



Business information systems

Faculty of Commerce and
Business Administration

Helwan University



Final year graduation project

Journee

Project no. 14

No.	Name	ID
1	Ahmed Mahmoud Labib	512190864
2	Nour Mohsen Sadeq	512191633
3	Hajar Ahmed Mohamed	512191666
4	Nada Mohamed Ibrahim	512191648
5	Esraa Ahmed Abd-El Kawy	512190881
6	Ali Ashraf Othman	512191230

Supervisors

(IT)

Prof. Ahmed Sedawy

(Business)

Prof. Hoda Ali

Year 2024

Declaration

we hereby certify that this material is entirely our own work, and that we have exercised reasonable care to ensure that the work is original and does not to the best of our knowledge breach any law of copyright and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of our work.

Team signature:

1) _____

2) _____

3) _____

4) _____

5) _____

6) _____

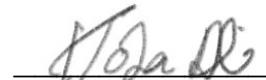
Registration number: _____

Date: _____

Drs. signatures

IT Prof.Dr.

Business Prof.Dr.



Abstract

Journee is a web-based platform designed to revolutionize travel planning by putting the needs of the traveler at the forefront. Recognizing that generic, one-size-fits-all itineraries fall short in delivering fulfilling travel experiences, Journee focuses on personalization. Users create detailed travel profiles outlining their preferences, from desired destinations and budgets to specific travel themes and interests. Journee's intelligent itinerary engine processes this information in conjunction with real-time travel data. This allows the platform to provide customized travel plans featuring recommended destinations, attractions, and activities aligned with the user's unique profile. Emphasizing flexibility, Journee also empowers users to modify and refine suggested itineraries, ensuring that their final travel plans are a perfect match.

Table of Contents

Declaration	I
Abstract	II
Table Of Contents	III
List Of Tables	VI
List Of Figures	VII
Chapter 1 Introduction	1
1.1 introduction	2
1.2 project idea and scope	2
1.3 system objectives	3
1.4 system features	4
1.5 competition in the niche market	6
1.6 system requirements	12
1.7 system users	13
1.8 system methodology	14
1.9 project time plan	15
Chapter 2 Business Plan	16
2.1 executive summary	17
2.2 vision and mission	22
2.3 business idea and market	23
2.4 marketing plan	29
2.5 legal form	35
2.6 start-up capital and its sources	36
2.7 financial plan	38

Chapter 3 System Analysis	41
3.1 introduction	42
3.2 user requirements	44
- functional requirements	44
- non-functional requirements	45
3.3 system requirements	46
3.4 system diagrams	47
- use case diagram	47
- sequence diagram	51
- activity diagram	56
Chapter 4 System Design	60
4.1 introduction	61
4.2 class diagram	61
4.3 data structure (ERD)	64
4.4 database mapping	67
Chapter 5 System Implementation	69
5.1 introduction	70
5.2 system screenshots	72
Chapter 6 Testing Methodology	96
6.1 introduction	97
6.2 testing methodology	97
6.3 login validation	98

6.4 sign-up validation	100
6.5 main functions validations	102
Chapter 7 Conclusion & Future work	108
7.1 introduction	109
7.2 conclusion	109
- a summary of key arguments and results	109
- a short discussion of the implications of research	110
- reflecting on the research process	110
- show what new knowledge you've contributed to the field	111
7.3 future work	112
References	114

List of tables

1.Comparative analysis	11
2.Executive summary	17
3.Owner 1	18
4.Owner 2	19
5.Owner 3	19
6.Owner 4	20
7.Owner 5	20
8.Owner 6	21
9.Survey analysis	30
10.Marketing campaign cost	35
11.Start-up capital	36
12.Items specifications	37
13.Start-up capital sources	37
14.Opening balance	38
15.Monthly sales	38
16.Monthly operational costs	39
17.Cash flow	39
18.Profit margin	40

List of figures

1. Tripadvisor Homepage	6
2. Google trips Homepage	7
3. Kayak Homepage	8
4. Culture Trip Homepage	9
5. TourRadar Homepage	10
6. Agile-Scrum Methodology framework	14
7. Project timeline	15
8. Use-case diagram	48
9. User Use-case diagram	49
10. Admin Use-case diagram	50
11. Register-Login Sequence	52
12. Itinerary-generation sequence	53
13. Payment sequence	54
14. Review sequence	55
15. Login activity	57
16. Itinerary generation activity	58
17. User account management activity	59
18. Review activity	59
19. Class diagram	63
20. ERD	66
21. Database mapping	68

Chapter 1

Introduction

1.1 Introduction

This chapter aims to give a basic overview of the whole Project. Starting with the core concept idea of our business and scope. The chapter will also touch upon both the objectives of the platform and the features it will deliver. As well as other competitors in the market exploiting the same niche. We are going to focus on the system requirements and the potential users of such a system as well. Lastly, the chapter will end with shedding light on the system methodology used to develop our platform and the timeline of each developed feature.

1.2 Project Idea and Scope

- Idea:

Journee's core functionality is simple, it's an AI-powered innovative web platform that leverages the power of artificial intelligence to create personalized travel itineraries tailored to each user's unique preferences, budget, and desired destinations. Journee aims to streamline the travel planning process by automating itinerary creation, allowing users to focus on the excitement of discovering new places and experiences and spare its customers the hustle and bustle of planning ahead and empowers them with creating a full itinerary with a click of a button.

- Scope:

In this initial phase, users will be able to create profiles to store their preferences and travel habits. They can then input their desired travel dates, budget, and destination, along with their preferred travel themes (e.g., history, culture, nature, relaxation). Journee's AI engine will then process this information to generate a customized itinerary that aligns with the user's unique interests and constraints. The generated

itinerary will be presented in a visually appealing and user-friendly format, allowing users to easily review the suggested activities, attractions, and accommodation options. While this initial version focuses on core itinerary generation, Journee's future development may encompass features such as real-time flight and hotel booking integration, social sharing tools for sharing itineraries with friends and family, or even comprehensive destination guides with user reviews and recommendations.

1.3 System Objectives

Journee's primary objectives center around delivering a seamless and empowering travel planning experience. The system aims to:

- **Provide** users with a streamlined way to create personalized travel itineraries based on their unique interests and preferences. This includes understanding their desired travel styles (e.g., adventure, relaxation, cultural immersion), preferred activities (e.g., sightseeing, hiking, food, and wine tasting), giving them the option to choose their travel group to be incorporated into their itinerary as well as any specific interests or accessibility needs, they may have.
- **Generate** itineraries that optimize for user-selected travel themes, budget limitations, and desired travel durations. Journee should strive to create itineraries that offer a fulfilling and balanced experience while adhering to the user's budget constraints and time limitations.
- **Enable** users to discover new destinations and attractions that align with their chosen travel styles. By leveraging AI and comprehensive travel data, Journee can introduce users to hidden gems and unique experiences that perfectly

match their preferences, potentially leading them beyond the typical tourist destinations.

- **Allow** users to refine and customize the AI-generated itineraries to ensure a perfect fit for their needs. Users should have the flexibility to adjust the suggestions provided by Journee, add their own personal touches, and tailor the itinerary to their specific preferences and interests.
- **Offer** a visually appealing and intuitive interface for easy interaction and navigation throughout the itinerary creation process. Journee's user interface should be user-friendly and visually engaging, allowing users to effortlessly input their preferences, explore generated suggestions, and personalize their itineraries with ease.

1.4 System Features

- Core Features:
 1. **User Accounts:** Secure registration and login allow users to save their travel preferences and past itineraries, streamlining future trip planning.
 2. **Personalized Travel Profiles:** Users can capture their favorite destinations, travel themes (adventure, relaxation, culinary, etc.), and other preferences to create a personalized profile that informs the AI engine and tailors future trip suggestions to their unique interests.
 3. **AI-Powered Itinerary Generation:** Journee's intelligent engine processes user input from their profile and trip planning preferences, including destinations, dates, budget, and travel themes. Based on this information, the AI generates a customized daily itinerary that suggests

activities, attractions, and locations that align with the user's interests and constraints.

4. **Intuitive Trip Planning:**

- a) A destination search function with autocomplete suggestions helps users efficiently discover new locations and explore their options.
- b) A flexible date selector allows users to choose their travel dates with ease, using a calendar view for a clear overview.
- c) A budget input tool enables users to define their overall trip budget and optionally allocate funds to specific categories like accommodation, transportation, and activities, providing them with greater control over their spending.

1.5 Competition in the niche market

1. Tripadvisor

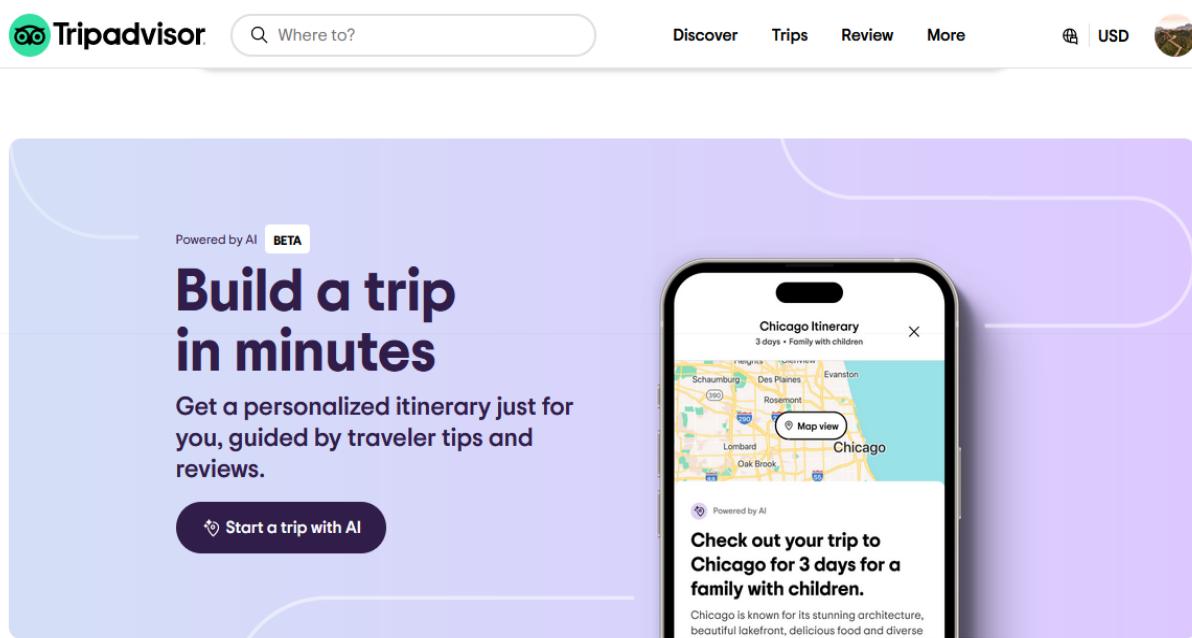


Fig.1: TripAdvisor Homepage

Our biggest competitor so far, especially after implementing their AI itinerary planner. A massive, crowdsourced platform for reviews, photos, and forums related to hotels, restaurants, and attractions worldwide.

It is excellent for comparing options and getting insights from fellow travelers. However, requires users to piece together their trip across various sections of the site (reviews, maps, forums), making it less streamlined for full itinerary planning. Also, their newly implemented feature doesn't take the user budget in consideration, giving Journee an edge over them.

2. Google trips

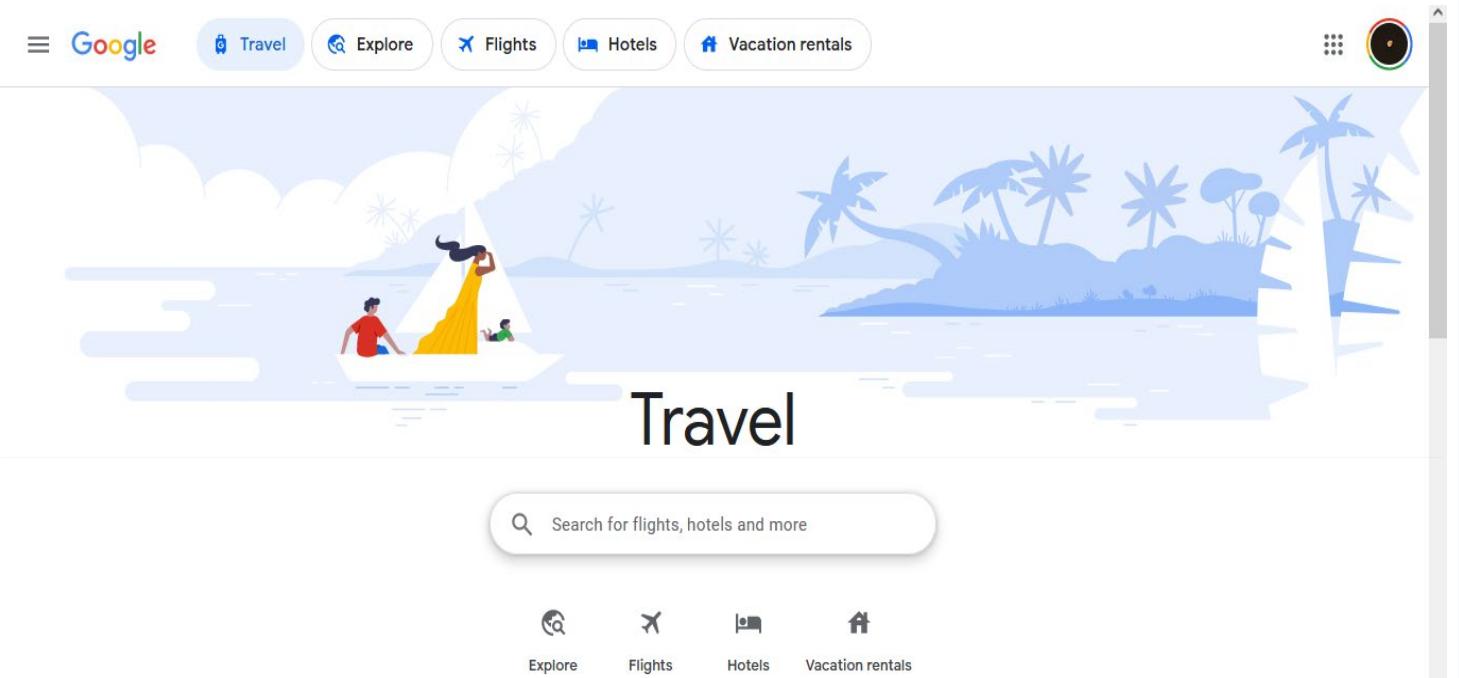


Fig.2: Google trips Homepage

A travel planning app that aggregates information from your Gmail (like flight or hotel reservations) and lets you access guides/recommendations on destinations. It leverages the power of Google Search and Maps integration. Good for quick destination research and saving places of interest but less robust as a dedicated itinerary generator.

3. Kayak

The screenshot shows the Kayak homepage. On the left, there's a sidebar with links: Sign in, Flights (selected), Stays, Car Rental, Explore, Direct, Travel Restrictions, Trips, KAYAK for Business (NEW), and English. The main area has a header "Where do you want to go?" with dropdowns for Return, 1 adult, Economy, 0 bags. Below is a search bar with "Cairo (CAI)" and a plus sign, a double arrow icon, a "To?" field, and date pickers for "Sun 7/4" and "Sun 14/4". To the right is a red search button. Below the search bar, there's a section titled "What KAYAK brings to the table." with four cards: "Big names, great deals" (Search 100s of travel sites to compare prices), "Search without worry" (The prices you see aren't affected by your searches), "Book with flexibility" (Easily find flights that have flexible bookings), and "Trusted and free" (We're completely free to use - no hidden charges or fees).

Fig.3: Kayak Homepage

Kayak is a metasearch tool, functioning as a one-stop shop to compare flight, hotel, and even travel package deals from a wide range of travel providers. Users can search for flights, hotels, or rental cars individually, or combine them into a single vacation package for potential savings. However, it doesn't have built-in itinerary creation tools or functionalities focused on personalizing a trip based on user preferences or interests. While it can help with the logistics of booking flights, hotels, and potentially car rentals, it doesn't assist in creating a day-to-day schedule of activities or attractions for a well-rounded travel experience.

4. Culture Trip

The screenshot shows the Culture Trip homepage with three travel offers:

- Japan By Train: The Grand Tour** (12 days . Rail) - From US\$4799 **US\$3839**. Booked in the last 24h.
- Edinburgh And The Scottish Highlands...** (7 days . Rail) - From US\$2799 **US\$2239**. Booked in the last 24h.
- A Taste Of Taiwan By Train** (9 days . Rail) - From US\$2199 **US\$1759**. Booked in the last 24h.

Fig.4: Culture Trip Homepage

A content-driven platform curating articles, stories, and videos focused on destinations, with an emphasis on arts, culture, food, and less conventional travel experiences. Great for inspiration and discovering unique things to do. Limited features for concrete travel planning (booking, detailed itineraries, etc.)

