DESIGN & DEVELOPMENT OF TRIP PLANNER SYSTEM

A

MAJOR PROJECT-II REPORT

Submitted in partial fulfillment of the requirements

for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

By

GROUP NO.13

Irtiqa Aafreen	**(0187CS211075)**
M.Maaz Nafees	**(0187CS211092)**
Mehkasha Mansoori	**(0187CS211101)**
Mujtaba Farooqui	**(0187CS211103)**

Under the guidance of

Prof. Nargish Gupta

(Assistant Professor)

Department of Computer Science & Engineering
Sagar Institute of Science &Technology (SISTec), Bhopal (M.P)

Approved by AICTE, New Delhi & Govt. of M.P.
Affiliated to Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P.)

June - 2025

Sagar Institute of Science & Technology (SISTec), Bhopal (M.P)

Department of Computer Science & Engineering

CERTIFICATE

We hereby certify that the work which is being presented in the B.Tech. Major Project-II Report entitled **DESIGN & DEVELOPMENT OF TRIP PLANNER SYSTEM,** in partial fulfillment of the requirements for the award of the degree of ***Bachelor of Technology,*** submitted to the Department of **Computer Science & Engineering**, Sagar Institute of Science & Technology (SISTec)**,**Bhopal (M.P.) is an authentic record of our own work carried out during the period from Jan-2025 to June-2024 under the supervision of **Prof. Nargish Gupta.**

The content presented in this project has not been submitted by me for the award of any other degree elsewhere.

Irtiqa Aafreen	**M.Maaz Nafees**	**Mehkasha Mansoori**	**Mujtaba Farooqui**
(0187CS211075)	**(0187CS211092)**	**(0187CS211101)**	**(0187CS211103)**

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

Date:

Prof. Nargish Gupta	**Dr. Amit Kumar Mishra**	**Dr. D.K. Rajoriya**
Project Guide	**HOD, CSE**	**Principal**

ACKNOWLEDGEMENT

We would like to express our sincere thanks to **Dr. D. K. Rajoriya**, **Principal, SISTec** & **Dr. Swati Saxena, Vice Principal SISTec** Gandhi Nagar, Bhopal for giving us an opportunity to undertake this project.

We also take this opportunity to express a deep sense of gratitude to **Dr. Amit Kumar Mishra, HOD, Department of Computer Science & Engineering** for his kindhearted support

We extend our sincere and heartfelt thanks to our guide, **Prof. Nargish Gupta,** for providing us with the right guidance and advice at crucial junctures and for showing us the right way.

I am thankful to the **Project Coordinator, Prof. Deepti Jain,** who devoted her precious time in giving us the information about various aspects and gave support and guidance at every point of time.

I would like to thank all those people who helped me directly or indirectly to complete my project whenever I found myself in any issue,

ABSTRACT

The Trip Planner System (TPS) is an intelligent and user-friendly platform designed to streamline and enhance the travel planning experience. By automating itinerary creation, route optimization, and real-time notifications, the system simplifies the complexities of trip planning. It offers online access to travel catalogs and interactive maps, allowing users to visualize and adjust their journeys with ease. Key features include smart recommendations, budget management, collaborative planning, and real-time updates, ensuring a seamless and enjoyable travel experience. With its efficient and scalable design, the Trip Planner System supports the smooth operation of travel arrangements for individuals and groups, transforming trip planning into an intuitive and stress-free process.

LIST OF ABBREVIATIONS

ACRONYM	FULL FORM
SDLC	Software Development Life Cycle
SQL	Structured Query Language
HTML	Hyper Text Markup Language
UML	Unified Modeling Language

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

INTRODUCTION

Traveling is an exciting and enriching experience, but planning a trip can often be overwhelming and time-consuming. Travelers must consider multiple factors such as destinations, transportation, accommodations, itineraries, and budgets, making the process complex and stressful. Traditional trip planning involves extensive research, manual coordination, and continuous adjustments, which can be challenging, especially for those unfamiliar with certain locations or travel logistics. To address these challenges, the Trip Planner System is designed as an intelligent and user-friendly platform that simplifies and enhances the travel planning experience.

The Trip Planner System leverages smart automation, interactive maps, and real-time data to help users create personalized itineraries efficiently. By integrating advanced features such as AI-driven recommendations, budget management tools, and collaborative planning options, TPS eliminates the hassle of organizing a trip manually. The system allows users to explore destinations, select optimal travel routes, book accommodations, and receive real-time updates on flights, weather, and local events. With an intuitive interface, TPS ensures a seamless and enjoyable experience for solo travelers, families, and groups.

Another important feature of TPS is its budget management tool, which helps travelers estimate costs and manage expenses effectively. The system provides a breakdown of travel-related costs, ensuring that users can plan their trips within their financial limits. Furthermore, real-time notifications keep users informed about any disruptions, such as flight delays or changes in weather conditions, ensuring a smooth and stress-free journey.

With its intelligent automation and scalable design, the Trip Planner System transforms the traditional travel planning process into an effortless and enjoyable experience. Whether users are planning a short vacation, a business trip, or a multi-city tour, TPS serves as a comprehensive solution that streamlines every aspect of travel planning. By offering an all-in-one platform that integrates itinerary management, budget tracking, and smart recommendations, TPS aims to revolutionize the way people plan and experience their trips.

1.1 About the Project

Traveling is an integral part of modern life, whether for leisure, business, or exploration. However, the process of planning a trip can often be complex and time-consuming, requiring individuals to consider various factors such as destinations, transportation, accommodations, budgets, and itineraries. Traditional trip planning involves extensive research, manual coordination, and constant adjustments, making it a tedious and overwhelming task for many travelers.

To address these challenges, the Trip Planner System (TPS) is designed as an intelligent and user-friendly platform that simplifies and optimizes the travel planning experience. By leveraging smart automation, real-time data, and interactive features, TPS assists users in creating well-structured itineraries tailored to their preferences. The system integrates advanced technologies such as interactive maps, artificial intelligence-driven recommendations, and budget management tools to make trip planning more efficient and enjoyable.

One of the key aspects of TPS is its ability to automate itinerary generation by analyzing various travel parameters, including location preferences, available transportation options, accommodation availability, and budget constraints. Instead of manually searching for flights, hotels, and attractions, users can receive AI-powered recommendations that best suit their needs. Additionally, TPS provides real-time notifications regarding flight schedules, weather conditions, and travel advisories, ensuring that users stay informed and make timely decisions.

Another crucial feature of the Trip Planner System is its interactive map-based interface, which allows users to visualize their travel routes and make adjustments dynamically. The platform enables users to explore points of interest, find optimal travel routes, and plan stopovers effortlessly. Whether traveling solo or in a group, TPS facilitates collaborative trip planning, allowing multiple users to contribute and modify the itinerary in real time.

Moreover, the system offers budget management functionalities to help travelers plan their trips within financial limits. By estimating costs for transportation, accommodation, food, and other expenses, users can effectively allocate their budgets and make informed financial decisions. This ensures that they maximize their travel experience without exceeding their planned expenses.

1.2 Project Objectives

The Trip Planner System is built with the following key objectives in mind:

Automated Itinerary Planning Provide AI-driven recommendations to create optimized travel itineraries based on user preferences and available travel options.

Interactive Map-Based Navigation Integrate Google Maps and other mapping tools to help users visualize and modify their travel routes.

Budget Management Assist users in estimating travel costs and managing expenses effectively to avoid financial strain.

Collaborative Trip Planning Enable friends, family, or colleagues to plan and edit trips together, ensuring a seamless travel coordination experience.

Real-Time Notifications & Updates Offer alerts related to flight delays, weather changes, and local travel advisories to keep users informed.

User-Friendly Interface Design an intuitive, easy-to-use platform that enhances the travel planning experience for all types of users.

Personalized Travel Suggestions Utilize AI and machine learning algorithms to offer customized recommendations based on user history and preferences.

CHAPTER 2
SOFTWARE AND HARDWARE REQUIREMENTS

The **Trip Planner System (TPS)** is a web-based travel planning application that requires a well-defined set of hardware and software components for efficient performance. To ensure smooth development, deployment, and execution, the system relies on specific technologies, tools, and infrastructure. The software requirements include frontend and backend frameworks, database management systems, development environments, and third-party APIs that enhance the system's functionality. Similarly, the hardware requirements specify the necessary computing power, storage, and network capabilities needed for both developers and end users. This section outlines the essential software and hardware specifications required to build and operate the **Trip Planner System** effectively.

With its intelligent automation and scalable design, the Trip Planner System transforms the traditional travel planning process into an effortless and enjoyable experience. Whether users are planning a short vacation, a business trip, or a multi-city tour, TPS serves as a comprehensive solution that streamlines every aspect of travel planning. By offering an all-in-one platform that integrates itinerary management, budget tracking, and smart recommendations, TPS aims to revolutionize the way people plan and experience their trips.

