Hemaraj, "A Review Paper On AI-Driven Travel Planning," *ResearchGate*, 2024.

[4] B. Barua and M. S. Kaiser, "Optimizing Travel Itineraries with AI Algorithms in a Microservices Architecture: Balancing Cost, Time, Preferences, and Sustainability," *arXiv preprint arXiv:2410.17943*, 2024.

[5] T. de la Rosa et al., "TRIP-PAL: Travel Planning with Guarantees by Combining Large Language Models and Automated Planners," *arXiv preprint arXiv:2406.10196*, 2024.

[6] D. Ju et al., "To the Globe (TTG): Towards Language-Driven Guaranteed Travel Planning," *arXiv preprint arXiv:2410.16456*, 2024

[7] B. Li, "EverywhereGPT: An AI Travel Planning Assistant Based on ChatGPT," in *Proceedings of the 4th International Conference on Artificial Intelligence and Computer Engineering*, 2024, pp. 995-1003.

[8] Mohith S, Sai & H, HemaMalini & .K, Karthik & N, Nithin & . R, Sanath. (2025). A Review Paper On AI-Driven Travel Planning. 10.5281/zenodo.14591573.

[9] Komal Londhe, Nikita Dharmadhikari, Parth Zaveri, Unal Sakoglu, "Enhanced Travel Experience using Artificial Intelligence: A Data-driven Approach", Procedia Computer Science,