5. TourRadar

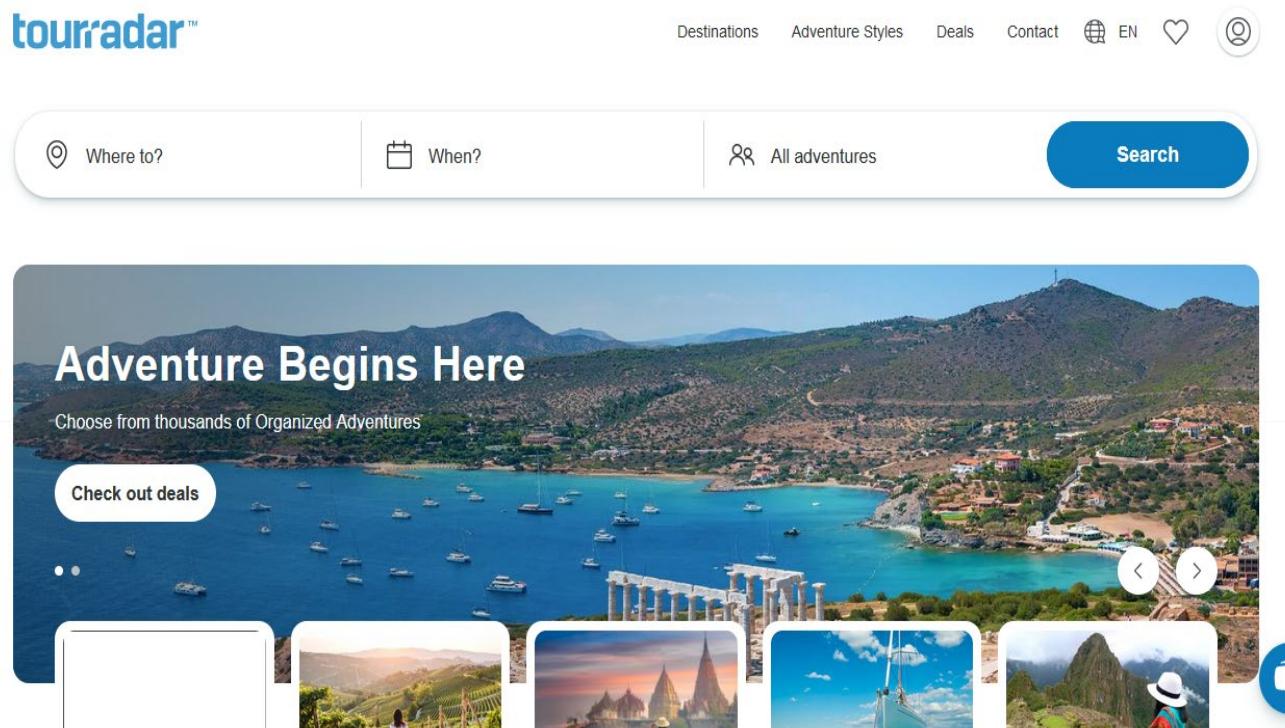


Fig.5: TourRadar Homepage

A marketplace for finding and booking pre-designed multi-day tours from various operators around the world. Useful if you want a fully arranged tour with a set itinerary. Offers little flexibility for independent travelers seeking complete customization.

Comparative analysis

- ✓: Yes
- ✗: No
- ■: Some

Business features \ Journee	Journee	Google Trips	TripAdvisor	Kayak	CultureTrip	TourRadar
Registration	✓	✓	✓	✓	✓	✓
AI-Powered Itinerary Suggestion	✓	■	✓	✗	✗	✗
Focus on Personalization	✓	■	✓	✗	■	✗
Niche Travel Theme Planning	✓	✗	✓	✗	■	✗
Itinerary Customization	✓	✗	■	✗	✗	✗
Visual Itinerary Presentation	✓	■	✓	✗	✓	✓
Flights Integration	✓	✓	✗	✓	✗	✗
Hotel Suggestion (aligned with budget)	✓	■	✗	■	✗	✗
Real-Time Cost Updates	✓	✓	■	■	✗	✗

Table 1: Comparative analysis

As can be concluded from the comparative analysis, Journee excels at giving utmost personalization options to users, ultimately freeing them from restrictions and technical constraints

1.6 System requirements

1. Hardware:

Journee is designed to be compatible with a wide range of devices, from [desktops and laptops](#) to tablets and smartphones. To ensure a seamless experience, a device with at least [2GB of RAM](#) is recommended. Since Journee doesn't require high-end hardware, it makes it naturally accessible to a broad range of users.

2. Software:

Journee is built to work with the most recent versions of popular web browsers. This includes [Google Chrome](#), [Mozilla Firefox](#), [Microsoft Edge](#), and [Safari](#).

3. Network:

A stable internet connection is essential to access Journee's features and functionalities. [Broadband or mobile data connections](#) are recommended for optimal performance. However, Journee is designed to be efficient, and even users with moderate internet speeds should be able to enjoy the core functionalities of the platform.

1.7 System Users

1. Typical user (traveler):

- Creating an account (registering and signing in)
- Building a detailed travel profile (preferences, Themes, etc.)
- Search for destinations.
- Inputting trip dates and budgets
- Receiving AI-generated itinerary suggestions
- Saving different of itinerary

2. Administrator:

- Database management (though this may be semi-automated)
- Monitoring system performance and addressing issues.
- Potentially adding or updating destination/attraction data
- Managing user accounts (in case of troubleshooting)

1.8 System Methodology

Journee's development process is adhering to [Agile](#) principles, specifically the [Scrum](#) framework. This iterative and adaptable methodology aligns well with our project goals for several reasons. First, Agile's focus on short development cycles ([sprints](#)) allows us to deliver core functionalities early and often (especially since we lack the expertise and are dealing with new stacks like [React](#) and [Django](#)). This facilitates early user feedback, which is crucial for refining Journee's functionalities and ensuring they meet user needs. Second, the collaborative nature of Agile, with its emphasis on daily stand-up meetings and sprint planning sessions, fosters open communication and efficient problem-solving within the development team. Regular sprint [retrospectives](#) will also be instrumental in continuously evaluating our progress and identifying areas for improvement throughout the development lifecycle.

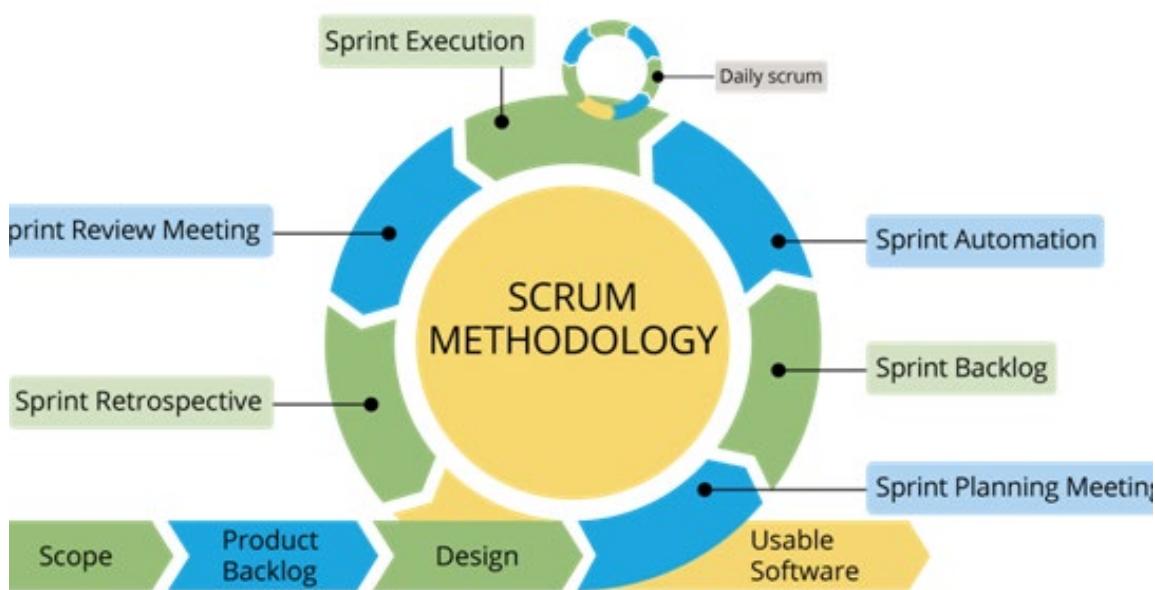


Fig.6: Agile-Scrum Methodology framework

1.9 Project Time Plan

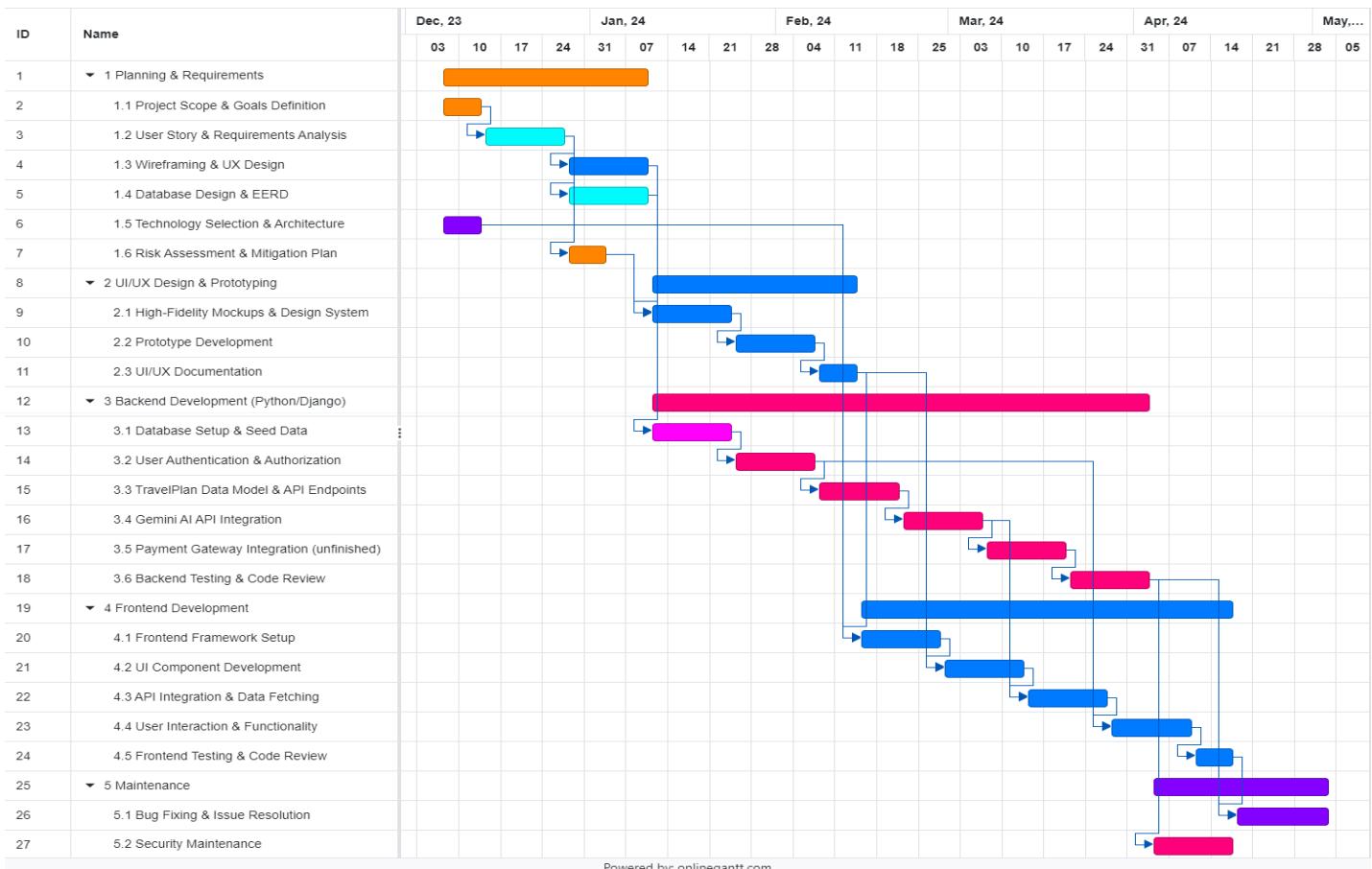


Fig.7: Project timeline

It begins with planning and requirements, encompassing defining project scope, user needs, and technical choices. This is followed by UI/UX design, where the website's looks are crafted. Backend development focuses on building the core functionality, including database management, user authentication, and integrating the AI API for itinerary generation. Frontend development connects the UI to the backend, creating a dynamic and interactive user experience. Thorough testing and deployment ensure the website is functional, user-friendly, and secure. Finally, ongoing maintenance and updates focus on bug fixes, feature enhancements, and security measures to keep Journee running smoothly and meeting user needs.

Chapter 2

Business Plan

2.1 Executive summary: Journee

Name of business	Journee
Legal form	General Partnership
Contact address	31-tahrir St. Cairo
Mobile	01067105055
Email	JourneeAITraveler@gmail.com
Type of business	Service provider

Table 2: Executive Summary

A. Brief description of the Business idea:

We provide an [AI-powered travel website](#) that Personalized itinerary generation tailored to budget, interests, and travel style. Curated recommendations for hotels, activities, and unique local experiences. Real-time itinerary adjustments in response to user feedback or unexpected events. Proactive budget tracking and recommendations to manage trip expenses.

B. Customers:

1. [Travelers seeking personalized experiences](#): This includes individuals tired of generic itineraries and craving trips tailored to their interests, budget, and travel style.

2. **Tech-savvy travelers:** Our website will resonate with those comfortable utilizing technology to research, plan, and book their vacations.
3. **Budget-conscious travelers:** Highlighting personalized budgeting tools and cost-effective recommendations will attract travelers aiming to maximize the value of their trip.

C. Owners:

1.

Name	Ahmed Labib
Address	Downtown, Cairo
Qualification	Strong Python and intermediate Django knowledge, experience with API integration
Function in business	Implement website logic and functionality, and handle communication with Vertex AI and other external APIs

Table 3: Owner1

2.

Name	Nada Mohamed
Address	Shubra, Cairo
Qualification	Understanding of financial modeling
Function in business	Market research and competitive analysis. Developing business plan and monetization strategies

Table 4: Owner2

3.

Name	Nour Mohsen
Address	6th October - Giza
Qualification	Strong Python and intermediate Django knowledge, Experience with testing and debugging
Function in business	Collaborate on database and website logic. Ensuring code quality and application robustness

Table 5: Owner3

4.

Name	Hagar Ahmed
Address	Al Marj- Cairo
Qualification	System and Requirements analysis
Function in business	Design UML and EERD diagrams. Translate user needs into technical specifications

Table 6: Owner4

5.

Name	Esraa Ahmed
Address	Faisal, Giza
Qualification	Great understanding of web basic stacks (HTML, CSS, JS). Good UI/UX design principles
Function in business	Create a visually appealing and intuitive website. Collaborating on interface design and functionality

Table 7: Owner5

6.

Name	Ali Ashraf
Address	Heliopolis- Cairo
Qualification	Great understanding of web basic stacks (HTML, CSS, JS). Good UI/UX design principles
Function in business	Create a visually appealing and intuitive website. Collaborating on interface design and functionality

Table 8: Owner6

Start-up capital

Investment	197,750
Working capital	672,000
Total	869,750

Source of capital

Partner	300,000
Family loan	569,750

2.2 vision & mission

A. Our vision:

We envision a world where travel planning is effortless and accessible for everyone.

B. Our mission:

Our mission is to revolutionize travel planning by leveraging the power of AI. We strive to:

- **Simplify** trip creation: Eliminate the overwhelming task of researching and coordinating travel by providing a user-friendly AI platform.
- **Personalize** every journey: Go beyond basic budgeting tools. Our AI analyzes motivations and habits to curate itineraries with hotels, flights, and activities that truly resonate with each traveler.
- **Democratize** travel: Make personalized travel experiences accessible to everyone, regardless of budget or travel expertise.
- **Maximize** travel value: Our AI optimizes trip planning, ensuring travelers get the most out of their budget and travel time.

2.3 business idea & market:

A. Description of the business idea:

We built an AI-powered travel website that targets customers who want trips tailored to their interests, budgets, and travel styles. Those who are at ease with technology will find the website appealing when it comes to researching, planning, and booking vacations. By offering personalized budgeting tools you can attract travelers who want to maximize the value of their trip.

Our services:

- Effortless Planning: Tell the budget, interests (culinary adventures, Historical deep dives), and travel style (relaxation or exploration), and our AI crafts the perfect itinerary.
- Proactive Expense Tracking: Stay on budget with real-time expense tracking and personalized recommendations. Our AI helps you maximize value throughout your trip.
- 24/7 Support: Have questions or need adjustments on the go? Our AI is accessible 24/7 to assist you throughout your journey.

Customers reaching:

- Social media marketing: Build a presence on popular platforms like Instagram, Facebook, and Pinterest. Share engaging content showcasing the benefits of your AI travel website. Use targeted advertising options to reach users interested in travel and exploration.
- Email Marketing: Build an email list and send targeted campaigns Those who have an interest in travel.
- Google ads

B. Description of the market:

The travel industry is undergoing a significant transformation, with personalization becoming the driving force. Travelers are ditching one-size-fits-all itineraries and seeking experiences that reflect their unique desires. This facilitates the process of spreading of the AI-powered travel website.

Objectives: Become a leading resource for travelers seeking AI-powered itinerary creation and trip planning tailored to their budget, interests, and travel style. We offer users a platform that goes beyond simple itinerary building.

Our goals:

SMART goals for our website:

Specific: Achieve 80% user satisfaction with the relevance and budget-friendliness of AI-generated itineraries.

Measurable: Conduct user surveys after itinerary generation to gauge satisfaction with personalization and budget adherence.

Achievable: Regularly refine the AI model with user data and travel industry trends to improve itinerary recommendations and with a focused marketing strategy.

Relevant: Highly personalized and budget-conscious itineraries are core differentiators for your website.

Time-bound: A 6-month timeframe allows for collecting sufficient user feedback and refining the AI model.

SWOT analysis:

In the following sections, we will explore each component of the SWOT analysis in the context of our AI-powered travel website, summarizing the important things to think about and the possible effects on the platform's expansion and success. We can understand the fundamental dynamics of the industry and set ourselves up for success in this dynamic sector by examining the industry's strengths, weaknesses, opportunities, and threats.

Strengths

- AI-Powered Personalization: Journee leverages AI to create customized itineraries, catering to individual preferences and budgets, a significant advantage in the competitive travel planning market
- Target Audience Alignment: Focusing on novice travelers with families aligns well with the large segment of the population seeking convenient and personalized travel solutions.
- Budget-Conscious Approach: Addressing the primary frustration of staying within budget directly resonates with many potential users.
- Accommodation Preference: Catering to the preference for hotels simplifies initial planning and partnerships.
- Trust in AI: A majority (74.7%) are open to AI-generated itineraries, indicating potential user acceptance.
- Desired Task Assistance: Journee's features directly address the desired tasks of booking flights/hotels at the best price and creating day-by-day itineraries.

Weakness

- Novel Technology: AI-based travel planning is still a relatively new concept, and some users might be hesitant or require education about its benefits.
- Dependence on AI: The system's accuracy and effectiveness rely heavily on the quality of AI algorithms and data, which can be susceptible to biases or errors.
- Limited Scope: Focusing primarily on hotel accommodations may exclude travelers who prefer alternative lodging options.
- Potential Lack of Flexibility: AI-generated itineraries might not account for spontaneous changes or unforeseen circumstances during travel.

Opportunities

- Market Growth: The travel industry is constantly expanding, presenting opportunities to reach a wider audience and cater to diverse travel interests.
- Technological Advancements: Continuous improvement in AI technology can enhance the accuracy and personalization of travel itineraries.
- Partnerships: Collaborations with airlines, hotels, and other travel service providers can offer exclusive deals and expand service offerings.
- Mobile Application: Developing a mobile app can increase accessibility and convenience for users on the go.
- Expanding Travel Themes: While initial focus is on relaxation, exploring other themes like adventure, culture, or food can attract diverse user segments.