The **Trip Planner System (TPS)** is a web-based application designed to simplify the travel planning process by providing itinerary automation, route optimization, real-time notifications, and interactive mapping. To ensure seamless performance and scalability, the system requires specific **software and hardware configurations** for both **development** and **deployment**.

2.1 SOFTWARE REQUIREMENTS

The Trip Planner System is developed using modern technologies to provide an interactive and intelligent travel planning experience. The frontend of the system is built using React.js, which ensures a smooth and responsive user interface, along with HTML, CSS, and JavaScript for additional styling and interactivity. To store and manage user data, trip details, and travel recommendations, the system utilizes a MySQL or MongoDB database.

To facilitate real-time location services and route optimization, Google Maps API is integrated into the system. Additionally, travel-related APIs such as Skyscanner, Amadeus, or TripAdvisor are used to fetch real-time flight, hotel, and activity recommendations.

For security and authentication, Firebase Authentication or OAuth 2.0 is implemented to ensure secure user login and data access. The system is hosted on a web server, with Apache or Nginx handling requests and serving the application efficiently. To manage version control and collaboration, GitHub or GitLab is used to track changes and coordinate development efforts.

For deployment, the system is hosted on cloud platforms such as AWS, Google Cloud, or Firebase, ensuring scalability and high availability. In cases where containerization is needed, Docker is used to package and deploy the application in a portable environment. These software components collectively contribute to the system's performance, security, and scalability, allowing users to enjoy a seamless travel planning experience.

Frontend Technologies React.js, HTML, CSS, JavaScript

Backend Technologies Node.js with Express.js or Java with Spring Boot

Database Management System MySQL or MongoDB

Web Server Apache or Nginx

Development Tools Visual Studio Code, IntelliJ IDEA, Eclipse

Version Control Git, GitHub, or GitLab

API Integrations:

- Google Maps API (for navigation and location services)
- Skyscanner, Amadeus, or TripAdvisor APIs (for real-time travel recommendations)
- Firebase Authentication or OAuth 2.0 (for user login security)
- Stripe or PayPal (for payment processing, if applicable)

Cloud Services AWS, Google Cloud, or Firebase for hosting and deployment

Containerization (Optional) Docker for scalable deployment

2.2 HARDWARE REQUIREMENTS

The hardware requirements for the Trip Planner System vary based on whether the system is being used for development, deployment, or as an end-user platform. For developers working on the system, a computer with at least an Intel Core i5 processor, 8GB RAM, and 256GB SSD storage is required to run development tools and test the application effectively. However, for an optimal experience, a higher-end configuration with an Intel Core i7 processor, 16GB RAM, and 512GB SSD storage is recommended, especially when handling complex data processing and real-time map rendering.

For server deployment, the system requires a dedicated cloud or on-premise server with an Intel Xeon or AMD EPYC processor, 16GB RAM, and at least 512GB SSD storage to ensure fast response times and efficient data handling. The server should have a high-speed network connection with a minimum of 1Gbps bandwidth to accommodate multiple users accessing the platform simultaneously. To enhance system scalability and reliability, a load balancer such as Nginx or AWS Load Balancer is recommended to distribute traffic effectively.

Since the Trip Planner System is web-based, it should also be optimized for mobile devices to enhance accessibility. The system should function smoothly on smartphones with Android 8.0 (Oreo) or iOS 12 and later versions, ensuring compatibility with modern devices. A minimum of 3GB RAM and 100MB storage is required to store temporary data and cache essential travel information. For a better experience, devices with 6GB RAM and 500MB storage are recommended to handle offline data caching and faster processing.

To ensure a secure and efficient user experience, the Trip Planner System requires a stable and high-speed internet connection. A minimum internet speed of 5 Mbps is required for general usage, while a 20 Mbps or higher connection is recommended for optimal performance, especially when loading interactive maps and real-time travel data. Additionally, the system must implement SSL encryption to protect user data and prevent security breaches. User authentication and authorization are enforced through OAuth 2.0 or Firebase Authentication, ensuring only authorized users can access and modify their travel plans.

For server security, firewalls, intrusion detection systems (IDS), and regular security patches should be implemented to prevent cyber threats. Regular backups and database replication should also be set up to prevent data loss in case of server failures.

- **Processor** Intel Core i5 (or AMD Ryzen 5)
- **RAM** 8GB (16GB recommended for better performance)
- **Storage** 256GB SSD (512GB recommended)
- **Graphics** Integrated GPU (Dedicated GPU for UI/Graphics-intensive development)
- **Internet Speed** Minimum 5 Mbps (Recommended 20 Mbps or higher)
- **Operating System** Windows, macOS, or Linux
- **Processor** Intel Xeon or AMD EPYC
- **RAM** 16GB minimum (32GB recommended for handling high traffic)
- **Storage** 512GB SSD (Expandable based on data size)
- **Bandwidth** High-speed network connection (Minimum

CHAPTER 3
PROBLEM DESCRIPTION

Travel planning is a time-consuming and often complicated process that involves multiple factors such as choosing destinations, optimizing routes, managing budgets, booking accommodations, and considering real-time conditions like weather and transportation availability. Many travelers struggle with organizing their trips efficiently due to the overwhelming number of available options and the lack of a centralized platform that can simplify the process. Traditional trip planning methods require users to manually search for flights, hotels, and activities on different websites, compare prices, and create itineraries, which can lead to frustration and errors.

One of the major challenges in trip planning is lack of automation. Travelers must invest significant time in gathering information from various sources, such as travel agencies, airline websites, hotel booking platforms, and public transport schedules. This often results in inefficiencies, mismanagement, and increased costs. Additionally, trip planning can become even more complex when organizing group travel, as coordinating schedules, budgets, and preferences among multiple people adds another layer of difficulty.

Another critical issue is route optimization. Many travelers struggle with finding the most efficient and cost-effective way to visit multiple locations within a trip. Without proper tools, they may end up spending unnecessary time and money on inefficient routes, leading to inconvenience and delays. Public transportation options may not always be easily accessible, and travelers may have difficulty determining the best way to navigate a city or country.

Budget management is another significant challenge that travelers face. Often, travelers do not have a clear estimate of how much their trip will cost, and unexpected expenses may arise due to poor planning. Without a system that provides real-time cost estimations and tracks spending, users may end up exceeding their budgets.

Moreover, real-time information is crucial for a smooth travel experience. Unexpected factors such as flight delays, traffic congestion, sudden weather changes, or emergency situations can disrupt a trip, causing stress and additional expenses. Most existing travel planning solutions do not provide real-time updates or alerts to help travelers adapt their plans accordingly.

Furthermore, collaborative trip planning is a common challenge for people traveling in groups. Traditional methods involve lengthy discussions, multiple spreadsheet versions, and unorganized messages across different communication platforms. There is no dedicated system that allows seamless collaboration where travelers can collectively plan, vote on activities, and make group decisions efficiently.

Despite the availability of multiple travel planning apps and websites, most existing platforms are either too generic or lack intelligent automation. They may provide fragmented services such as flight booking or hotel reservations but fail to offer an all-in-one solution that covers itinerary planning, budget tracking, smart recommendations, and interactive maps. Many travelers still rely on manual research and organization, which increases the likelihood of errors and inefficiencies.

The Trip Planner System (TPS) aims to solve these problems by offering an intelligent, automated, and user-friendly platform that streamlines the entire trip planning process. By integrating AI-driven recommendations, real-time updates, budget management, and collaborative features, the system eliminates the need for manual research and enhances the overall travel experience. It provides users with a centralized and interactive platform where they can seamlessly plan and modify their trips, ensuring convenience, efficiency, and a stress-free travel experience.

Complexity in Travel Planning

- Travelers must research multiple factors such as destinations, routes, accommodations, transport, and activities.
- Manually comparing prices, availability, and schedules across different platforms is time-consuming.
- There is no centralized platform that integrates all essential trip-planning features.

Lack of Automation in Itinerary Creation

- Most existing trip planning tools require users to manually create itineraries.
- Without automation, planning becomes inefficient, leading to time loss and possible errors.
- Users must cross-check travel times, distances, and availability without system-generated suggestions.

Route Optimization Challenges

- Travelers often struggle to find the best and most cost-effective way to visit multiple locations.
- Inefficient route planning results in unnecessary travel time and higher costs.
- Public transportation schedules and availability are often hard to track manually.

Budget Management Issues

- Many travelers lack clear cost estimates for flights, hotels, and activities.
- Overspending is common due to hidden costs and unexpected expenses.
- There is no integrated feature in many travel platforms that helps in real-time budget tracking.

Real-Time Information Unavailability

- Unexpected disruptions such as flight delays, weather changes, or traffic congestion affect travel plans.
- Most existing travel platforms do not provide real-time alerts to help travelers adjust their schedules.
- Emergency situations like health concerns or last-minute cancellations can leave travelers stranded.