Threats

- Competition: The online travel planning market is highly competitive, with established players and new entrants vying for market share.
- Data Privacy Concerns: Collecting and utilizing user data for personalization raises potential privacy concerns that need to be addressed transparently.
- Economic Fluctuations: Economic downturns or travel restrictions can negatively impact the travel industry and user demand.
- Technological Disruptions: Emergence of new technologies or platforms may pose challenges to Journee's current approach.

- Comment on strengths:

Plan smarter, travel easier. Journee combines top-notch service with an intuitive platform to make planning your dream trip a breeze. Our AI-powered recommendations go beyond the generic, creating personalized itineraries tailored to your budget and interests.

- Comment on weakness:

Being a new business means taking a fresh and creative approach to the market, even though there may be some challenges involved. Even if the initial clientele might not be well-established, effective marketing techniques and top-notch service might help.

- Comment on opportunities:

As the travel market grows up, we utilize the reach of social media to connect with travelers and share the transformative power of AI-powered travel planning. and with the future of AI, offering a seamless and personalized approach to facilitate the traveling process.

- Comments on threats:

New technologies like AI may raise questions, and online security remains a top concern for all travelers. Additionally, the travel industry is a competitive space. providing clear and transparent information about how our AI technology personalizes travel experiences. We implement robust security measures to safeguard user data, fostering a trusted environment. Additionally, we view competition as an opportunity to innovate.

C. Customer segmentation:

1. geographic:

customers can make & book trips online from any location to any location.

2. demographic:

age: people aged from 18 years old.

gender: females & males.

Occupation: travelers

3.Psychographic:

Social class: all social classes.

Lifestyle: People who are interested in traveling, trying new things and want to know a plan that fits their budget.

4. Behavioral segmentation:

Purchasing behavior: people who are interested in traveling, trust AI and have smartphones and laptops to make plans online.

Personality: love traveling, anyone who's interested in having a perfect plan.

Benefits expectations: Reliable and fast website, reliable information, and availability.

2.4 marketing plan:

A. Primary Market Research:

Understanding our users is critical to the success of Journee.

Survey:

This user survey documentation introduction emphasizes the role of the survey in informing various aspects of our project development. It highlights how gathered information will be used to shape user stories, optimize the user experience, prioritize features, and ultimately improve the effectiveness of the AI model integration. We have got 146 responses in our survey and here's our questions and the answers.

Age	Between 18 to 25
Budget	39.2% between 5000-9999 24.7% <4999
Travel experience	58.2% novice traveler
Travel group	Most of them travel with family by 71.9%
Travel inspiration source	55.5% get their travel inspiration from social media while 62.3% get their travel inspiration from friends and family
Biggest frustrations with travel planning	Most of them has biggest frustrations from staying budget by 67%
Your top 3 desired travel themes	Most of them prefer relaxation by 72.6%
Preferred accommodation type	71.9% prefer staying in hotels
Typical duration of stay	Between 4 to 7 nights
Would you trust an AI-generated itinerary (travel plan)?	74.7% said yes. 25.3% said no.
What specific tasks would you seek the help of an AI travel assistant to handle for you?	67% booking flights and hotels at best price while 56.9% creating a day-by-day itinerary

Table 9: Survey analysis

Insights gathered from the survey:

Young Travelers: Your primary audience is between 18-25 years old. This means our website design and marketing should resonate with this demographic's preferences for visual interfaces, social trends, and a desire for unique experiences.

Budget-Conscious: A significant portion of potential users have budgets under 5000 EGP. Journee should highlight cost-effective options and include features that help users track estimated expenses during their itinerary creation.

Focus on Newbies: Over half of respondents are novice travelers. This means providing extra guidance, resources, and addressing common traveler anxieties will be valuable.

Opportunities to Address User Needs:

Social Element: Since friends, family, and social media are primary inspiration sources, we should include easy itinerary sharing features.

Budgeting Tools: The top frustration is staying on budget. We can exploit this opportunity by offering AI-powered suggestions that help users monitor and control costs while planning.

Relaxation is Key: Tailor your suggested itineraries to the high demand for relaxation-themed travel. Provide options for wellness activities, scenic locations, and accommodations suited for unwinding.

AI Acceptance: Most users are open to AI-generated itineraries. Emphasize the convenience and customization that your AI system will provide.

Features for "Journee":

Price-focused Search: Allow users to search for destinations, accommodations, and flights with a strong emphasis on filtering by budget.

Integrated Cost Tracking: Display a real-time estimated trip cost activities and accommodations in the itinerary.

AI-Powered "Relaxation Mode": The AI could prioritize stress-free activities, slower-paced itineraries, and destinations known for tranquility.

B. Secondary market research:

1. The global travel and expense management software (budget managing) market size was valued at USD 6.9 billion in 2019 and is expected to grow at a compound annual growth rate (CAGR) of 12.4% from 2020 to 2027. Several businesses have been expanding internationally, leading to an increase in the volume of travel and related expenses.
2. An August 2022 survey asked travelers worldwide about their willingness to trust Artificial Intelligence (AI) for trip planning in 2033. Among the different travel aspects considered, accommodation planning had the highest share of respondents who would rely on AI, with 75 percent of survey participants indicating they would use it for that purpose.
3. the overall outlook for the travel industry:

The travel industry is projected to continue its recovery trend. The World Travel & Tourism Council (WTTC) estimates the sector could reach US\$9.5 trillion in 2023 and potentially exceed pre-pandemic levels in 2024. After years of restrictions, there's strong pent-up demand for leisure travel, fueling growth.

4. the factors might disrupt it:

A potential recession or economic slowdown could curb travel spending, particularly for discretionary trips. Ongoing conflicts, like the war in Ukraine, can create regional travel instability and impact fuel prices. Not to mention inflation, rising costs of travel components like fuel and accommodation could hinder affordability. Also staffing shortages in airlines, hotels, and other travel sectors could lead to delays and reduced service quality.

Marketing mix:

Marketing mix is the most fundamental concept of marketing which is a set of controllable marketing tools that a company uses to create a desired response in the targeted market. Marketing Mix' is also referred and known as the Four P's i.e. product, price, place of distribution, and promotion. Each element plays a crucial role in shaping the overall marketing strategy and ensuring the success of a service in the market.

1.service:

As previously stated, our web application will:

- Users simply input their desired destination, budget constraints, travel style preferences (adventure, cultural, family-friendly, etc.), and travel group composition (solo, couple, family with children's ages).
- Our advanced AI engine then generates personalized itinerary that adheres to user-defined budgets.
- This comprehensive plan includes flight options (travel class and layovers), accommodation suggestions, recommended sightseeing locations with estimated costs and opening hours, and restaurant suggestions with price ranges and cuisine types.

2.price:

Traveler's price: registration will be free and the price for each itinerary will be 5\$ after the free trial (one itinerary)

Pricing strategy:

One-time purchases strategy:

A one-time purchase model is a pricing strategy in which the customer pays a single upfront cost to buy a product or service. After the initial purchase, there are no further payments required.

- We Offer a free trial for new users to explore your website's functionalities.
- The free trial allows users to test the AI's capabilities and see if your website meets their travel planning needs.
- After the free trial, users can choose to pay for each itinerary they generate.

3.Place:

We will provide our services through an online web application.

4.Promotion:

Through

- Social media marketing: Build a presence on popular platforms like Instagram, Facebook, and Pinterest.
- Email Marketing: Build an email list and send targeted campaigns Those who have an interest in travel.
- Google ads

Our promotion startup will be 4 months containing our ad, link and data on Facebook, Instagram, Pinterest, google ads.

We'll be spending around 9k L.E (with an average CPE of 2.50 L. E which totals 3600 engagements) which could be scaled up every 4 months. We'll be anticipating a conversion rate of about 5% which would leave us with around 180 users.

Planned marketing campaign cost		
Ads	Ad /month	/4 months
Facebook	1500	6000
Instagram	1500	6000
Google ads	1500	6000
Pinterest	1500	6000
Reddit	1500	6000
Google ads	1500	6000
Total	9000	36000

Table 10: Marketing campaign cost

2.5 legal form

The legal form of the business will be [general partnership](#).

Reasons for choosing this legal form:

- General partnerships offer the flexibility to structure businesses however partners see fit. This gives those partners the ability to control operations more closely.
- Partnerships can offer tax advantages by allowing income to be taxed at the partner's individual tax rate, rather than corporate rates.
- The partnership helps in processing new and diversified ideas and brainstorming which support the continuity of productivity and increase the chances of success and survival.

- Having a business partner can allow you to share the financial burden for expenses and capital expenditures needed to run the business. This could help your business grow more quickly and be more competitive.
- General partnerships are defined by their ability to split management responsibilities between all partners. Whether this is an equal split or an agreed split between multiple partners, each party controls their share in the partnership.

2.6 Start-up Capital & its Sources

Investment	Amounts
Cash	90,000 L.E.
Software	13,750 L. E
Equipment	90,000 L.E.
Supplies	4,000 L.E.
Total Investment	197,750 L.E.
12 months of Salaries	672,000 L.E.
Total working Capital	672,000 L.E.
Total startup capital	869,750 L.E.

Table 11: Start-up capital

Specification of investment items:

Investment item	Specification	Price
Software	Figma + ChatGPT + VS code + Jira + Trello + Gemini advanced	11650+2100 L. E
Equipment	6 laptops	90,000 L.E
Supplies	Papers, pens, additional stationery	4000 L. E
Total acquisition cost		107,750 L. E

Table 12: items specifications

Sources of Start-Up Capital:

Type	Source	Conditions	Amount
Equity capital	Partners	50,000 of each partner with 0 interest.	300,000 L.E
Loan1	Family	4 years family Loan with 0% interest rate and the first instalment will be in the second year of operation.	569,750 L. E
Total funding			869,750 L. E

Table 13: Start-up capital sources

2.7 financial plan:

1. opening balance:

Assets	value	Liabilities	value
Fixed assets			
Equipment	90,000	Equity	300,000
Software	13,750	Family loan	569,750
Total fixed assets	103,750	Total long-term liabilities	869,750
Current assets			
Supplies	4,000	Total current liabilities	0
Cash and bank	762,000		
Total current assets	766,000		
Total assets	869,750	Total liabilities & Net worth	869,750

Table 14: Opening Balance

2. Monthly Sales Plan:

Itinerary generator	month	1	2	3	4	5	6	7	8	9	10	11	12
	Price	250	250	250	250	250	250	250	250	250	250	250	250
	Quantity	180	190	200	210	220	235	245	260	270	285	300	315
	Total	45000	47500	50000	52500	55000	58750	61250	65000	67500	71250	75000	78750

Table 15: Monthly sales

3. Monthly Operational Cost Plan:

Month		1	2	3	4	5	6	7	8	9	10	11	12
Itinerary generator	quantity	180	190	200	210	220	235	245	260	270	285	300	315
Marketing	cost	9000	9000	9000	9000								
Software	cost	1145	1145	1145	1145	1145	1145	1145	1145	1145	1145	1145	1145
Supplies	cost	333	333	333	333	333	333	333	333	333	333	333	333
Materials	total cost	10478	10478	10478	10478	1478	1478	1478	1478	1478	1478	1478	1478
+Staff	total cost	56000	56000	56000	56000	56000	56000	56000	56000	56000	56000	56000	56000
+Rent	total cost	1400	1400	1400	1400	1400	1400	4500	4500	4500	4500	4500	4500
=Operation	total cost	67878	67878	67878	67878	58878	58878	61978	61978	61978	61978	61978	61978
+Capital cost	interest	0	0	0	0	0	0	0	0	0	0	0	0
+Capital cost													
=	depreciation	1875	1875	1875	1875	1875	1875	1875	1875	1875	1875	1875	1875
	Total cost	69753	69753	69753	69753	60753	60753	63853	63853	63853	63853	63853	63853

Table 16: Monthly operational costs

4. Cash flow plan:

month	pre-operation	1	2	3	4	5	6	7	8	9	10	11	12
Cash at the beginning of the month	0	90000	65247	42994	23241	5988	235	(1768)	(4371)	(3224)	423	7820	18967
equity	300000												
loans	569750												
sales		45000	47500	50000	52500	55000	58750	61250	65000	67500	71250	75000	78750
any other		0	0	0	0	0	0	0	0	0	0	0	0
total cash in	869750	135000	112747	92994	75741	60988	58985	59482	60629	64276	71673	82820	97717
investment	197750												
operational cost		69753	69753	69753	69753	60753	60753	63853	63853	63853	63853	63853	63853
total cash out	197750	69753	69753	69753	69753	60753	60753	63853	63853	63853	63853	63853	63853
cash at the end of the month	672000	65247	42994	23241	5988	235	(1768)	(4371)	(3224)	423	7820	18967	33864

Table 17: Cash flow

5. Profit margin (monthly estimation of net profit):

month	1	2	3	4	5	6	7	8	9	10	11	12
Itinerary quantity	250	250	250	250	250	250	250	250	250	250	250	250
generator turnover	180	190	200	210	220	235	245	260	270	285	300	315
total sales	45000	47500	50000	52500	55000	58750	61250	65000	67500	71250	75000	78750
operation total cost	69753	69753	69753	69753	69753	60753	63853	63853	63853	63853	63853	63853
capital cost depreciation	0	0	0	0	0	0	0	0	0	0	0	0
total cost	67878	67878	67878	67878	67878	58878	58878	61978	61978	61978	61978	61978
profit before tax	(22878)	(20378)	(17878)	(15378)	(3878)	(128)	(728)	3022	5522	9272	13022	16772
income tax	22.50%											
profit after tax	(22878)	(20378)	(17878)	(15378)	(3878)	(128)	(728)	2342.05	4279.55	7185.8	10092.05	12998.3

Table 18: Profit margin

Chapter 3

system analysis

3.1Introduction

In Business, System analysis and design refers to the process of examining a business situation with the intent of improving it through better procedures and methods. system analysis and design relate to shaping organizations, improving performance and achieving objectives for profitability and growth. the emphasis is on system in action, the relationships among subsystems and their contribution to meeting a common goal.

Particularly, System analysis is the process of gathering and interpreting facts, identifying problems, and using the information to recommend improvements to the system, through system analysis you can:

- **Understand the System:** This involves gathering information about the system's purpose, its inputs and outputs, the activities it performs, and the people involved. Techniques like interviews, document reviews, and observation are used for this purpose.
- **Identify Needs:** System analysis focuses on identifying the needs of the stakeholders involved with the system. This could be the users of a software program, the employees in a business process, or any entity that interacts with the system.
- **Improve Efficiency:** A core objective of system analysis is to find ways to improve the system's efficiency and effectiveness. This could involve streamlining processes, eliminating bottlenecks, or recommending new technologies or procedures.

- **Define Requirements:** Through analysis, functional and non-functional requirements are defined. Functional requirements specify what the system should do, while non-functional requirements detail how the system should perform in terms of aspects like speed, security, and usability.

Once the business needs are set, planning and analysis begins. By following a systematic approach to analysis, you can gain valuable insights to optimize a system's performance and meet its intended objectives.

In this chapter we will discuss:

1. user requirements:
 - a. functional requirements
 - b. non- functional requirements
2. system requirements
3. system diagrams:
 - a. use case diagrams.
 - b. sequence diagrams.
 - c. activity diagram

3.2 User Requirements

1. functional requirements:

are specifications that detail what a system must do. They define the functionalities and features a system should possess to deliver its intended purpose

a. User Account Management:

- Users shall be able to create a secure account using an email and password.
- Users shall be able to log in to their existing accounts.
- Users shall be able to reset their password if forgotten.
- Users shall be able to edit their profile information (name, contact details, etc.).
- Users shall be able to view their past travel itineraries.

b. Travel Profile Creation:

- Users shall be able to select their preferred travel themes (e.g., history, food, adventure, relaxation).
- Users shall be able to indicate their travel style preferences (e.g., budget-conscious, luxury, family-friendly).
- Users shall be able to list their specific interests (e.g., museums, hiking, culinary experiences).
- Users shall have the option to provide information about any accessibility needs.
- Users shall be able to save their travel profiles for use in planning future trips.

c. Trip Planning:

- Users shall be able to search for destinations by country, city, or specific attractions.
- Users shall be able to input their desired travel dates.
- Users shall be able to specify their overall trip budget.
- Users shall be able to set budget allocations for categories like accommodation, transportation, and activities.
- Users shall be able to initiate the itinerary generation process.

2. non-functional requirements:

These focus on how well the system performs, addressing aspects like usability, security, and performance

a. Performance:

- The website shall load pages within 3 seconds on average.
- Search results shall be displayed within 2 seconds.
- Itinerary generation shall take no longer than 30 seconds for typical trip durations.

b. Usability:

- The interface shall have a clean and intuitive design.
- Navigation between different sections of the website should be clear and logical.
- Input forms shall have clear guidance and error prevention.
- The itinerary presentation format shall be visually appealing and easy to understand.
- The website shall be responsive for use across desktop, tablet, and mobile devices.

c. Security:

- User passwords shall be stored securely using encryption.
- The website shall have measures to protect against common web vulnerabilities (e.g., SQL injection attacks).
- Any financial transactions (future feature) shall be processed using secure payment gateways.

d. Reliability:

- The system shall have an uptime of at least 99%.
- Scheduled maintenance shall be clearly communicated to users.

3.3 System Requirements

Hardware:

Journee is designed to be compatible with a wide range of devices, from [desktops](#) and [laptops](#) to [tablets](#) and [smartphones](#). To ensure a seamless experience, a device with at least [2GB of RAM](#) is recommended. Since Journee doesn't require high-end hardware, it makes it naturally accessible to a broad range of users.

Software:

Journee is built to work with the most recent versions of popular web browsers. This includes [Google Chrome](#), [Mozilla Firefox](#), [Microsoft Edge](#), and [Safari](#).

Network:

A stable internet connection is essential to access Journee's features and functionalities. [Broadband](#) is recommended for optimal performance. However, Journee is designed to be efficient, and even users with moderate internet speeds should be able to enjoy the core functionalities of the platform.

3.4 System Diagrams

1. Use case diagram:

A Use Case Diagram is a visual representation of the interactions between users (actors) and the system. It depicts the functionalities (use cases) that the system offers and how different user types interact with them.