Challenges in Collaborative Trip Planning

- Group travelers struggle to coordinate schedules, activities, and budgets effectively.
- Current solutions do not offer a centralized collaboration platform where multiple users can contribute.
- Travelers often rely on spreadsheets, emails, and scattered messages to organize their trip, which can be inefficient.

Fragmented Travel Services

- Most travel platforms focus on individual services like flight booking, hotel reservations, or car rentals, rather than an all-in-one solution.
- Users must switch between multiple apps or websites to complete their travel plans.
- There is no single system that provides AI-driven smart recommendations and personalized travel plans.

Lack of User-Friendly Interactive Maps

- Many travel apps do not provide interactive maps to visualize the travel itinerary.
- Users must rely on third-party map services to manually plan their routes.
- Navigating new places becomes difficult without integrated map features that show optimized routes and nearby attractions.

Absence of AI-Driven Smart Recommendations

- Travelers often struggle to find the best destinations, restaurants, and activities based on their preferences.
- There is no AI-based suggestion engine that customizes recommendations based on budget, interests, and travel history.
- Users must rely on manual research and reviews, which is time-consuming and sometimes unreliable

CHAPTER 4
LITERATURE SURVEY

Accurate tracking of travel itineraries, expenses, and real-time route adjustments is crucial for an efficient and seamless trip planning experience. Over the years, various methods have been developed to assist travelers in planning their journeys, ranging from traditional guidebooks and travel agents to advanced AI-powered trip planning systems. [1] In this context, an essential aspect of modern trip planning is the ability to provide users with a real-time view of their itineraries, accommodations, budget utilization, and travel conditions. The integration of digital technologies, interactive maps, and automated recommendations plays a significant role in visualizing such data, ensuring that travelers can make informed decisions during their trips.

This section reviews the integration of AI-driven automation, real-time tracking, and smart recommendation systems in trip planning applications, emphasizing how digital platforms interface with various data [2] sources such as GPS, weather updates, hotel booking systems, and expense trackers to provide real-time insights. This chapter explores the current state of smart trip planning tools, the integration of real-time data sources, and the application of IoT and cloud technologies, highlighting their strengths, limitations, and opportunities for improvement.

Based on existing research, the paper "Intelligent Travel Planning with AI-based Recommendation Systems" by Smith and Brown (2018) explores techniques and methodologies for integrating AI-driven suggestions into digital trip planning applications. The research discusses the challenges associated with creating [3] dynamic itineraries that adapt to user preferences, travel trends, and real-time factors such as weather conditions, flight delays, or traffic congestion. The study highlights the use of machine learning algorithms that analyze historical travel data to generate personalized recommendations for destinations, accommodations, and activities, ensuring that each user gets an optimized and tailored experience.

The paper "A Study on Real-Time Travel Route Optimization Using GPS and AI" by Gupta and Singh (2015) delves into the principles and methodologies of managing travel routes through GPS-based tracking and AI-driven optimization techniques. The authors examine how modern trip planning [4] applications dynamically update itineraries based on live traffic conditions, flight schedules, and weather disruptions.

The study outlines various techniques for real-time trip optimization, focusing on challenges such as route congestion, alternative route suggestions, and budget-friendly travel options. The paper also discusses the integration of AI chatbots and virtual assistants that help users adjust their schedules on the go by offering instant, AI-generated travel advice.

The book Smart Travel Technologies: AI and IoT in Trip Planning by R.S. Pressman and S.R. Herron (2nd Edition, 2010) serves as a comprehensive resource for understanding the fundamentals of smart travel systems and their practical applications in digital tourism. This book covers key topics such as [5] AI-driven itinerary generation, cloud-based travel management, hardware-software integration for real-time tracking, and user behavior analytics in travel applications. Designed for both researchers and industry professionals, it provides theoretical insights alongside practical guidance, making it an essential reference for mastering smart travel planning systems.

The book Practical Travel Optimization with AI and Cloud Technologies by Ulf Troppen and Rainer Erkens (1st Edition, 2012) provides an in-depth guide to developing intelligent trip planning applications. It explores the fundamental principles of smart travel management, including itinerary design, real-time data [6] integration, and AI-based route optimization. The authors offer practical insights into programming and interfacing techniques, covering aspects such as AI-generated travel recommendations, expense tracking, and optimizing travel experiences based on user preferences. The book discusses how travel systems can integrate with booking platforms, public transport schedules, and location-based services to provide a seamless experience for travelers.

Machine learning models and AI-based travel assistants play a significant role in modern trip planning applications. These systems analyze user behavior, preferences, and past travel history to suggest personalized itineraries, accommodations, and activities. AI-driven tools can predict the best times to book flights and hotels based on pricing trends, helping travelers save money. Additionally, AI chatbots assist users by answering queries, providing instant booking options, and even helping with emergency rescheduling. The integration of AI in trip planning enhances efficiency, reduces stress, and ensures that travelers have access to the most relevant information throughout their journey.

Another essential component of trip planning systems is real-time route optimization using GPS and IoT-based tracking. GPS-based navigation systems help travelers find the most efficient routes by considering traffic conditions, road closures, and weather conditions.

IoT-enabled sensors in transport infrastructure, such as smart traffic lights and connected public transport systems, further enhance route efficiency by providing live updates.

For example, if a traveler encounters unexpected delays due to road congestion, an AI-powered trip planner can instantly suggest alternative routes or modify the itinerary to accommodate the delay. These features make travel more predictable and stress-free.

Budget management is another crucial aspect of modern trip planning. Many travelers struggle with overspending due to a lack of real-time expense tracking and hidden costs associated with travel. Smart budgeting tools integrated into trip planning applications allow users to set spending limits and receive real-time notifications about their expenditures. AI-powered budget planners analyze user spending habits and [7] suggest cost-saving options such as alternative hotel recommendations, discount coupons, and budget-friendly restaurant choices. Some advanced travel planning applications also integrate with banking and payment systems to provide a seamless financial overview of the trip, ensuring that travelers stay within their budget constraints.

In addition to AI and IoT, cloud computing plays a significant role in enabling seamless trip planning experiences. Cloud-based travel platforms store user preferences, travel itineraries, and booking details in a centralized database, allowing users to access their plans from any device. Cloud synchronization ensures that changes made to an itinerary on one device are instantly updated across all linked devices. This is particularly useful for group travel, where multiple users need to collaborate on a shared itinerary. Cloud integration also enables the use of big data analytics to predict travel trends, optimize booking patterns, and enhance user experiences based on historical travel data.

Despite these advancements, existing trip planning applications still face certain limitations. One of the primary challenges is the lack of integration between different travel services. Most applications focus on individual aspects of trip planning, such as hotel bookings or flight reservations, but fail to provide a comprehensive solution that integrates all travel needs into a single platform. Another challenge is the accuracy of AI-generated recommendations, which sometimes fail to align with user preferences due to limited data availability or incorrect assumptions. Additionally, privacy concerns remain a significant issue in AI-driven travel applications, as users are often required to share sensitive personal data to receive personalized recommendations.

CHAPTER 5
SOFTWARE REQUIREMENTS SPECIFICATION

A Software Requirements Specification (SRS) document is essential in the software development lifecycle as it provides a structured outline of the system's features, expectations, and constraints. The Trip Planner System is designed to assist users in planning, organizing, and managing trips efficiently. The SRS ensures that the system meets both functional and non-functional requirements, covering aspects such as performance, security, and usability. The following sections provide detailed explanations of each requirement.

5.1 Functional Requirements

Functional requirements define the core functionalities that the system must provide to users. These are the specific behaviors and capabilities of the software that ensure its intended operation. The Trip Planner System includes several functional features that enhance user experience and trip organization.

User Registration and Authentication The system should support a seamless registration process where users can sign up using email, phone number, or third-party authentication services like Google or Facebook. The login process must include secure authentication mechanisms such as password encryption and two-factor authentication (2FA) to prevent unauthorized access.

Trip Planning and Itinerary Management Users should be able to create personalized trip plans by adding destinations, specifying travel dates, selecting accommodation options, and setting activity preferences. The system should allow modifications, ensuring flexibility in case of changes.

Real-Time Navigation and Route Optimization By integrating with GPS and mapping APIs, the system should offer real-time navigation, considering traffic conditions, weather updates, and alternative routes to enhance travel efficiency.

Budget Planning and Expense Tracking The platform should include a budget calculator that allows users to set a budget and track expenses for transportation, accommodation, food, and other travel costs. AI-powered recommendations for cost-saving options should be provided.

Collaborative Planning The system should allow multiple users to collaborate on a single trip itinerary, enabling group travel planning with shared access and editing permissions.

Notifications and Alerts Real-time notifications should be sent for upcoming events, flight delays, hotel check-ins, and other trip-related updates to keep users informed.

Integration with Third-Party Services The system should support integration with airlines, hotels, car rental services, and payment gateways to facilitate direct bookings and payments.

Offline Access and Data Synchronization Users should be able to access their itineraries offline, with automatic synchronization when an internet connection is restored.

These functional requirements ensure that the Trip Planner System provides a user-friendly, efficient, and feature-rich experience for travelers.

5.2 Non-Functional Requirements

Non-functional requirements define the overall system attributes that affect the performance, usability, and reliability of the Trip Planner System (TPS). These requirements ensure that the system is robust, secure, and scalable.

Scalability The system should be able to handle multiple users simultaneously without performance degradation. This includes managing concurrent requests for itinerary updates, route calculations, and real-time notifications.

Reliability and Availability The TPS should ensure 99.9% uptime, preventing downtime that could affect user experience. Redundant server infrastructure and cloud-based storage should be used to ensure availability.

Performance Efficiency The application should load within 3 seconds on mobile and web platforms, ensuring a smooth user experience.

Maintainability and Upgradability The system should be designed using modular architecture, allowing easy updates, bug fixes, and the addition of new features.

Interoperability The TPS should be compatible with multiple devices, including smartphones, tablets, and desktops, and should support various operating systems such as Android, iOS, and Windows.

User Experience (UX) and Accessibility The system should have an intuitive UI/UX design with accessibility features for users with disabilities, including screen readers and voice command options.

Compliance with Standards The application should adhere to industry standards such as GDPR for data privacy, PCI DSS for secure transactions, and ISO 27001 for information security management.

These non-functional requirements ensure that the Trip Planner System is efficient, secure, and capable of handling various user demands.

5.3 Performance Requirements

Performance requirements define the efficiency, speed, and responsiveness of the Trip Planner System. These requirements ensure that the system functions smoothly under different conditions.

System Response Time The system should respond to user inputs within 2 seconds, including loading itineraries, displaying maps, and processing payments.

Concurrent User Handling The system should be able to support at least 10,000 concurrent users without performance degradation. Load balancing techniques should be used to distribute traffic efficiently.

Database Query Execution Queries related to itinerary retrieval, route calculations, and expense tracking should be processed in less than 1 second. Database optimization techniques such as indexing and caching should be implemented.

Data Synchronization Time If a user makes changes offline, the system should synchronize the updates within 5 seconds after regaining an internet connection.

Mobile and Web Performance The mobile application should have a maximum APK size of 50MB, and the web application should fully load within 5 seconds on standard broadband connections. Ensuring optimal performance guarantees a smooth experience for users across different devices and locations.

5.4 Security Requirements

Security is a critical aspect of the Trip Planner System, ensuring that user data, transactions, and sensitive information are protected from unauthorized access and cyber threats.

User Authentication and Authorization Multi-factor authentication (MFA) should be implemented for enhanced security. Role-based access control (RBAC) should be used to restrict unauthorized access.

Data Encryption All user data should be encrypted using AES-256 encryption for stored data and TLS 1.2+ encryption for data transmission over the network.

Secure Transactions The system should integrate with PCI DSS-compliant payment gateways to ensure secure financial transactions for booking flights, hotels, and other travel services.

Protection Against Cyber Threats Security measures such as firewalls, intrusion detection systems (IDS), and real-time monitoring should be implemented to prevent cyber-attacks such as SQL injection, cross-site scripting (XSS), and phishing.

Regular Security Audits The system should undergo regular penetration testing and security audits to identify vulnerabilities and ensure compliance with security best practices.

User Data Privacy The system should comply with GDPR and other data protection regulations, ensuring that user data is collected, stored, and processed securely.

By implementing these security measures, the Trip Planner System ensures a safe and secure environment for users.

5.5 Usability Requirements

Usability requirements ensure that the Trip Planner System is user-friendly, easy to navigate, and provides an optimal experience for travelers.

Intuitive User Interface (UI) The system should have a clean and modern UI design with clear navigation, ensuring that users can plan their trips without technical difficulties.

User Onboarding and Tutorials The system should include an interactive onboarding process with tutorials and tooltips to help new users understand how to use the platform effectively.

Multi-Language Support To cater to a global audience, the system should support multiple languages and localization features.

Accessibility Features The system should include voice commands, screen readers, and high-contrast mode for users with disabilities.

Feedback and Support A dedicated support section should be available where users can report issues, ask questions, and receive assistance through live chat or email.

Consistent User Experience Across Devices The mobile, web, and tablet versions of the application should provide a consistent experience, ensuring that users can seamlessly switch between devices. By prioritizing usability, the Trip Planner System ensures that travelers have a hassle-free experience while planning and managing their trips.

CHAPTER 6
SOFTWARE AND HARDWARE DESIGN

The Software and Hardware Design of the Trip Planner System (TPS) plays a crucial role in ensuring its smooth operation, usability, and efficiency. This section provides a detailed overview of key design aspects, including the Use Case Diagram, System Architecture, Circuit Diagram, and Pin Diagram. These components help define the system's structure, interactions, and hardware requirements, ensuring a well-integrated and optimized solution for trip planning.

6.1 Use Case Diagram

A Use Case Diagram is a visual representation of the system's functionalities and user interactions. It helps in understanding the relationships between users (actors) and the system's features.
The Use Case Diagram for the Trip Planner System defines how different types of users interact with the system and what functionalities they can access.

Defining User Roles It distinguishes between different types of users, such as travelers, administrators, and service providers.

Mapping Functionalities It visually represents key features such as trip creation, itinerary management, budget planning, and real-time updates.

Enhancing System Understanding It provides a high-level overview of the system's core interactions, helping developers and stakeholders understand the project's scope.

Actors in the Use Case Diagram

Traveler (User) Can register, log in, create trips, modify itineraries, check route suggestions, and track expenses.

Admin Manages the platform, verifies travel services, and ensures data integrity.

Service Provider (Hotels, Airlines, Transport Services) Updates availability and pricing of travel services.

Use Cases for the Trip Planner System

User Registration & Authentication Travelers create an account and log in securely.

Trip Planning & Itinerary Management Users create, edit, and manage their travel plans.

Budget Planning & Expense Tracking Users set budgets and monitor spending.

Route Optimization & Navigation System provides the best routes based on real-time data.

Booking Services Users can book hotels, flights, and car rentals.

Collaboration & Sharing Users share itineraries with friends or co-travelers.

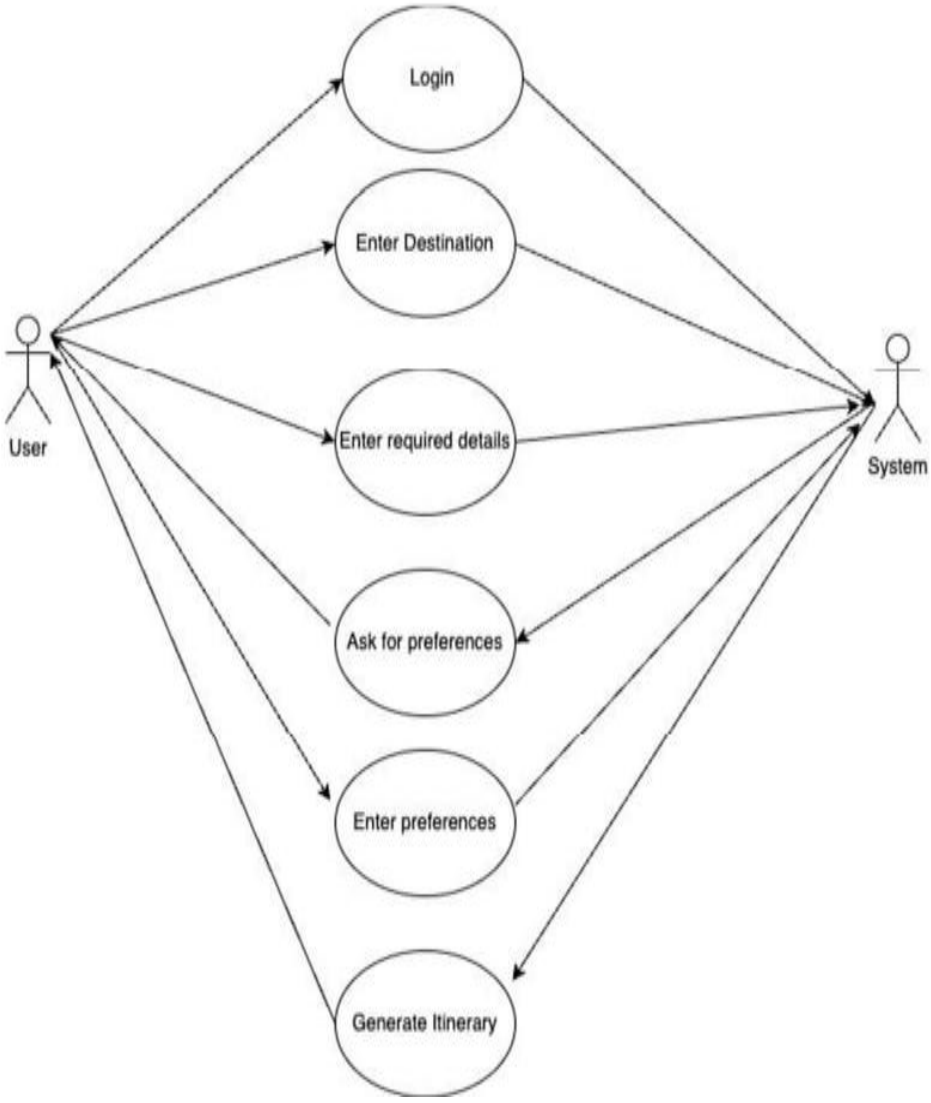

Figure 6.1: Use Case Diagram

6.2 System Architecture

The System Architecture of the Trip Planner System outlines the software and hardware components, their interactions, and the overall data flow.