UML use case diagrams are ideal for:

- Representing the goals of system-user interactions
- Defining and organizing functional requirements in a system
- Specifying the context and requirements of a system
- Modeling the basic flow of events in a use case

Overall, the use cases represent the functional requirements or the “what” of the system.

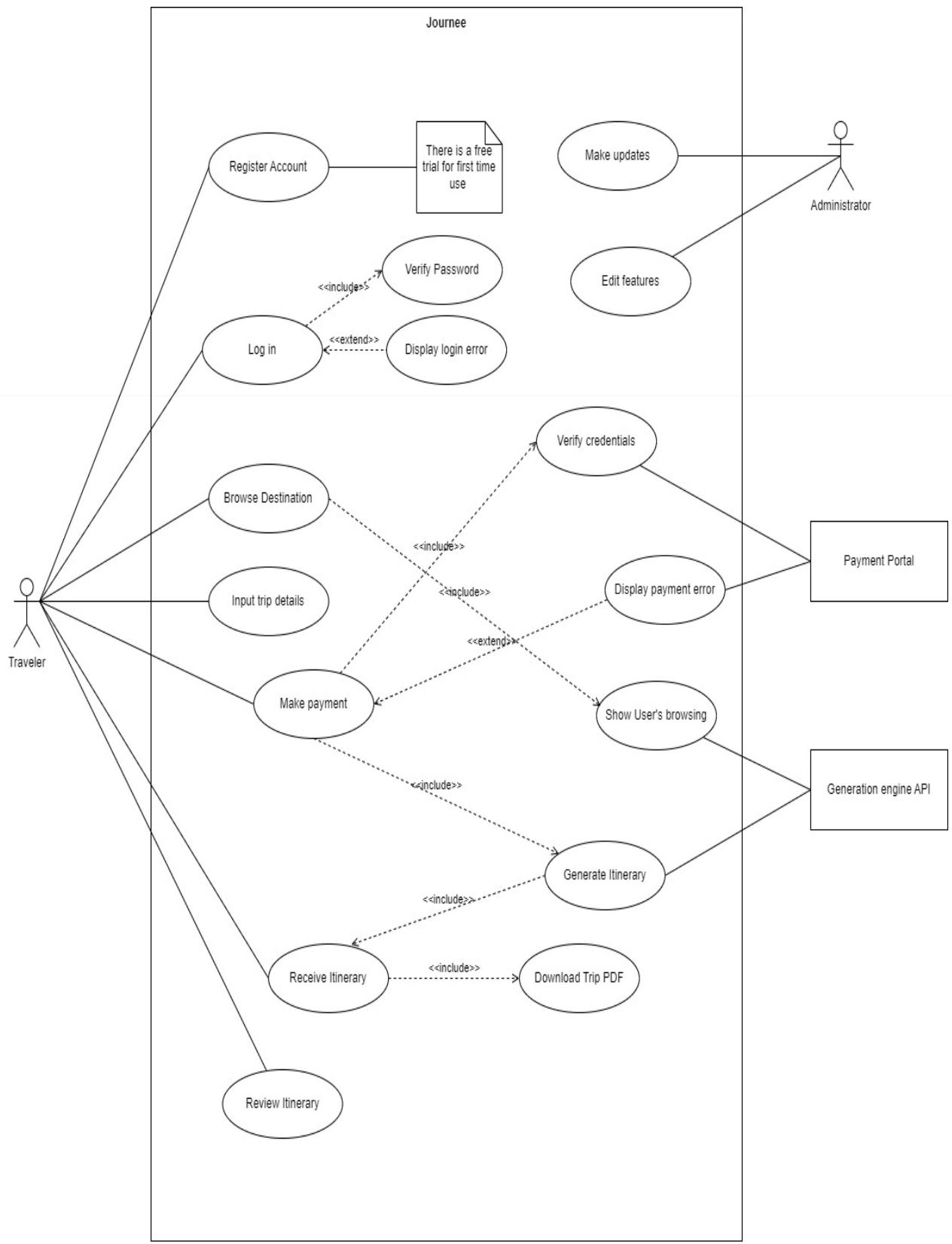


Fig.8: Use-case diagram



Fig.9: User Use-case diagram

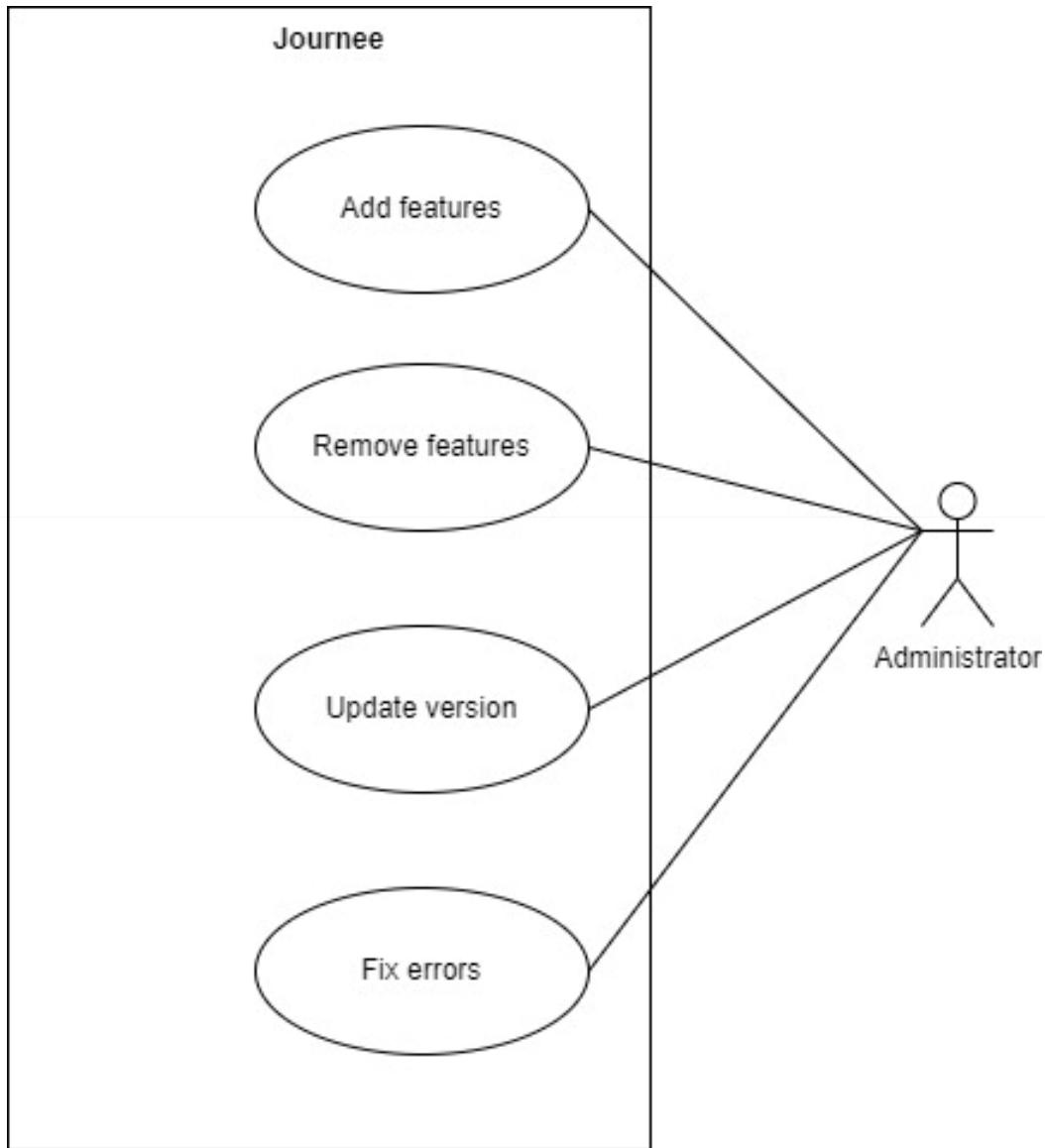


Fig.10: Admin Use-case diagram

2.Sequence diagram:

illustrates the interactions between different objects or processes in a system arranged in a time sequence. It visually depicts the messages exchanged between these objects chronologically, allowing you to understand the flow of communication within a specific scenario.

Sequence diagrams can be useful references for businesses and other organizations.

Try drawing a sequence diagram to:

- **Represent** the details of a UML use case.
- **Model** the logic of a sophisticated procedure, function, or operation.
- See how objects and components **interact** with each other to complete a process.
- **Plan** and **understand** the detailed functionality of an existing or future scenario.