Key Components of the System Architecture

Presentation Layer (Frontend) The user interface (UI) of the application, developed using React, HTML, CSS, and JavaScript, enables seamless user interaction.

Application Layer (Backend) This layer, built using Node.js and Express.js, processes business logic and manages trip planning algorithms.

Database Layer MySQL or MongoDB stores user details, trip data, budget information, and service provider details.

API Layer Integrates third-party services such as Google Maps API, Booking APIs, and Payment Gateways to enhance functionality.

Security Layer Ensures data encryption, user authentication, and secure transactions through OAuth, JWT, and SSL protocols.

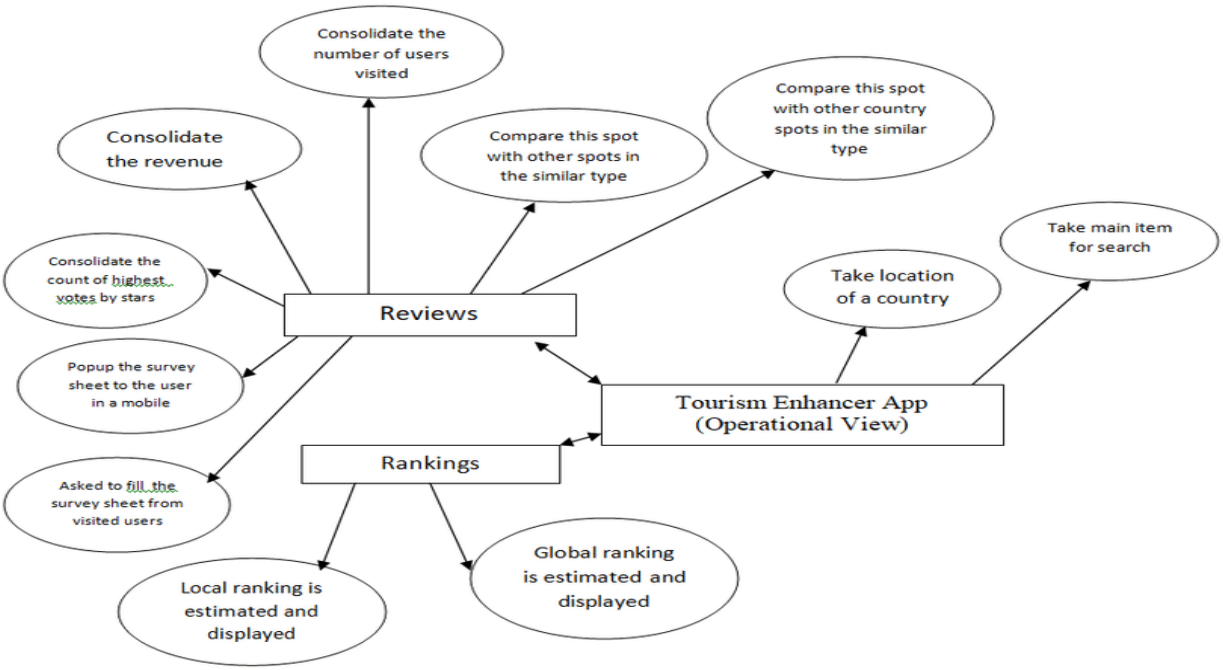

Figure 6.2: System Architecture Diagram

6.3 Data Flow in the System

User Interaction The traveler accesses the system via a web or mobile application.

Backend Processing User requests are processed in the application layer, interacting with the database and external APIs.

Response & Display The system fetches relevant data (trip details, maps, expenses) and displays it to the user.

Real-Time Updates Notifications, alerts, and route changes are updated dynamically.

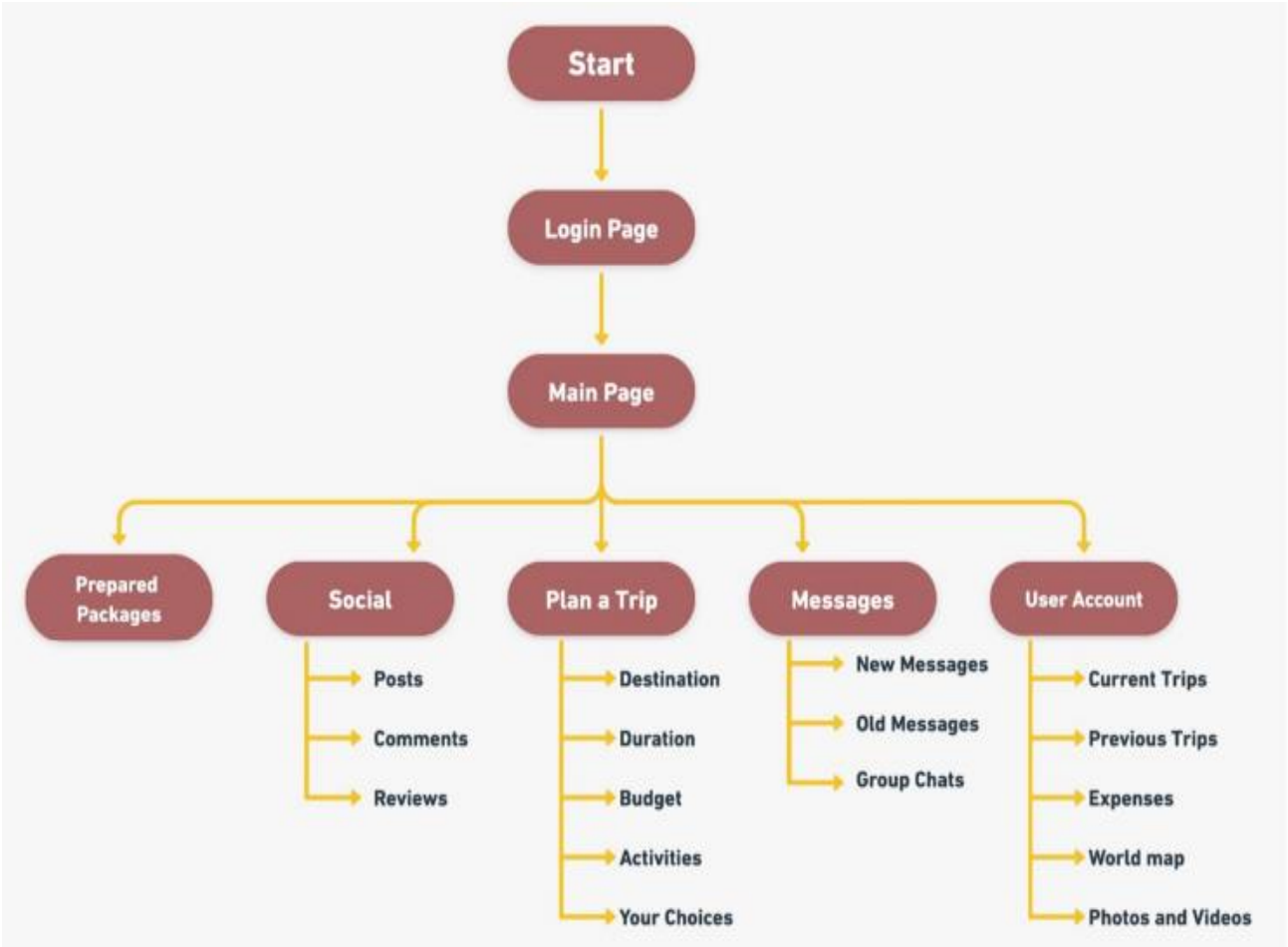

Figure 6.3: Data Flow Diagram

CHAPTER 7
CODING

TripPlanner

```
import React, { useState } from 'react';
import { Card, CardContent } from "@/components/ui/card";
import { Button } from "@/components/ui/button";
import { Input } from "@/components/ui/input";
import { Textarea } from "@/components/ui/textarea";
import { CalendarIcon, Plus } from "lucide-react";
import { format } from "date-fns";

export default function TripPlanner() {
  const [trips, setTrips] = useState([]);
  const [trip, setTrip] = useState({
    destination: '',
    startDate: '',
    endDate: '',
    activities: ''
  });

  const handleChange = (e) => {
    setTrip({ ...trip, [e.target.name]: e.target.value });
  };

  const addTrip = () => {
    if (!trip.destination || !trip.startDate || !trip.endDate) return;
    setTrips([...trips, trip]);
    setTrip({ destination: '', startDate: '', endDate: '', activities: '' });
  };

  return (
    <div className="p-6 max-w-3xl mx-auto space-y-6">
      <h1 className="text-3xl font-bold text-center">Trip Planner</h1>

      <Card className="p-4">
        <CardContent className="space-y-4">
          <Input
            name="destination"
            placeholder="Destination"
            value={trip.destination}
            onChange={handleChange}
          />
          <div className="flex gap-4">
            <Input
              name="startDate"
              type="date"
              value={trip.startDate}
              onChange={handleChange}
              icon={<CalendarIcon />}
            />
```

```
            <Input
              name="endDate"
              type="date"
              value={trip.endDate}
              onChange={handleChange}
            />
          </div>
          <Textarea
            name="activities"
            placeholder="Planned activities"
            value={trip.activities}
            onChange={handleChange}
          />
          <Button onClick={addTrip} className="w-full">
            <Plus className="mr-2 h-4 w-4" /> Add Trip
          </Button>
        </CardContent>
      </Card>

      <div className="space-y-4">
        {trips.map((t, idx) => (
          <Card key={idx}>
            <CardContent className="p-4 space-y-2">
              <h2 className="text-xl font-semibold">{t.destination}</h2>
              <p className="text-sm text-gray-500">
                {format(new Date(t.startDate), "MMM d, yyyy")} - {format(new Date(t.endDate),
"MMM d, yyyy")}
              </p>
              <p>{t.activities}</p>
            </CardContent>
          </Card>
        ))}
      </div>
    </div>
  );
}


import React from "react";
import ReactDOM from "react-dom/client";
import { BrowserRouter } from "react-router-dom";
import App from "./App";
import "./index.css"; // Tailwind CSS

ReactDOM.createRoot(document.getElementById("root")).render(
  <React.StrictMode>
    <BrowserRouter>
      <App />
    </BrowserRouter>
  </React.StrictMode>
);
```

```
import { useState } from "react";
import { useNavigate } from "react-router-dom";

export default function AddTrip() {
  const navigate = useNavigate();
  const [form, setForm] = useState({
    destination: "",
    startDate: "",
    endDate: "",
    activities: "",
  });

  const handleChange = (e) =>
    setForm({ ...form, [e.target.name]: e.target.value });

  const handleSubmit = () => {
    const existing = JSON.parse(localStorage.getItem("trips") || "[]");
    localStorage.setItem("trips", JSON.stringify([...existing, form]));
    navigate("/trips");
  };

  return (
    <div className="max-w-xl mx-auto bg-white p-6 rounded shadow space-y-4">
      <h2 className="text-xl font-bold">Add a New Trip</h2>
      <input
        className="w-full border p-2 rounded"
        name="destination"
        placeholder="Destination"
        value={form.destination}
        onChange={handleChange}
      />
      <div className="flex gap-4">
        <input
          type="date"
          className="w-full border p-2 rounded"
          name="startDate"
          value={form.startDate}
          onChange={handleChange}
        />
        <input
          type="date"
          className="w-full border p-2 rounded"
          name="endDate"
          value={form.endDate}
          onChange={handleChange}
        />
      </div>
      <textarea
        className="w-full border p-2 rounded"
        name="activities"
```

```
        placeholder="Planned Activities"
        value={form.activities}
        onChange={handleChange}
      />
      <button onClick={handleSubmit} className="w-full bg-blue-600 text-white p-2 rounded">
        Save Trip
      </button>
    </div>
  );
}


import { useEffect, useState } from "react";
import TripCard from "../components/TripCard";

export default function AllTrips() {
  const [trips, setTrips] = useState([]);

  useEffect(() => {
    const stored = localStorage.getItem("trips");
    if (stored) setTrips(JSON.parse(stored));
  }, []);

  return (
    <div className="space-y-4">
      <h2 className="text-2xl font-bold">All Trips</h2>
      {trips.length === 0 ? (
        <p className="text-gray-500">No trips planned yet.</p>
      ) : (
        trips.map((trip, i) => <TripCard key={i} trip={trip} />)
      )}
    </div>
  );
}

// firebaseConfig.js
import { initializeApp } from "firebase/app";
import { getFirestore } from "firebase/firestore";

const firebaseConfig = {
  apiKey: "YOUR_API_KEY",
  authDomain: "YOUR_DOMAIN",
  projectId: "YOUR_PROJECT_ID",
  storageBucket: "YOUR_BUCKET",
  messagingSenderId: "YOUR_SENDER_ID",
  appId: "YOUR_APP_ID"
};

const app = initializeApp(firebaseConfig);
const db = getFirestore(app);

export { db };
```

```
import { NavigationContainer } from "@react-navigation/native";
import { createNativeStackNavigator } from "@react-navigation/native-stack";
import Home from "./screens/Home";
import AddTrip from "./screens/AddTrip";

const Stack = createNativeStackNavigator();

export default function App() {
  return (
    <NavigationContainer>
      <Stack.Navigator>
        <Stack.Screen name="Trips" component={Home} />
        <Stack.Screen name="AddTrip" component={AddTrip} />
      </Stack.Navigator>
    </NavigationContainer>
  );
}



import { View, Text, Button, FlatList } from "react-native";
import { useEffect, useState } from "react";
import { db } from "../firebaseConfig";
import { collection, getDocs } from "firebase/firestore";

export default function Home({ navigation }) {
  const [trips, setTrips] = useState([]);

  const fetchTrips = async () => {
    const snapshot = await getDocs(collection(db, "trips"));
    const data = snapshot.docs.map((doc) => ({ id: doc.id, ...doc.data() }));
    setTrips(data);
  };

  useEffect(() => {
    const unsubscribe = navigation.addListener("focus", fetchTrips);
    return unsubscribe;
  }, [navigation]);

  return (
    <View style={{ flex: 1, padding: 20 }}>
      <Button title="Add New Trip" onPress={() => navigation.navigate("AddTrip")} />
      <FlatList
        data={trips}
        keyExtractor={(item) => item.id}
        renderItem={({ item }) => (
          <View style={{ marginTop: 20, backgroundColor: "#eee", padding: 15, borderRadius: 10
}}>
```

```
            <Text style={{ fontWeight: "bold", fontSize: 18 }}>{item.destination}</Text>
            <Text>{item.startDate} → {item.endDate}</Text>
            <Text>{item.activities}</Text>
          </View>
        )}
      />
    </View>
  );
}


import { View, TextInput, Button } from "react-native";
import { useState } from "react";
import { db } from "../firebaseConfig";
import { collection, addDoc } from "firebase/firestore";

export default function AddTrip({ navigation }) {
  const [form, setForm] = useState({
    destination: "",
    startDate: "",
    endDate: "",
    activities: ""
  });

  const handleChange = (name, value) => setForm({ ...form, [name]: value });

  const saveTrip = async () => {
    await addDoc(collection(db, "trips"), form);
    navigation.goBack();
  };

  return (
    <View style={{ padding: 20 }}>
      <TextInput
        placeholder="Destination"
        value={form.destination}
        onChangeText={(text) => handleChange("destination", text)}
        style={{ borderBottomWidth: 1, marginBottom: 10 }}
      />
      <TextInput
        placeholder="Start Date"
        value={form.startDate}
        onChangeText={(text) => handleChange("startDate", text)}
        style={{ borderBottomWidth: 1, marginBottom: 10 }}
      />
      <TextInput
        placeholder="End Date"
        value={form.endDate}
        onChangeText={(text) => handleChange("endDate", text)}
        style={{ borderBottomWidth: 1, marginBottom: 10 }}
      />
```

```
      <TextInput
        placeholder="Activities"
        value={form.activities}
        onChangeText={(text) => handleChange("activities", text)}
        style={{ borderBottomWidth: 1, marginBottom: 20 }}
      />
      <Button title="Save Trip" onPress={saveTrip} />
    </View>
  );
}
```

```
const { Sequelize } = require("sequelize");
require("dotenv").config();

const sequelize = new Sequelize(
  process.env.DB_NAME,
  process.env.DB_USER,
  process.env.DB_PASS,
  {
    host: process.env.DB_HOST,
    dialect: "mysql",
  }
);

module.exports = sequelize;
```

```
const { DataTypes } = require("sequelize");
const sequelize = require("../config/db");

const Trip = sequelize.define("Trip", {
  destination: {
    type: DataTypes.STRING,
    allowNull: false,
  },
  startDate: {
    type: DataTypes.DATEONLY,
    allowNull: false,
  },
  endDate: {
    type: DataTypes.DATEONLY,
    allowNull: false,
  },
  activities: {
    type: DataTypes.TEXT,
  },
});
```

```
const express = require("express");
const router = express.Router();
const Trip = require("../models/Trip");

// Create trip
router.post("/", async (req, res) => {
  try {
    const newTrip = await Trip.create(req.body);
    res.status(201).json(newTrip);
  } catch (err) {
    res.status(400).json({ error: err.message });
  }
});

// Get all trips
router.get("/", async (req, res) => {
  try {
    const trips = await Trip.findAll();
    res.json(trips);
  } catch (err) {
    res.status(500).json({ error: err.message });
  }
});

// Get single trip
router.get("/:id", async (req, res) => {
  try {
    const trip = await Trip.findByPk(req.params.id);
    if (!trip) return res.status(404).json({ error: "Not found" });
    res.json(trip);
  } catch (err) {
    res.status(500).json({ error: err.message });
  }
});

// Delete trip
router.delete("/:id", async (req, res) => {
  try {
    const deleted = await Trip.destroy({ where: { id: req.params.id } });
    res.json({ deleted });
  } catch (err) {
    res.status(500).json({ error: err.message });
  }
});
```