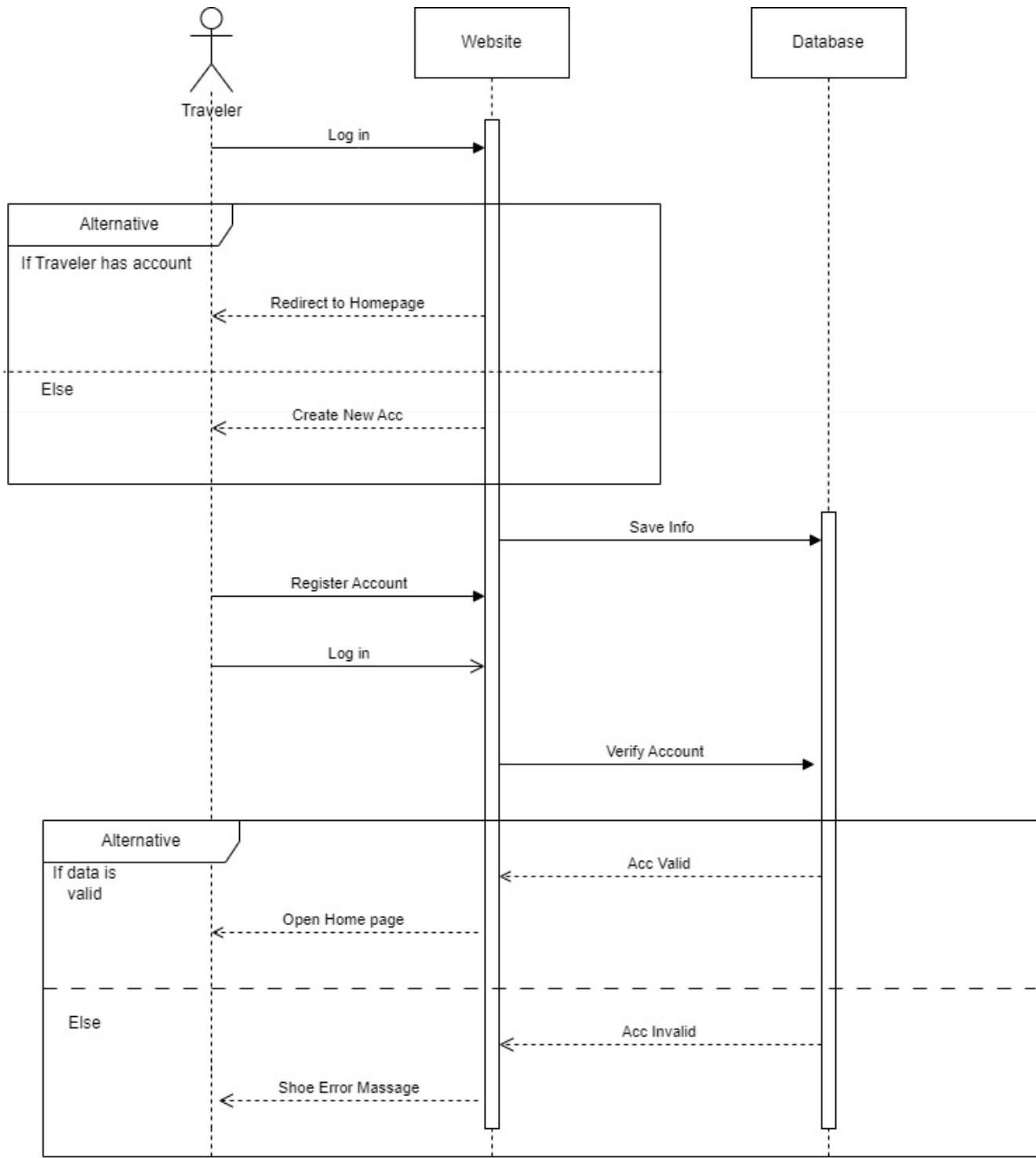


Fig.11: Register-Login Sequence

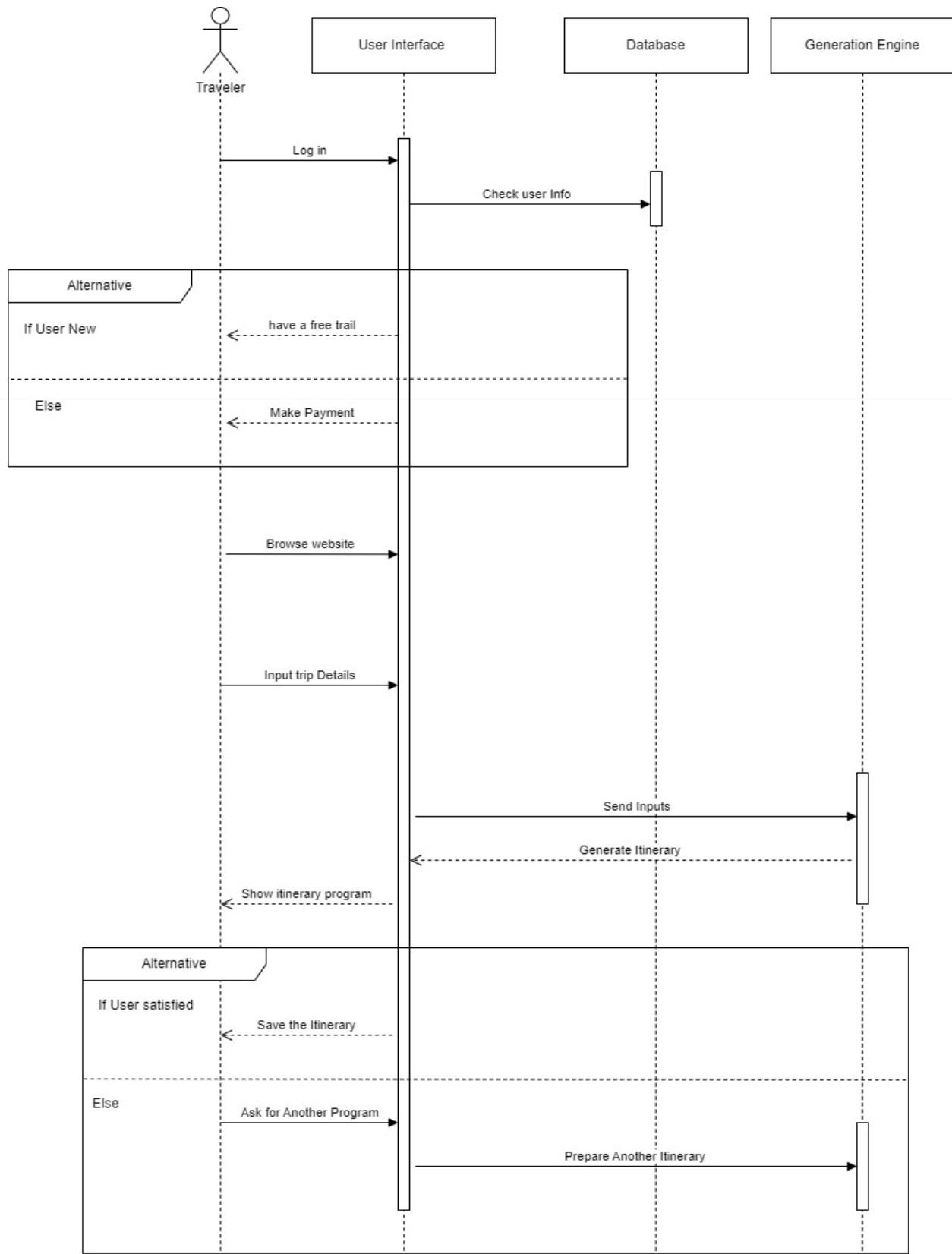


Fig.12: Itinerary-generation sequence

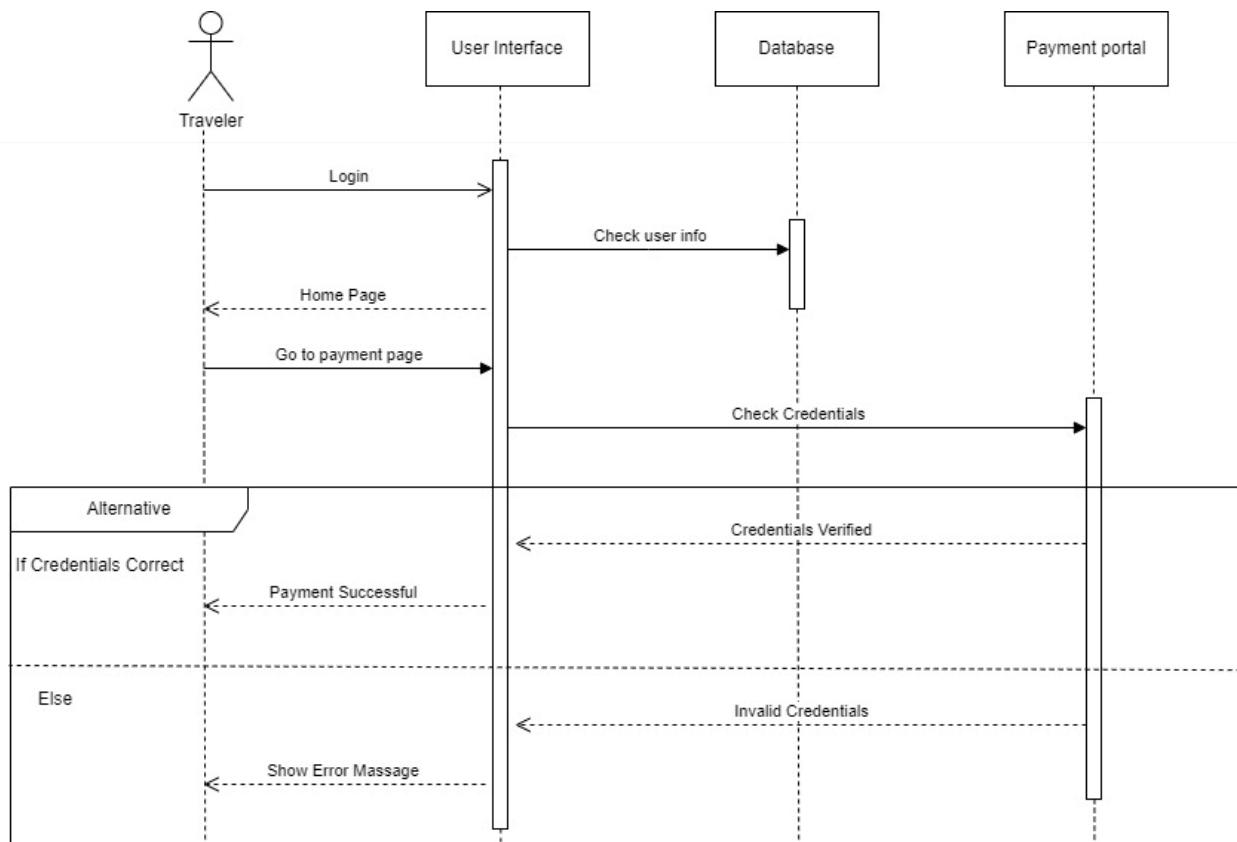


Fig.13: Payment sequence

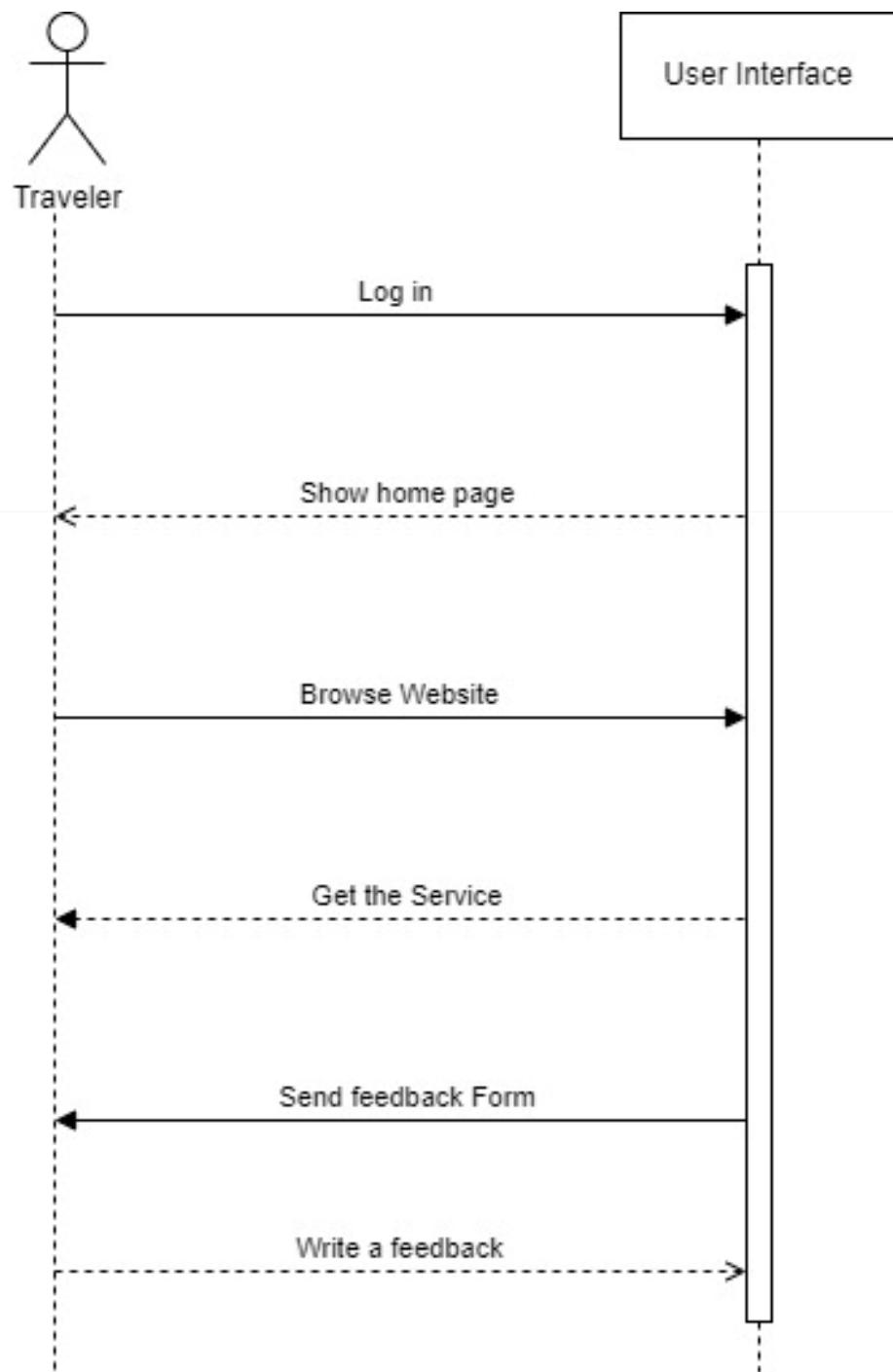


Fig.14: Review sequence

3. Activity diagram:

visually presents a series of actions or flow of control in a system like a flowchart or a data flow diagram. Activity diagrams are often used in business process modeling. They can also describe the steps in a use case diagram. Activities modeled on can be sequential and concurrent.

Activity diagrams are used to:

- Demonstrate the logic of an algorithm.
- Describe the steps performed in a UML use case.
- Illustrate a business process or workflow between users and the system.
- Simplify and improve any process by clarifying complicated use cases.
- Model software architecture elements, such as method, function, and operation.

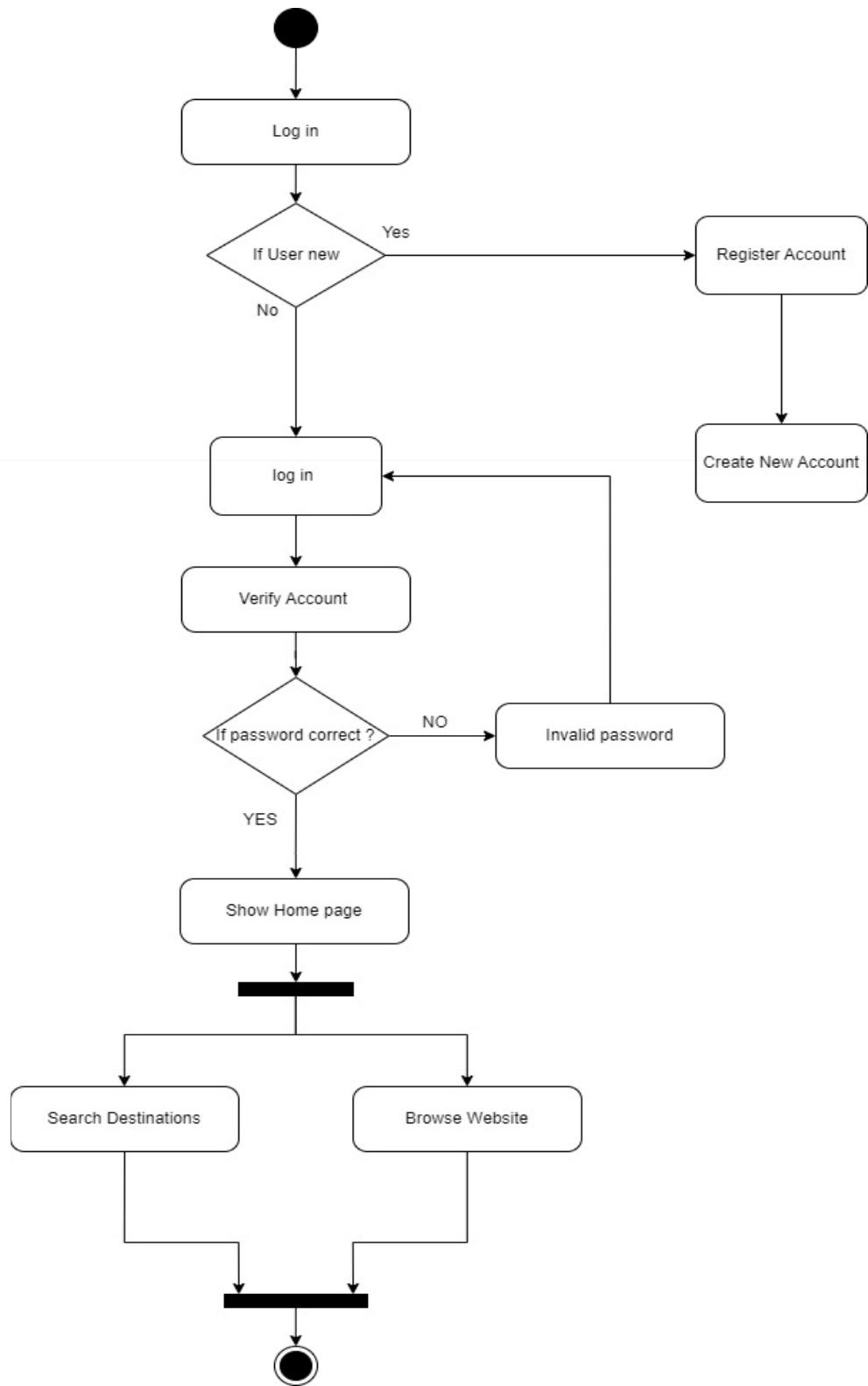


Fig.15: Login activity

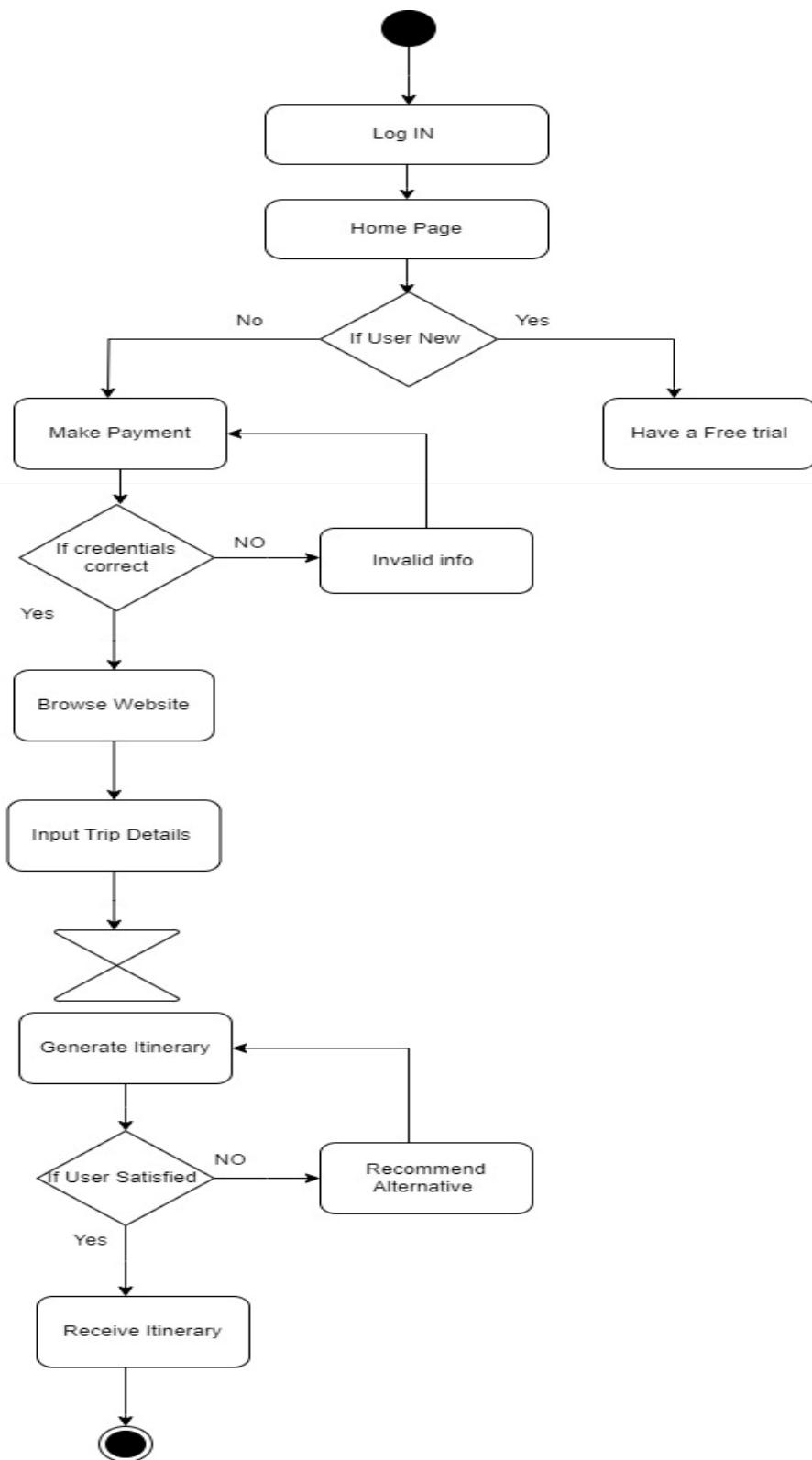


Fig.16: Itinerary generation activity

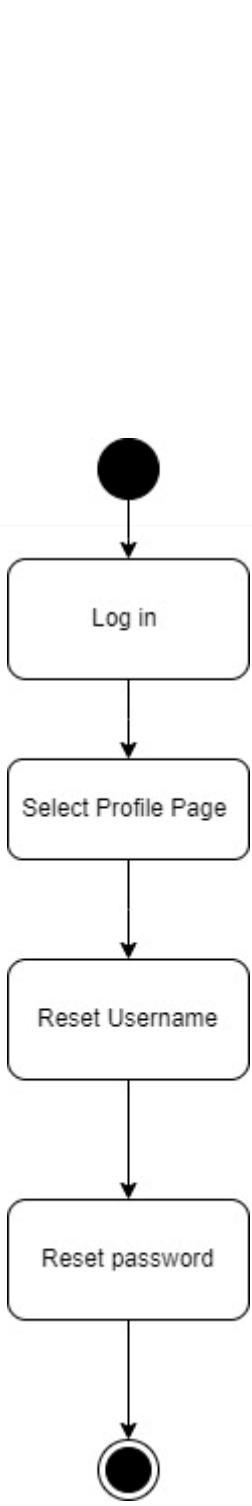


Fig.17: User account management activity

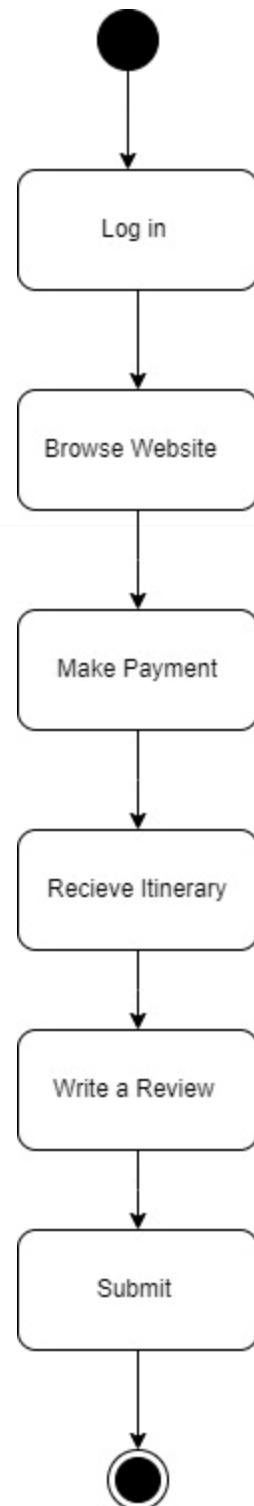


Fig.18: Review activity

Chapter 4

System Design

4.1 Introduction

This chapter delves into the detailed design of the Journee system, outlining the internal structure and relationships between its components. We'll explore the foundational elements that underpin the system's functionality and data management. This includes the [Entity-Relationship Diagram \(ERD\)](#), which visually depicts the relationships between data entities and their attributes, ensuring a well-structured and organized database. We'll also present a [Class Diagram](#), illustrating the classes within the system and their interactions, providing a blueprint for code development and ensuring a clear understanding of object relationships. Finally, we'll examine [database mapping](#), which defines the connection between database tables and the objects within the system, enabling smooth data exchange and efficient data manipulation. This chapter serves as a comprehensive guide to the system's design, laying the groundwork for a robust and scalable implementation.

4.2 Class diagram

The class diagram provides a visual representation of the classes that make up the Journee system and their relationships. It serves as a blueprint for the object-oriented structure of the application, fostering code organization and maintainability. This diagram helps us understand:

- [Classes](#): The primary components of the system, each representing a distinct entity or concept. For example, "Traveler," "Plan," "Accommodation," and "Activities" are classes that encapsulate data and operations related to these respective entities.

- **Attributes:** Properties that define the characteristics of each class. For instance, the "Plan" class has attributes like "duration", "budget", and "destination" which store specific information about each plan.
- **Operations:** Methods or functions that define the actions and behaviors of each class. These operations allow objects of the class to interact with each other and with the system. For example, the "Traveler" class might have operations like "register," "login," and "makePayment" to manage the traveler's account and booking activities.
- **Relationships:** Connections between classes that show how they interact with each other. These relationships can be of different types, such as association, aggregation, and inheritance. For example, the "Plan" class might have an association relationship with the "Accommodation" and "Activities" classes, indicating that a plan can include multiple accommodations and activities.

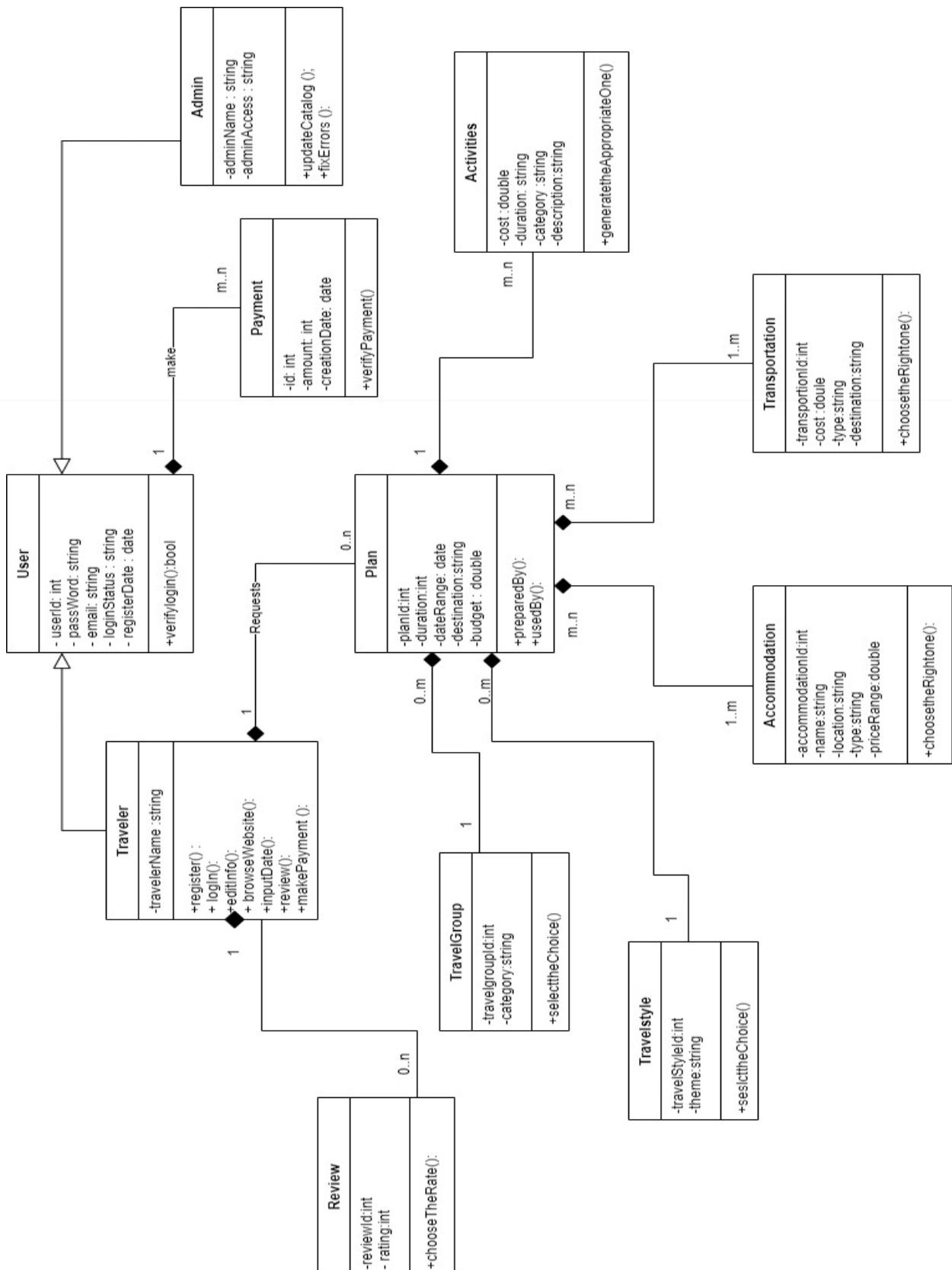


Fig.19: Class diagram

4.3 Database structure (ERD)

The Entity-Relationship Diagram (ERD) is a visual representation of the database schema for Journee. It depicts the entities (tables) and their relationships within the system. The ERD ensures the logical organization of data, promoting data integrity and efficient retrieval.

Here's a breakdown of the key elements within the ERD:

Entities: Entities represent the fundamental building blocks of the database. Each entity corresponds to a table, storing specific information about a particular type of object. For example, "User," "Plan," "Accommodation," and "Activities" are entities that hold data about users, travel itineraries, accommodations, and activities, respectively.

Attributes: Each entity has attributes, representing the specific data fields that hold information about that entity. For instance, the "User" entity might have attributes like "UserID," "Username," "Password," and "Email," which store data about individual users.

Relationships: Relationships define how entities are connected to each other. These relationships represent how different tables interact and share data. For instance, a "Plan" entity might have a "Has" relationship with an "Accommodation" entity, indicating that a plan can include multiple accommodations. The ERD depicts the type of relationship (e.g., one-to-many, many-to-many) and the cardinality (the number of instances involved in the relationship).

Keys: Primary keys are unique identifiers for each entity (table), ensuring that each row within the table can be uniquely identified. Foreign keys are used to establish relationships between tables, linking data from one table to another.

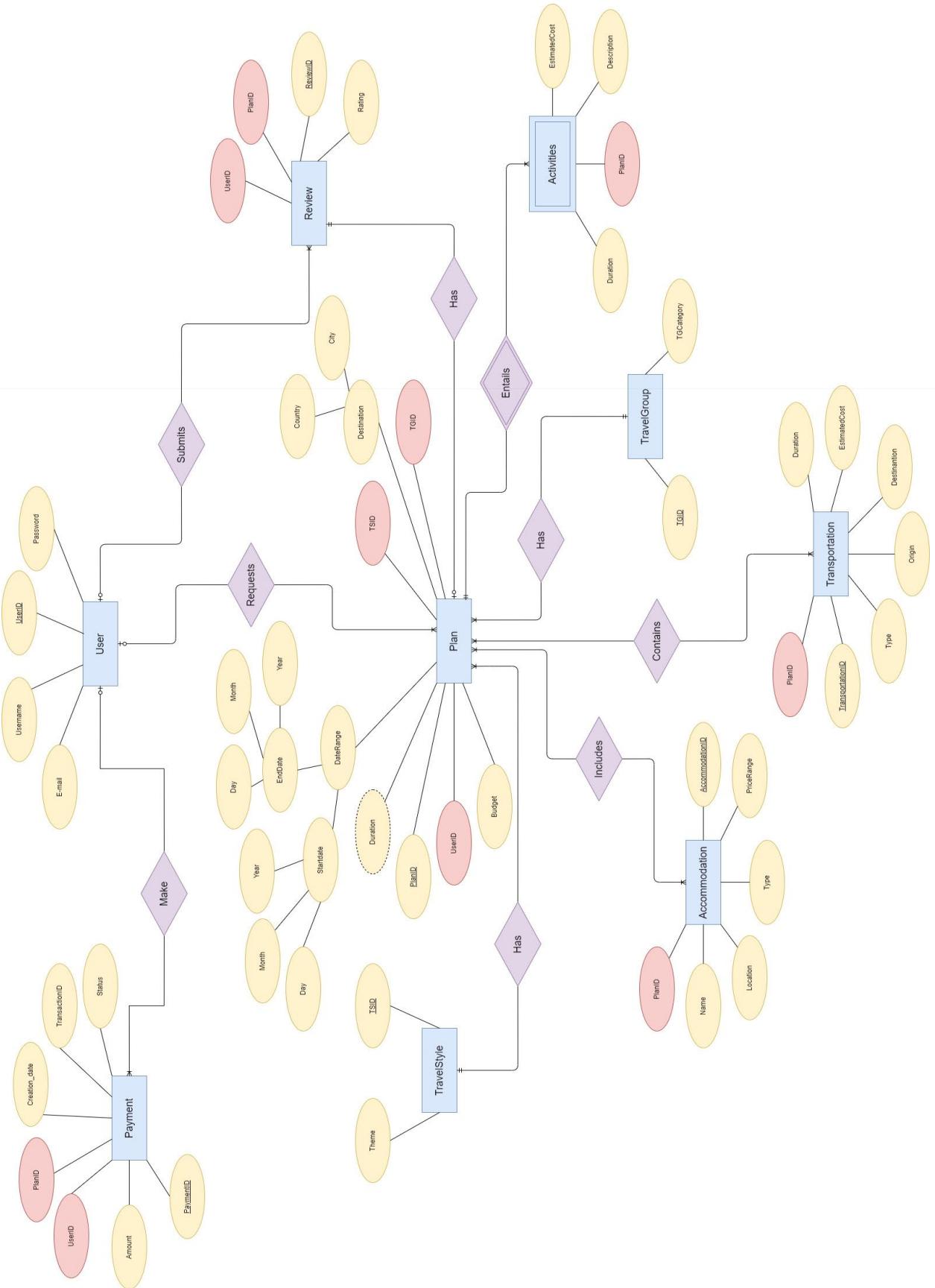


Fig.20: ERD

4.4 Database mapping

Database mapping is the process of connecting the structured data stored in a database to the objects and classes used in application code. It acts as a bridge between the database schema (defined in the ERD) and the object-oriented structure of the application (represented in the class diagram).

For our Journee website, database mapping ensures that data is seamlessly exchanged between the database and the Python code (using Django) that powers the website. Imagine a "Plan" entity in the database. This entity might have attributes like "PlanID," "Destination," "Budget," and "Duration." In the Python code, there would be a corresponding "Plan" class with similar attributes. Database mapping defines how each attribute in the database table is mapped to a corresponding attribute in the Python class. This allows the website to retrieve data from the database, use it to create objects in Python, manipulate those objects, and then update the database with any changes made to those objects. This process ensures that data is consistently stored, accessed, and updated, facilitating a smooth and efficient user experience.

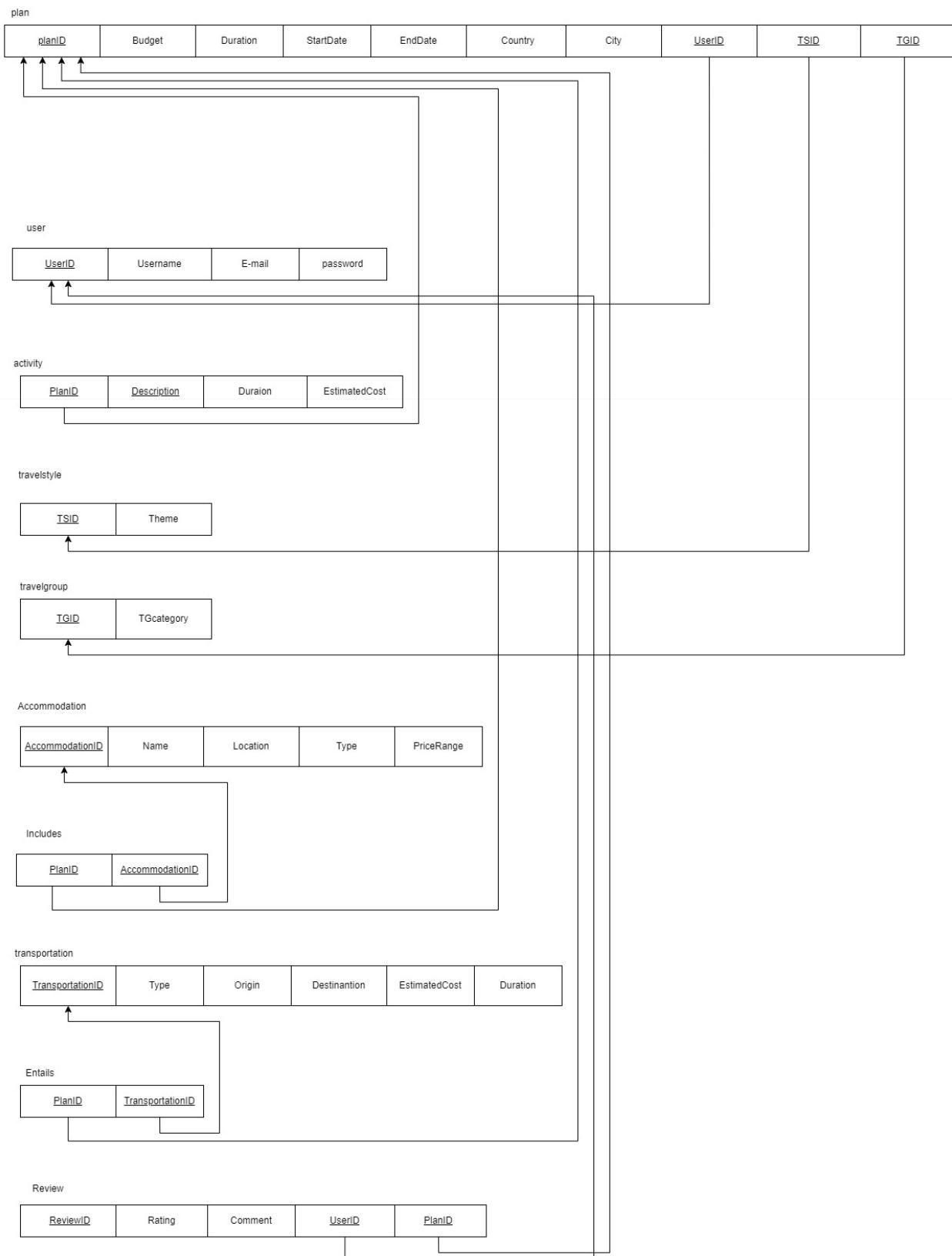


Fig.21: Database mapping

Chapter 5

System

Implementation

5.1 Introduction

This chapter outlines the implementation of the Journee system, bringing the design outlined in previous chapters to life. We will delve into the technologies and methodologies employed to build a functional and user-friendly travel planning platform.

The implementation process involves translating the design specifications into tangible code, utilizing various programming languages, frameworks, and tools. Specifically, we will focus on:

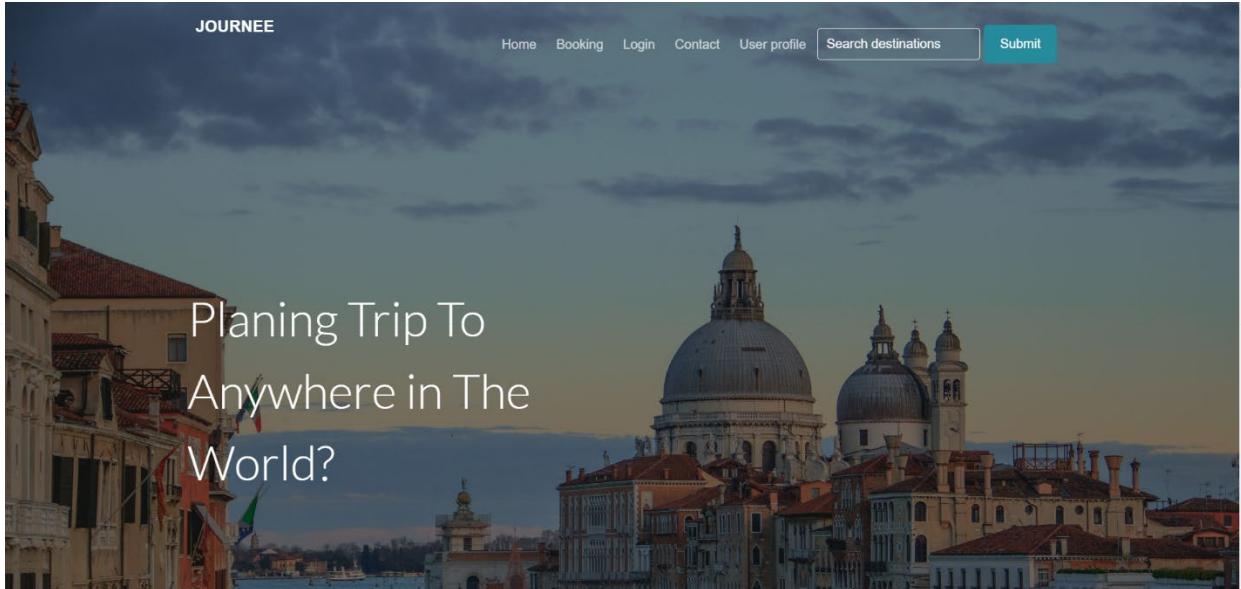
- **Frontend Development:** Utilizing HTML, CSS, and JavaScript to create the interactive and visually appealing user interface of the Journee website. This involves designing the layout, styling elements, and implementing interactive features that allow users to navigate the site, input travel preferences, and view generated itineraries.
- **Backend Development:** Employing Python and the Django framework to build the server-side logic that powers Journee. This involves handling user requests, interacting with the database to retrieve and store data, processing travel preferences, utilizing Google Gemeni AI API to generate personalized itineraries, and delivering responses to the frontend.
- **Database Integration:** Implementing a robust database system using SQLite to store user data, travel preferences, generated itineraries, and other relevant information. We will leverage SQL and Django's ORM to interact with the database, ensuring data integrity and efficiency.
- **API Integration:** Implementing communication with external APIs like Google Gemeni AI to leverage their powerful capabilities for itinerary generation. This involves sending requests to these APIs with user-specific

travel preferences and receiving responses in a string markdown-based format that can be processed and displayed on the Journee website.

- **Testing and Deployment:** Rigorously testing the functionality and performance of the system to ensure it meets the specified requirements. Once successful, we will deploy the Journee platform to a suitable hosting environment, making it accessible to users.

This chapter will provide an overview of the key components and processes involved in the implementation phase, laying the groundwork for a comprehensive and insightful description of the development journey.

5.2 System Screenshots



Homepage

A screenshot of a travel destination comparison section. The title "Most Popular Destination" is at the top, followed by the subtext "Choose from the best of the best.". Three destination cards are shown: New York, USA; Seoul, South Korea; and Paris, France. Each card includes a thumbnail image, the destination name, a brief description, and a "Schedule a Trip" button. A small "127.0.0.1:8000" is visible in the bottom left corner.

How It Works

It really is rather simple!

1

Input trip details

You first input your trip details and customize it according to your wants

2

Plan generation

Our AI generation engine will get to work by processing your input and making your plan

3

receive your plan

You receive your plan consisting of 3 section formatted in markdown where you can download it!

Our Success

We are really humble :)

196

DESTINATION

500

HOTELS IN PLAN

200

TRAVELERS

199

HAPPY CUSTOMER

ABOUT US

an AI-powered innovative web platform that leverages the power of artificial intelligence to create personalized travel itineraries tailored to each user's unique preferences, budget, and desired destinations. Journee aims to streamline the travel planning process by automating itinerary creation, allowing users to focus on the excitement of discovering new places and experiences and spare its customers the hustle and bustle of planning ahead and empowers them with creating a full itinerary with a click of a button.

DESTINATION

Europe
Australia
Asia
Canada
Dubai

HOTELS

Luxe Hotel
Italy 5 Star hotel
Dubai Hotel
Deluxe Hotel
BoraBora Hotel

GET IN TOUCH

01067105055
JourneeAI@gmail.com
Live Chat

Trip Form

Plan Your Trip

Budget

Travel Style
 Select your travel theme

Travel Group
 Select your travel group

Destination

Country of origin

Start Date Travel
 dd----yyyy

End Date Travel
 dd----yyyy

Plan Your Trip

Budget
 90000

Travel Style
 Culture

Travel Group
 Solo

Destination
 Germany, berlin

Country of origin
 Egypt

Start Date Travel
 25-Jun-2024

End Date Travel
 29-Jun-2024

Signup and login

The image displays three vertically stacked wireframe prototypes of a login/signup interface, each featuring a central modal window over a dark background.