CHAPTER 8
RESULTS AND OUTPUT SCREENS

Figure 8.1: Front Page

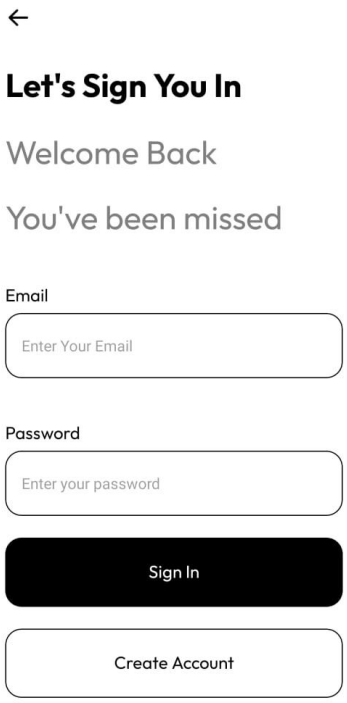

Figure 8.2: Signup Page

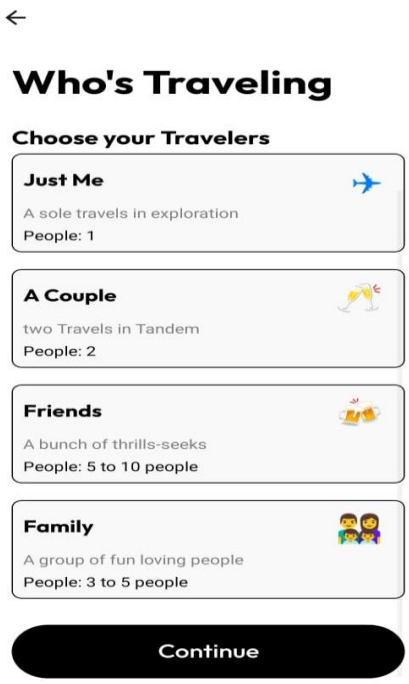

Figure 8.3: Choose your Travel

Figure 8.4: Review

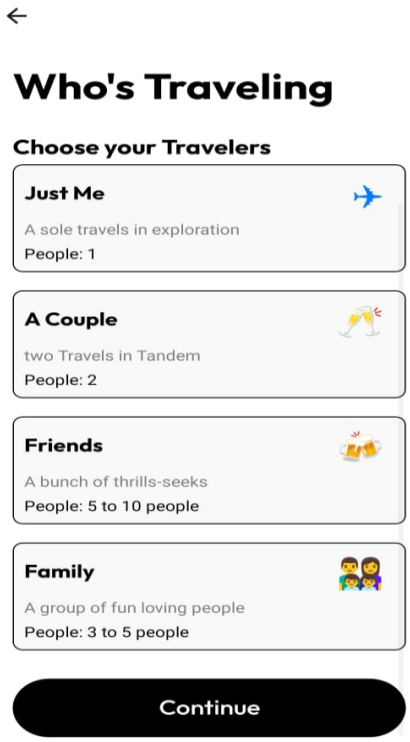

Figure 8.5: Who's Traveling

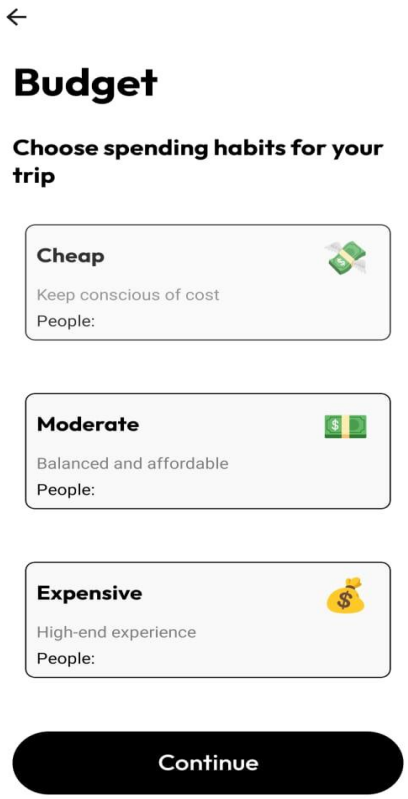

Figure 8.6: Budget

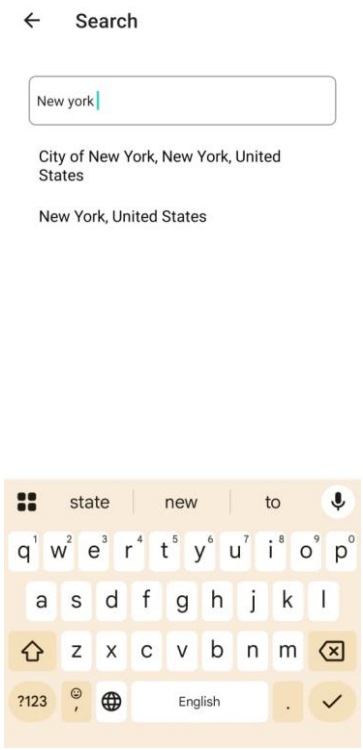

Figure 8.7: Search

CHAPTER 9
CONCLUSION AND FUTURE WORK

The Trip Planner System (TPS) is a dynamic and intelligent platform designed to streamline the travel planning process through automation, real-time data, and interactive features. It assists users in creating personalized itineraries, optimizing routes, managing budgets, and collaborating with co-travelers. While the current system is functional and effective, there are several areas where future enhancements could improve its efficiency, usability, and scalability.

Advanced AI-Based Recommendations

The Trip Planner System could integrate advanced artificial intelligence (AI) and machine learning (ML) algorithms to enhance travel recommendations. By analyzing user preferences, past travel history, and seasonal trends, the system could offer personalized suggestions for destinations, accommodations, restaurants, and activities. AI could also optimize itineraries based on real-time data such as weather conditions, traffic congestion, and local events.

Real-Time Traffic and Route Optimization

Incorporating real-time traffic updates and alternative route suggestions would enhance the user experience. By integrating with Google Maps API, GPS tracking, and public transport data, the system could provide users with the fastest and most efficient travel routes. Additional features such as voice-guided navigation, offline maps, and predictive route planning would further improve usability, especially for travelers exploring unfamiliar locations.

Smart Budget Management and Expense Tracking

Future versions of the Trip Planner System could include an automated budget manager that tracks expenses and suggests cost-saving options. The system could categorize expenses into accommodation, transportation, food, and activities, allowing users to set spending limits and receive alerts when exceeding their budget. Integration with digital payment gateways and currency converters would also improve financial planning for international travelers.

Cloud-Based Data Logging and Travel History

A cloud-based storage system could enable users to save their past travel itineraries and access them from any device. This feature would be useful for revisiting favorite trips, sharing travel logs with friends, or using previous plans as templates for new journeys. Additionally, historical data analysis could provide insights into travel patterns, preferred destinations, and seasonal cost variations to help users make informed decisions.

Social Integration and Collaborative Planning

The Trip Planner System could enhance collaboration by allowing users to share travel plans with friends and family. Features such as group itinerary editing, real-time chat, and voting on travel activities would make it easier for groups to coordinate trips. Integration with social media platforms could also allow users to share their itineraries, travel experiences, and recommendations directly from the app.

Enhanced Security and Data Privacy

As the system handles sensitive personal and financial data,enhanced security measures are essential. Future enhancements could include multi-factor authentication (MFA), end-to-end encryption, and biometric login options to protect user accounts. Additionally, secure payment processing with fraud detection mechanisms would ensure safer transactions for bookings and reservations.

IoT and Smart Device Integration

Future iterations of the Trip Planner System could incorporate Internet of Things (IoT) capabilities, allowing seamless connectivity with smart travel devices. Examples include:

- **Smart luggage tracking** Users could track their luggage location in real-time.
- **Hotel room automation** Integration with smart hotel systems for keyless entry and customized room settings.
- **Wearable travel assistants** Notifications on smartwatches for flight updates, weather changes, or itinerary reminders.