Top Prototype: Shows a central modal with two sections: "LOGIN" on the left and "SIGN UP" on the right. Both sections contain placeholder text ("Lorem ipsum dolor sit amet, consectetur adipiscing elit.") and a button ("LOGIN" for the left, "SIGN UP" for the right).

Middle Prototype: Shows a central modal with two sections: "LOGIN" on the left and "SIGN UP" on the right. The "SIGN UP" section is expanded, showing four input fields labeled "username", "email", "password", and "confirm password".

Bottom Prototype: Shows a central modal with two sections: "LOGIN" on the left and "SIGN UP" on the right. The "LOGIN" section contains two input fields labeled "username" and "Password", and a "LOGIN" button. The "SIGN UP" section contains placeholder text ("Lorem ipsum dolor sit amet, consectetur adipiscing elit.") and a "SIGN UP" button.

Program output

Here's your program

Activities

Itinerary for Berlin Trip (Solo Traveler, Culture Enthusiast)

Day 1 (Tuesday, June 25th):

- **Morning:** Check into your hotel and leave your luggage.
- **Activity:** Explore Museum Island () Spend the morning wandering through the museums on this UNESCO World Heritage Site. Tickets cost around €18 (\$20.44). Take the U-Bahn (underground train) to Museumsinsel station (Zones AB), costing €3.40 (\$3.87) for a single ticket.
- **Afternoon:** Take a walking tour of the Brandenburg Gate () and surrounding area. Tours typically last 2-3 hours and cost around €15 (\$17.08).
- **Evening:** Enjoy a traditional German dinner in Mitte, Berlin's historic center. Expect to spend around €20 (\$22.77) for dinner.

Estimated Costs for Day 1:

- Museum Island Tickets: €18 (\$20.44)
- U-Bahn Tickets: €3.40 (\$3.87)
- Walking Tour: €15 (\$17.08)
- Dinner: €20 (\$22.77)
- **Total:** €56.80 (\$64.16)

Day 2 (Wednesday, June 26th):

- **Morning:** Visit the Reichstag Building () for a free tour of the parliament building and its famous dome. Tickets can be booked online in advance. Take the U-Bahn to Bundestag station (Zones AB), costing €3.40 (\$3.87) for a single ticket.
- **Afternoon:** Explore the remnants of the Berlin Wall at the East Side Gallery (), an open-air gallery featuring murals painted on the remaining sections of the wall. The
- **Afternoon:** Head to the airport for your departure flight back to Egypt.

Total Estimated Costs for Activities (excluding flights and accommodation):

- Day 1: €56.80 (\$64.16)
- Day 2: €26.80 (\$30.51)
- Day 3:

Flight

- SWISS LX239 LX966 departing at 3:10 AM, arriving at 1:55 PM, priced from EGP 19,742 (round trip) will take 11 hours and 45 minutes with a layover in Zürich (ZRH) for 6 hours.
- Austrian OS866 OS233 departing at 3:40 AM, arriving at 5:45 PM, priced from EGP 20,073 (round trip) will take 15 hours and 5 minutes with a layover in Vienna (VIE) for 10 hours and 20 minutes.
- Lufthansa LH587 LH1942 departing at 3:25 AM, arriving at 3:05 PM, priced from EGP 20,277 (round trip) will take 12 hours and 40 minutes with a layover in Munich (MUC) for 7 hours and 35 minutes.
- EgyptAir MS731 departing at 11:35 AM, arriving at 2:45 PM, priced from EGP 23,891 (round trip) will take 4 hours and 10 minutes (non-stop).
- Lufthansa LH585 LH190 departing at 5:20 AM, arriving at 4:55 PM, priced from EGP 20,709 (round trip) will take 12 hours and 35 minutes with a layover in Frankfurt (FRA) for 6 hours and 55 minutes.

Accommodation

- [Berlin Marriott Hotel](#) - EGP 12,223 per night
- [Melia Berlin](#) - EGP 12,032 per night
- [Hotel NH Collection Berlin Mitte Friedrichstrasse](#) - EGP 11,839 per night
- [Grand Hyatt Berlin](#) - EGP 18,242 per night
- [The Ritz-Carlton, Berlin](#) - EGP 20,482 per night

[Download Program](#)

Contact

The screenshot shows a contact form on a website. At the top left, the word "JOURNEE" is displayed in white capital letters. On the right side, there is a navigation bar with links: "Home", "booking", "Login", "Sign Up", and "Contact". Below the header, a large banner features a dark blue and orange cloudy background. In the center of the banner, the words "DON'T BE SHY" are written in small white capital letters, followed by the large, bold, white text "Get In Touch". The main content area has a brown background. On the left, there's a "Get In Touch" section containing four input fields: "Your firstname", "Your email address", "Your subject of this message", and a larger text area for "Write us something". A teal button at the bottom of this section says "Send Message". On the right, there's a "Contact Information" section with icons and text: a location pin for "31 Mohamed Mahmoud st. Downtown Cairo", a phone icon for "01067105055", an envelope icon for "JourneeAl@gmail.com", and a globe icon for "Journee.com". A small upward-pointing arrow is located in the bottom right corner of the page.

JOURNEE

Home booking Login Sign Up Contact

DON'T BE SHY

Get In Touch

Get In Touch

Your firstname

Your email address

Your subject of this message

Write us something

Send Message

Contact Information

31 Mohamed Mahmoud st.
Downtown Cairo

01067105055

JourneeAl@gmail.com

Journee.com

Admin dashboard

The screenshot shows the Django Admin dashboard. At the top, there's a header bar with "Django administration" on the left and "WELCOME, AHMEDLABIB" with links for "VIEW SITE / CHANGE PASSWORD / LOG OUT" on the right. Below the header, the main content area has two main sections:

- AUTHENTICATION AND AUTHORIZATION**: Contains "Groups" and "Users" with "Add" and "Change" buttons.
- TRAVEL_PLAN**: Contains "Accommodations", "Activity", "Plans", "Reviews", "Transportations", "Travel groups", and "Travel styles" with "Add" and "Change" buttons.

On the right side of the dashboard, there are two panels:

- Recent actions**: A list of recent activities, each with a green plus sign and a brief description.
- My actions**: A list of specific actions taken by the user, such as "Activity Island Hopping and Snorkeling", "Activity Table Mountain Hike and Waterfront Exploration", "Activity Island Exploration and Water Activities", and "Activity Visit the City Palace and Jagdish Temple".

User table

The screenshot shows the SQLite Viewer application displaying the contents of the "auth_user" table from the "JOURNEE" database. The table has 15 rows and 9 columns. The columns are: last_login, is_superuser, username, last_name, email, is_staff, and is_active. The data is as follows:

	last_login	is_superuser	username	last_name	email	is_staff	is_active
1	2024-06-21 15:11:43.443124	0	hagora	--	hagariragy@gmail.com	0	1
2	2024-06-22 22:14:02.456913	1	ahmedlabib	--	aa535399@gmail.com	1	1
3	--	0	medo	--	medo123@gmail.com	0	1
4	--	0	hamo	--	hamo@gmail.com	0	1
5	--	0	sam	--	sam5555@gmail.com	0	1
6	--	0	sarsora	--	soso@gmail.com	0	1
7	--	0	abohamed	--	ahmed@gmail.com	0	1
8	--	0	monomon	--	monaali@gmail.com	0	1
9	--	0	emoo	--	eman@gmail.com	0	1
10	--	0	yossef nader	--	yossefnader34@gmail.com	0	1
11	--	0	weso	--	wesam@gmail.com	0	1
12	--	0	shisho	--	shalimaa@gmail.com	0	1
13	--	0	dalida	--	dalida123@gmail.com	0	1
14	--	0	hoda mohamed	--	hoda123@gmail.com	0	1
15	--	0	hanan ibrahim	--	hananibrahim3@gmail.com	0	1

Travel style table

Backend > JOURNEE > db.sqlite3

Tables: auth_group, auth_group_permissions, auth_permission, auth_user, auth_user_groups, auth_user_user_permissions, django_admin_log, django_content_type, django_migrations, django_session, sqlite_sequence, travel_plan_accommodation, travel_plan_activity, travel_plan_payment, travel_plan_plan, travel_plan_review, travel_plan_transportation, travel_plan_travelpgroup, travel_plan_travestyle

Rows: 7

TSID	Theme
1	Adventure
2	Relaxation
3	Culture
4	Food-Tours (Culinary)
5	Religious-Pilgrimage
6	Eco-tourism
7	Nature exploration

Travel group table

Backend > JOURNEE > db.sqlite3

Tables: auth_group, auth_group_permissions, auth_permission, auth_user, auth_user_groups, auth_user_user_permissions, django_admin_log, django_content_type, django_migrations, django_session, sqlite_sequence, travel_plan_accommodation, travel_plan_activity, travel_plan_payment, travel_plan_plan, travel_plan_review, travel_plan_transportation, travel_plan_travelpgroup, travel_plan_travestyle

Rows: 20

PlanID	destination_c	destination_city	start_date	end_date	budget	user_id	travel_c	travel_g
1	1	USA	New York City	2024-04-15	2024-04-15	2000	9	4
2	2	India	Udaipur	2024-10-28	2024-11-04	1500	27	2
3	3	Sweden	Jukkasjärvi	2024-02-18	2024-02-25	1000	4	3
4	4	China	Hong Kong	2024-07-01	2024-07-08	1800	15	1
5	5	Canada	Lake Louise	2024-08-10	2024-08-17	1200	19	2
6	6	Peru	Iquitos	2024-05-20	2024-05-27	800	11	3
7	7	USA	Maui	2024-12-20	2024-12-27	2500	25	4
8	8	Tanzania	Serengeti National Park	2024-06-05	2024-06-12	2000	26	3
9	9	United Arab Emirates	Dubai	2024-03-10	2024-03-17	3000	20	4
10	10	Maldives	Baa Atoll	2024-11-01	2024-11-08	4000	3	4
11	11	Italy	Venice	2024-09-15	2024-09-22	1500	28	3
12	12	USA	Grand Canyon Village	2024-04-01	2024-04-08	800	6	2
13	13	Monaco	Monte Carlo	2024-07-20	2024-07-27	2000	17	4
14	14	Indonesia	Nusa Dua, Bali	2024-01-10	2024-01-17	1800	16	1
15	15	Kenya	Nairobi	2024-06-15	2024-06-22	1200	18	3

Plan table

Backend > JOURNEE > db.sqlite3

Tables: auth_group, auth_group_permissions, auth_permission, auth_user, auth_user_groups, auth_user_user_permissions, django_admin_log, django_content_type, django_migrations, django_session, sqlite_sequence, travel_plan_accommodation, travel_plan_activity, travel_plan_payment, travel_plan_plan, travel_plan_review, travel_plan_transportation, travel_plan_travelpgroup, travel_plan_travestyle

Rows: 4

TGID	TGCategory
1	Solo
2	Family
3	Friends
4	Couple

Activity table

SQLite Viewer screenshot showing the Activity table. The table has columns: id, Duration, PlanID, Description, and Estimate. Data rows include:

id	Duration	PlanID	Description	Estimate
1	8640000000	1	Explore Central Park: Start your day with a leisurely stroll...	100
2	17280000000	2	Visit the City Palace and Jagdish Temple: Immense yours...	200
3	8640000000	3	Experience the Icehotel: Spend a day exploring the Iceho...	150
4	8640000000	4	Explore Victoria Peak and Star Ferry: Ascend to Victoria P...	150
5	17280000000	5	Hiking and Lake Activities: Embark on a scenic hike alon...	250
6	25920000000	6	Amazon Rainforest Adventure: Experience the wonders ...	300
7	8640000000	7	Snorkeling and Beach Relaxation: Spend a day snorkelin...	150
8	25920000000	8	Safari Adventure: Embark on a thrilling safari adventure i...	400
9	8640000000	9	Visit the Burj Khalifa and Dubai Mall: Ascend to the top ...	100
10	17280000000	10	Island Hopping and Snorkeling: Experience the beauty o...	300
11	17280000000	11	Gondola Ride and Piazza San Marco: Take a romantic go...	200
12	17280000000	12	Grand Canyon Views and Hiking: Explore the breathakin...	200
13	8640000000	13	Monte Carlo Casino and Promenade: Visit the legendary...	150
14	17280000000	14	Beach Relaxation and Water Activities: Relax on the pristi...	200
15	17280000000	15	Safari Adventure and Wildlife Encounters: Embark on a t...	200

Permissions table

SQLite Viewer screenshot showing the Permissions table. The table has columns: id, content_type, codename, and name. Data rows include:

id	content_type	codename	name
1	1	add_logentry	Can add log entry
2	2	change_logentry	Can change log entry
3	3	delete_logentry	Can delete log entry
4	4	view_logentry	Can view log entry
5	5	add_permission	Can add permission
6	6	change_permission	Can change permission
7	7	delete_permission	Can delete permission
8	8	view_permission	Can view permission
9	9	add_group	Can add group
10	10	change_group	Can change group
11	11	delete_group	Can delete group
12	12	view_group	Can view group
13	13	add_user	Can add user
14	14	change_user	Can change user
15	15	delete_user	Can delete user

Migrations table

SQLite Viewer screenshot showing the Migrations table. The table has columns: id, app, name, and applied. Data rows include:

id	app	name	applied
1	contenttypes	0001_initial	2024-06-18 20:51:05.053329
2	auth	0001_initial	2024-06-18 20:51:05.070436
3	admin	0001_initial	2024-06-18 20:51:05.085381
4	admin	0002_logentry_remove_auto_add	2024-06-18 20:51:05.098334
5	admin	0003_logentry_add_action_flag_choices	2024-06-18 20:51:05.106310
6	contenttypes	0002_remove_content_type_name	2024-06-18 20:51:05.124841
7	auth	0002_alter_permission_name_max_length	2024-06-18 20:51:05.137882
8	auth	0003_alter_user_email_max_length	2024-06-18 20:51:05.148775
9	auth	0004_alter_user_username_opts	2024-06-18 20:51:05.157535
10	auth	0005_alter_user_last_login_null	2024-06-18 20:51:05.170400
11	auth	0006_require_contenttypes_0002	2024-06-18 20:51:05.173895
12	auth	0007_alter_validators_add_error_messages	2024-06-18 20:51:05.184862
13	auth	0008_alter_user_username_max_length	2024-06-18 20:51:05.196820
14	auth	0009_alter_user_last_name_max_length	2024-06-18 20:51:05.208783
15	auth	0010_alter_group_name_max_length	2024-06-18 20:51:05.219743

Accommodation table

The screenshot shows the SQLite Viewer application interface. The title bar reads "db.sqlite3 M" and "Backend > JOURNEE > db.sqlite3". The main window displays a table titled "Accommodation" with 15 rows of data. The columns are: ID, Name, Type, PriceRange, Location, and PlanID. The table includes filters for each column and a "Filter 20 rows..." button. The left sidebar lists tables from the database, with "travel_plan_accommodation" selected. The bottom status bar shows "SILICATE-VIEWER v0.5.8" and "Page 1 / 1".

ID	Name	Type	PriceRange	Location	PlanID
1	The Ritz-Carlton	Hotel	\$700-\$2000	New York City, USA	1
2	The Oberoi Udaivilas	Hotel	\$500-\$1500	Udaipur, India	2
3	The Icehotel	Hotel	\$300-\$500	Jukkasjärvi, Sweden	3
4	The Peninsula	Hotel	\$600-\$1800	Hong Kong, China	4
5	The Fairmont Chateau Lake Louise	Hotel	\$400-\$1000	Lake Louise, Canada	5
6	The Treehouse Lodge	Lodge	\$200-\$400	Iquitos, Peru	6
7	The Four Seasons Resort Maui at Wailea	Resort	\$600-\$1500	Wailea, Maui, USA	7
8	The Serengeti Safari Lodge	Lodge	\$700-\$1200	Serengeti National Park, Tanzania	8
9	The Burj Al Arab Jumeirah	Hotel	\$1000-\$3000	Dubai, United Arab Emirates	9
10	The Maldives Four Seasons Landaa Giraavaru	Resort	\$1500-\$3000	Baa Atoll, Maldives	10
11	The Belmond Hotel Cipriani	Hotel	\$500-\$1200	Venice, Italy	11
12	The Grand Canyon Railway Hotel	Hotel	\$150-\$300	Grand Canyon Village, Arizona, USA	12
13	The Hotel de Paris	Hotel	\$600-\$1500	Monte Carlo, Monaco	13
14	The Amankila	Resort	\$800-\$2000	Nusa Dua, Bali, Indonesia	14
15	The Giraffe Manor	Hotel	\$400-\$800	Nairobi, Kenya	15

Settings.py

```
settings.py
...
Django settings for JUNIORF project.

Generated by 'django-admin startproject' using Django 3.0.0.

For more information on this file, see
https://docs.djangoproject.com/en/3.0/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/3.0/ref/settings/
...

from pathlib import Path

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/3.0/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'django-insecure-74q44dejuaw24ub0j1j4uia811x0u5j0'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = ['*']

# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'djongo',
    'users',
    'news',
    'travel'
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.BrokenLinkMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'djongo.middleware.DjangoMongoDBEngineMiddleware',
    'djongo.middleware.DjangoMongoDBCollectionMiddleware',
]

ROOT_URLCONF = 'JUNIORF.urls'

TEMPLATES = [
    {
        'NAME': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [BASE_DIR / 'templates'],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.request',
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'JUNIORF.wsgi.application'

# Database
# https://docs.djangoproject.com/en/3.0/ref/settings/#databases

DATABASES = [
    'default': {
        'ENGINE': 'djongo.db.backends.django',
        'NAME': BASE_DIR / 'db_np'
    }
]

# Password validation
# https://docs.djangoproject.com/en/3.0/ref/settings/#auth-password-validation

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]

# Internationalization
# https://docs.djangoproject.com/en/3.0/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_TZ = True

# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/3.0/howto/static-files/

STATIC_URL = 'static/'
STATICFILES_DIRS = [
    BASE_DIR / 'static',
]

MEDIA_URL = 'media/'
MEDIA_ROOT = BASE_DIR / 'media'
...

# Email settings
EMAIL_BACKEND = 'djongo_email.backends.djangoEmailBackend'
EMAIL_HOST = 'smtp.gmail.com' # Replace with your SMTP server
EMAIL_PORT = 587 # Typically port 587 for TLS, 465 for SSL
EMAIL_USE_TLS = True # Use TLS
EMAIL_HOST_USER = 'teamnameemail@gmail.com'
EMAIL_HOST_PASSWORD = 'password'
EMAIL_FROM_EMAIL = 'teamnameemail@gmail.com'
EMAIL_REPLY_TO = 'teamnameemail@gmail.com'
EMAIL_SUBJECT_PREFIX = 'Team Name'

# Default primary key field type
# https://docs.djangoproject.com/en/3.0/ref/settings/#default-primary-key-field

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
```