Voice and Chatbot Assistance

Adding AI-powered virtual assistants would provide users with real-time assistance for travel inquiries. A chatbot could handle booking confirmations, weather updates, and local attraction suggestions, making travel planning more interactive and convenient. Integration with voice assistants like Google Assistant, Alexa, or Siri would further improve accessibility for hands-free trip planning.

Offline Mode for Remote Travel

Many travelers visit remote locations with limited or no internet connectivity. Enhancing the Trip Planner System with an offline mode would allow users to access their itinerary, view offline maps, and track expenses without an internet connection. Once connected to a network, the system could sync data automatically.

Sustainable Travel and Eco-Friendly Recommendations

To promote eco-friendly tourism, the system could include features that encourage users to choose sustainable travel options. This could involve recommendations for eco-friendly hotels, carbon footprint calculators, public transportation over private vehicles, and tips for responsible travel. Incentives such as discounts for choosing green options could motivate users to adopt sustainable travel habits.

REFERENCES

JOURNALS / RESEARCH PAPERS

[1] Zhang, L., & Sun, X. (2016). Designing a Trip Planner Application for Groups: Exploring Group Tourists' Trip Planning Requirements. In Proceedings of the 2016 CHI Conference Extended Abstracts (pp. 1325–1332)..

[2] Sun, D., Peng, Z.-R., Shan, X., Chen, W., & Zeng, X. (2011). Development of Web-Based Transit Trip-Planning System Based on Service-Oriented Architecture. Transportation Research Record, 2217(1), 73–81

[3] Ulloa, L., Lehoux-Lebacque, & Roulland, F. (2018). Trip Planning within a Multimodal Urban Mobility. IET Intelligent Transport Systems, 12(2), 138–147.

BOOKS

[4] Road Trip Planner Journal Essential Travel Organizer to Record and Plan Your Road Trip Adventures .This journal helps in organizing and recording road trip adventures, serving as both a planner and a memory book.

[5] Lamare Travel Journal F Bucketlist, Roadtrip & Vacation Planner . An undated travel diary and planner that includes a bucket list, road trip planning, and space for journaling experiences.

WEBSITES (with exact URL up to page)

[6] Geeks for Geeks: GeeksforGeeks

[7] W3School: https://w3schools.com

PROJECT SUMMARY

About Project

Title of the project	DESIGN & DEVELOPMENT OF TRIP PLANNER SYSTEM
Semester	8th
Members	4th
Team Leader	Mehkasha Mansoori
Describe role of every member in the project	M Maaz Nafees Frontend Developer (UI/UX Designer) Irtiqa Aafreen Backend Developer (API Handler) Mehkasha Mansoori Backend Developer (Database) Mujtaba Farooqui Frontend Developer (UI/UX Designer)
What is the motivation for selecting this project?	The motivation for developing a Trip Planner System stems from the growing need for a seamless, efficient, and user-friendly travel planning solution. In today's fast-paced world, travelers face numerous challenges, including selecting destinations, optimizing routes, managing budgets, and coordinating accommodations. A smart trip planning system provides a centralized platform where users can organize their travel itineraries, access real-time information, and receive personalized recommendations based on their preferences.
Project Type (Desktop Application, Web Application, Mobile App, Web)	Web Application

Tools &Technologies

Programming language used	ReactNative NodeJS
IDE used (with version)	IntelliJ IDE
Front End Technologies (with version, wherever Applicable)	NA
Back End Technologies (with version, wherever applicable)	NA
Database used (with version)	FireBase

Software Design& Coding

Is prototype of the software developed?	Yes
SDLC model followed (Waterfall, Agile, Spiral etc.)	Agile
Why above SDLC model is followed?	Agile is a SDLC model that defines how software development needs to be done. It's not a single or specific method, and it is the collection of various methodologies and best practices that follow the value statement signed with the customer
Justify that the SDLC model mentioned above is followed in the project.	Since, we didn't exactly know all the functionalities or the functionalities were frequently changing, we use Agile model, so that we could make desired changes whenever needed.
Software Design approach followed (Functional or Object Oriented)	
Name the diagrams developed (According to the Design approach followed)	Use Case diagram, Architecture
No. of Tiers (example 3-tier)	3-tier
Total no. of front-end pages	
Front end validations applied (Yes / No)	Yes
Session management done (in case of web applications)	
Is application browser compatible (in case of web applications)	
Exception handling done (Yes / No)	No
Commenting done in code (Yes / No)	Yes
Naming convention followed (Yes / No)	Yes
Give titles of Use-cases	Use-case Diagram

Project Requirements

MVC architecture followed (Yes / No)	
If yes, write the name of MVC architecture followed (MVC-1, MVC-2)	
Design Pattern used (Yes / No)	
If yes, write the name of Design Pattern used	
Interface type (CLI / GUI)	
No. of Actors	
Name of Actors	
Total no. of Functional Requirements	
List few important non-Functional Requirements	Correctness, Flexibility, Reliability and Maintainability

Testing

Which testing is performed? (Manual or Automation)	Manual
Is Beta testing done for this project?	No

Write project narrative covering above mentioned points

This project aims to enable users to plan trips by selecting destinations, adding schedules, and viewing routes or itineraries, all controlled via interactive UI components. This involves designing forms, buttons, and displays using technologies like React, while leveraging state management to update and reflect changes in the user's itinerary dynamically.

Just like each segment of a 7-segment display needs to be turned on/off to show numbers, each feature of the planner (e.g., place name, time, transport method) must be actively managed in the UI. For example, showing a full itinerary requires all components to be filled and displayed, while a quick stopover might need just a destination and time.

Writing code to convert user input into planned routes and schedules forms the core of the software logic. This can be enhanced with features like date pickers, maps integration, and dynamic lists. Additionally, extending the system to support multi-day itineraries introduces the concept of component reuse and dynamic rendering, analogous to multiplexing in hardware – optimizing the interface to handle complex data while keeping the UI clean and efficient.

Guide Signature
(Prof. Nargish Gupta)

Irtiqa Aafreen	0187CS211075
M.Maaz Nafees	0187CS211095
Mehkasha	0187CS211101
Mujtaba Farooqui	0187CS211103

APPENDIX-1 GLOSSARY OF TERMS

(In alphabetical order)

A

API (Application Programming Interface)	A set of functions and protocols that allow different software components to communicate with each other. In the Trip Planner System, APIs are used to fetch travel-related data such as flight schedules, hotel availability, weather updates, and map services.

B

Backend Development	The server-side logic responsible for handling user requests, processing trip data, and storing information in the database. It involves technologies like Node.js, Express.js, and MySQL to manage the core functionalities of the system.

C

Cloud Computing	The use of remote servers hosted on the internet to store, manage, and process data. Cloud platforms such as AWS, Google Cloud, or Microsoft Azure provide scalability, security, and real-time accessibility for users.

D

Database Management System	A structured system that manages user profiles, travel itineraries, historical trip data ,and booking records .MySQL,PostgreSQL ,or MongoDB are commonly used for handling large-scale travel-related data efficiently.

E

Error Handling	A crucial component of software development, ensuring that the Trip Planner System remains stable even when unexpected errors occur. This includes exception handling, debugging tools, and automated error logging mechanisms.

F

Frontend Development	The design and implementation of the User Interface (UI) and User Experience (UX) using web technologies like React.js, HTML, CSS, and JavaScript.The frontend allows users to interact with the system seamlessly.

G	
GPS (Global Positioning System)	A navigation system used to determine real-time locations, helping users plan optimized routes, track their journey, and receive location-based recommendations. The system integrates Google Maps API or OpenStreetMap for accurate travel assistance.
I	
Interactive Maps	A feature that allows users to visualize their travel routes, explore destinations, and receive location-based alerts .The system integrates interactive maps with features such as real-time traffic updates, distance calculations, and nearby attractions.
L	
Localization	The adaptation of the Trip Planner System for different languages, currencies, time zones, and cultural preferences to provide a personalized experience for users from various regions.
M	
Mobile App Development	The development of a mobile version of the Trip Planner System using React Native, Flutter, or Swift (for iOS) and Kotlin (for Android). The mobile app ensures on-the-go trip planning and notifications.
R	
Real-Time Updates	A critical feature that provides users with live updates on flight schedules, weather forecasts, traffic conditions, and hotel availability to ensure a smooth travel experience.
S	
Security Features	Measures taken to protect user data and Multi-Factor Authentication (MFA) ,Adding extra layers of security for login, Restricting unauthorized access to user accounts and admin panels.
U	
User Authentication	A secure login mechanism ensuring that only authorized users can access their accounts. It includes ,Email & Password Authentication , Social Media Login (Google, Facebook, Apple ID).
W	
Web Development	The process of building the Trip Planner System's website using technologies like HTML & CSS For layout and styling ,React.js For dynamic user interfaces,Node.js & Express.js For backend services.