Urls.py

```
urls.py

"""
URL configuration for JOURNEE project.

The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/5.0/topics/http/urls/
Examples:
Function views
1. Add an import: from my_app import views
2. Add a URL to urlpatterns: path('', views.home, name='home')
Class-based views
1. Add an import: from other_app.views import Home
2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
Including another URLconf
1. Import the include() function: from django.urls import include, path
2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
"""

from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('core.urls')),
    path('users/', include('users.urls')),
    path('travel_plan/', include('travel_plan.urls')),
]
```

Booking

```
bookings.html

<h2>Plan Your Trip</h2>
<!-- templates/travel_plan.html -->
<form
    method="post"
    action="{% url 'travel_plan:booking' %}"
    style="margin-top: 50px">
    {% csrf_token %}
    <div class="row form-group">
        <div class="col-md-12">
            <label for="Budget">Budget</label>
            <input
                type="number"
                id="Budget"
                name="budget"
                class="form-control"
            >
        </div>
    </div>

    <div class="row form-group">
        <div class="col-md-12">
            <h4 style="color: black;">
                <label for="travel_Style">Travel Style</label>
            </h4>
            <select
                name="travel_style"
                id="travel_Style"
                class="form-control"
            >
                <option value="" disabled selected>
                    Select your travel theme
                </option>
                <option value="Adventure">Adventure</option>
                <option value="Relaxation">Relaxation</option>
                <option value="Culture">Culture</option>
                <option value="Culinary">Culinary (food tours)</option>
                <option value="Religious">Religious/ Pilgrimage</option>
                <option value="Eco-tourism">Eco-tourism</option>
                <option value="Nature">Nature exploration</option>
            </select>
        </div>
    </div>

    <div class="row form-group">
        <div class="col-md-12">
            <h4 style="color: black;">
                <label for="travel_group">Travel Group</label>
            </h4>
            <select
                name="travel_group"
                id="travel_group"
                class="form-control"
            >
                <option value="" disabled selected>
                    Select your travel group
                </option>
                <option value="Solo">Solo</option>
                <option value="Family">Family</option>
                <option value="Friends">Friends</option>
                <option value="Couple">Couple</option>
            </select>
        </div>
    </div>
```

```
booking.html

<div class="row form-group">
    <div class="col-md-12">
        <h4 style="color: black;">
            <label for="destination">Destination</label>
        </h4>
        <input
            type="text"
            name="destination"
            class="form-control"
        >
    </div>
</div>
<div class="row form-group">
    <div class="col-md-12">
        <h4 style="color: black;">
            <label for="Country_of_origin">Country of origin</label>
        </h4>
        <input
            type="text"
            name="coo"
            class="form-control"
        >
    </div>
</div>

<div class="row form-group">
    <div class="col-md-12">
        <h4 style="color: black;">
            <label for="date-start">Start Date Travel</label>
        </h4>
        <input
            type="date"
            id="date-start"
            name="date_start"
            class="form-control"
        >
    </div>
</div>

<div class="row form-group">
    <div class="col-md-12">
        <label for="date-end">End Date Travel</label>
        <h4 style="color: black;">
            <input
                type="date"
                id="date-end"
                name="date_end"
                class="form-control"
            >
        </h4>
    </div>
</div>

<div class="row form-group">
    <div class="col-md-12">
        <input
            type="submit"
            class="btn btn-primary btn-block"
            value="Submit"
            style="background: #95d5e8"
        >
    </div>
</div>
```



booking.css

```
.container-b{  
    overflow: hidden;  
    display: flex;  
    justify-content: space-between;  
    background:rgb(6,58,81);  
    height: 950px;  
  
}  
.tab-content{  
    width: 500px;  
    padding: 3px 10px 3px 10px;  
    margin-top: 90px !important;  
    background: #d3e7e7;  
    border-radius: 10px;  
}  
.tab-content-inner>h2{  
    padding-top: 10px;  
    text-align: center;  
}
```

Edit profile

```
edit_profile.html

<body style="background-color: #007bff; overflow: hidden;">

    <div class="form-container">
        <h2>Edit Profile</h2>
        <form id="editProfileForm" method="POST" action="{% url 'users:edit_profile' %}">
            {% csrf_token %}
            <label for="user_name">Name:</label>
            <input type="text" id="username" name="username" value="{{ user.username }}">

            <label for="email">Email:</label>
            <input type="email" id="email" name="email" placeholder="Enter email address" value="{{ user.email }}">

            <label for="password">Change Password:</label>
            <input type="password" id="password" name="password" placeholder="Enter password">

            <button type="submit">Save Changes</button>
            <button type="button" class="cancel-btn" id="cancelButton">Cancel</button>
        </form>
    </div>

    <script>
        document.getElementById("cancelButton").onclick = function() {
            window.location.href = "{% url 'users:view_profile' %}";
        };
    </script>
</body>
```

```
editprofile.css

.form-container {
    background-color: #f5f5f7;
    padding: 15px;
    border-radius: 8px;
    top: 50%;
    left: 50%;
    box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
    max-width: 400px;
    width: 100%;
    margin: 70px auto 10px auto;
    position: absolute;
    transform: translate(-50%, -50%);
    z-index: 1000; /* Ensure it's above other content */
}

h2 {
    text-align: center;
    color: #333;
    margin-bottom: 10px;
}

form {
    display: flex;
    flex-direction: column;
}

label {
    margin-bottom: 5px;
    color: #666;
    font-weight: bold;
}

input[type="text"],
input[type="number"],
input[type="email"],
input[type="password"] {
    margin-bottom: 10px;
    padding: 8px;
    border: 1px solid #ddd;
    border-radius: 5px;
    font-size: 16px;
}

button {
    padding: 8px;
    border: none;
    border-radius: 5px;
    font-size: 16px;
    cursor: pointer;
}

button[type="submit"] {
    background-color: #28a745;
    color: #fff;
    margin-bottom: 10px;
}

button.cancel-btn {
    background-color: #dc3545;
    color: #fff;
}

button:hover {
    opacity: 0.9;
}

@media (max-width: 400px) {
    .form-container {
        padding: 10px;
    }
}

h2 {
    font-size: 24px;
}

label {
    font-size: 14px;
}

input[type="text"],
input[type="number"],
input[type="email"],
input[type="password"] {
    font-size: 14px;
}

button {
    font-size: 14px;
}
```

Payment form

```
paymentform.css
@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@100;300;400;600;800&display=swap');

* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    border: none;
    outline: none;
    font-family: 'Poppins', sans-serif;
    text-transform: capitalize;
    transition: all 0.2s linear;
}

body {
    background: #007bff;
    overflow: hidden;
}

.container {
    display: flex;
    justify-content: center;
    align-items: center;
    min-height: 100vh;
    padding: 25px;
}

.container form {
    width: 700px;
    padding: 20px;
    background-color: #fff;
    box-shadow: 0px 8px 30px rgba(0, 0, 0, 0.2);
}

.container form .row {
    display: flex;
    flex-wrap: wrap;
    gap: 15px;
}

.container form .row .col {
    flex: 1 1 250px;
}

.col .title {
    font-size: 20px;
    color: #007bff;
    padding-bottom: 5px;
}

.col .inputBox {
    margin: 15px 0;
}

.col .inputBox label {
    margin-bottom: 3px;
    display: block;
}

.col .inputBox input,
.col .inputBox select {
    width: 100%;
    border: 1px solid #ccc;
    padding: 10px 15px;
    font-size: 15px;
}

.col .inputBox input:focus,
.col .inputBox select:focus {
    border: 4px solid #000;
}

.col .flex {
    display: flex;
    gap: 15px;
}

.col .flex .inputBox {
    flex: 1 1;
    margin-top: 5px;
}

.col .inputBox img {
    height: 34px;
    margin-top: 5px;
    filter: drop-shadow(0 0 1px #000);
}

.container form .submit_btn {
    width: 100px;
    padding: 15px;
    font-size: 17px;
    background-color: #007bff;
    color: #fff;
    margin-top: 5px;
    cursor: pointer;
    letter-spacing: 1px;
}

.container form .cancel_btn {
    width: 100px;
    padding: 15px;
    font-size: 17px;
    background-color: #247140;
    color: #fff;
    margin-top: 5px;
    cursor: pointer;
    letter-spacing: 1px;
}

.container form .submit_btn:hover {
    background-color: #33a77b;
}

.container form .cancel_btn:hover {
    background-color: #33a77b;
}

input::-webkit-inner-spin-button,
input::-webkit-outer-spin-button {
    display: none;
}
```

Sign up and login

```
auth.html

<div class="cont_forms" >
    <div class="cont_img_back_">
        
    </div>
    <div>
        <form class="cont_form_login" method="POST" action="{% url 'users:login' %}">
            {% csrf_token %}

            <a href="#" onclick="hidden_login_and_sign_up()" ><i class="material-icons">8*xE504;<i><a>
            <h2>LOGIN<h2>

            <input type="text" name="username" placeholder="username" required >
            <input type="password" name="password" placeholder="Password" required >
            <button class="btn_login" type="submit" name="login_button" >LOGIN<button>
        </form>
    </div>
```

```
auth.html

<div>
    <form class="cont_form_sign_up" method="POST" action="{% url 'users:register' %}">
        {% csrf_token %}

        <a href="#" onclick="hidden_login_and_sign_up()" ><i class="material-icons">8*xE504;<i><a>
        <h2>SIGN UP<h2>
        <input type="text" name="username" id="username" placeholder="username" required >
        <input type="text" name="email" id="email" placeholder="email" required >
        <input type="password" name="password" id="password" placeholder="password" required >
        <input type="password" name="confirm_password" id="confirm_password" placeholder="confirm password" required >
        <button class="btn_sign_up" type="submit">SIGN UP<button>
    </form>
</div>
```

```
signUp.css

.cont_login {
    position: relative;
    width: 640px;
}

.cont_back_info {
    position: relative;
    float: left;
    width: 640px;
    height: 280px;
    overflow: hidden;
    background-color: #fff;
    box-shadow: 1px 10px 30px -10px rgba(0, 0, 0, 0.5);
}

.cont_forms {
    position: absolute;
    overflow: hidden;
    top: 0px;
    left: 0px;
    width: 320px;
    height: 280px;
    background-color: #eee;
    -webkit-transition: all 0.5s;
    -moz-transition: all 0.5s;
    -ms-transition: all 0.5s;
    -o-transition: all 0.5s;
    transition: all 0.5s;
}

.cont_forms_active_login {
    box-shadow: 1px 10px 30px -10px rgba(0, 0, 0, 0.5);
    height: 420px;
    top: -60px;
    left: 0px;
    -webkit-transition: all 0.5s;
    -moz-transition: all 0.5s;
    -ms-transition: all 0.5s;
    -o-transition: all 0.5s;
    transition: all 0.5s;
}

.cont_forms_active_sign_up {
    box-shadow: 1px 10px 30px -10px rgba(0, 0, 0, 0.5);
    height: 420px;
    top: -60px;
    left: 320px;
    -webkit-transition: all 0.5s;
    -moz-transition: all 0.5s;
    -ms-transition: all 0.5s;
    -o-transition: all 0.5s;
    transition: all 0.5s;
}
```

Travel program

```
program_output.html

{%
  load static %
}
{%
  load markdown_extras %
}
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Travel Program</title>
    <link rel="stylesheet" href={% static 'travel_plan/css/travelprogram.css' %}>
    <link rel="stylesheet" href={% static "travel_plan/css/styles.css" %}>
  </head>
  <body>
    <h2 class="page-title">Here's your program</h2>
    <div class="card">
      <div class="card-content" id="travelProgramContent">
        <!-- Travel program content will be dynamically inserted here -->
        <h3 style="text-align: center;">Activities</h3>
        {{activities|markdown}}
        <hr>
        <h3 style="text-align: center;">Flight</h3>
        {{flight|markdown}}
        <hr>
        <h3 style="text-align: center;">Accommodation</h3>
        {{accommodation|markdown}}
        <hr>
      </div>
    </div>
    <button onclick="downloadTravelProgram()">Download Program</button>
    <script>
      // Function to download the travel program as a file
      function downloadTravelProgram() {
        const content = document.getElementById(
          "travelProgramContent"
        ).innerHTML;
        const blob = new Blob([content], { type: "text/html" });
        const url = URL.createObjectURL(blob);
        const a = document.createElement("a");
        a.href = url;
        a.download = "travel_program.html";
        document.body.appendChild(a);
        a.click();
        document.body.removeChild(a);
        URL.revokeObjectURL(url);
      }

      // Call the fetchTravelProgram function when the page loads
      fetchTravelProgram();
    <script>
  </body>
</html>
```

```
travelprogram.css

body {
    display: flex;
    flex-direction: column;
    align-items: center;
    margin: 0;
    background-color: #45A9B3;
}

.page-title {
    font-size: 24px;
    color: #FFF;
    margin-top: 20px;
    text-align: center;
}

.card {
    border: 1px solid #CCC;
    border-radius: 5px;
    padding: 20px;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
    background-color: #A8F5E0;
    max-width: 600px;
    width: 100%;
    margin-top: 20px; /* Adjust as needed */
}

.card-content {
    font-size: 16px;
}

button {
    margin-top: 20px;
    padding: 10px 20px;
    font-size: 16px;
    color: #FFF;
    background-color: #4CAF50;
    border: none;
    border-radius: 5px;
    cursor: pointer;
}

button:hover {
    background-color: #45D9C9;
}

/* Form of satisfaction */
form {
    display: flex;
    flex-direction: column;
    align-items: center;
    margin-top: 20px; /* Adjust as needed */
}

form button {
    margin-top: 10px; /* Adjust spacing between buttons */
    width: 150px; /* Adjust width as needed */
}

/* Additional styles for Yes and No buttons */

.yes-button, .no-button {
    margin-top: 10px; /* Adjust spacing between buttons */
    width: 150px; /* Adjust width as needed */
}

button#save {
    background-color: #E0F2F1;
    color: #4CAF50;
}

button#save:hover {
    background-color: #4CAF50;
    color: #FFF;
}

button#refresh {
    background-color: #E0F2F1;
    color: #4CAF50;
}

button#refresh:hover {
    background-color: #4CAF50;
    color: #FFF;
}
```

```

models.py

from django.db import models
from django.contrib.auth.models import User
from django.conf import settings
from django.utils import timezone

# Create your models here.

class TravelGroup(models.Model):
    TGD = models.AutoField(primary_key=True)
    TGCategory = models.CharField(max_length=100)

    def __str__(self):
        return self.TGCategory

class TravelStyle(models.Model):
    TSD = models.AutoField(primary_key=True)
    Theme = models.CharField(max_length=100)

    def __str__(self):
        return self.Theme

class Plan(models.Model):
    PlanID = models.AutoField(primary_key=True)
    user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
    destination_country = models.CharField(max_length=100)
    destination_city = models.CharField(max_length=100)
    start_date = models.DateField()
    end_date = models.DateField()
    budget = models.DecimalField(max_digits=10, decimal_places=2)
    travel_group = models.ForeignKey(TravelGroup, on_delete=models.CASCADE)
    travel_style = models.ForeignKey(TravelStyle, on_delete=models.CASCADE)

    def __str__(self):
        return f'{self.destination_city}, {self.destination_country} ({self.start_date} - {self.end_date})'

class Accommodation(models.Model):
    AccommodationID = models.AutoField(primary_key=True)
    PlanID = models.ForeignKey(Plan, on_delete=models.CASCADE)
    Name = models.CharField(max_length=100)
    PriceRange = models.CharField(max_length=50)
    Location = models.CharField(max_length=100)
    Type = models.CharField(max_length=100)

    def __str__(self):
        return self.Name

class Transportation(models.Model):
    TransportationID = models.AutoField(primary_key=True)
    PlanID = models.ForeignKey(Plan, on_delete=models.CASCADE)
    Type = models.CharField(max_length=100)
    Origin = models.CharField(max_length=100)
    Destination = models.CharField(max_length=100)
    Duration = models.DurationField()
    EstimatedCost = models.DecimalField(max_digits=10, decimal_places=2)

    def __str__(self):
        return f'{self.Type} from {self.Origin} to {self.Destination}'

class Review(models.Model):
    ReviewID = models.AutoField(primary_key=True)
    PlanID = models.ForeignKey(Plan, on_delete=models.CASCADE)
    Rating = models.IntegerField()
    user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)

    def __str__(self):
        return f'Review {self.ReviewID} for Plan {self.PlanID}'"

class Activity(models.Model):
    PlanID = models.ForeignKey(Plan, on_delete=models.CASCADE)
    Duration = models.DurationField()
    EstimatedCost = models.DecimalField(max_digits=10, decimal_places=2)
    Description = models.TextField()

    def __str__(self):
        return f'Activity {self.Description} for Plan {self.PlanID}'"

# ... your existing models ...

class Payment(models.Model):
    PAYMENT_STATUS_CHOICES = [
        ('PENDING', 'Pending'),
        ('COMPLETED', 'Completed'),
        ('FAILED', 'Failed'),
    ]
    payment_id = models.AutoField(primary_key=True)
    plan = models.ForeignKey(Plan, on_delete=models.CASCADE)
    user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
    amount = models.DecimalField(max_digits=10, decimal_places=2)
    payment_method = models.CharField(max_length=100) # e.g., 'Credit Card', 'Paypal'
    status = models.CharField(
        max_length=10, choices=PAYMENT_STATUS_CHOICES, default='PENDING'
    )
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
    transaction_id = models.CharField(max_length=255, blank=True, null=True) # Optional field for storing external payment transaction ID

    def __str__(self):
        return f'Payment {self.payment_id} for Plan {self.plan.PlanID}'

    def complete(self, transaction_id):
        """
        Mark the payment as completed and update the transaction ID.
        """
        self.status = 'COMPLETED'
        self.transaction_id = transaction_id
        self.updated_at = timezone.now()
        self.save()

    def fail(self):
        """
        Mark the payment as failed.
        """
        self.status = 'FAILED'
        self.updated_at = timezone.now()
        self.save()

```

```

from django.http import HttpResponseRedirect
from django.urls import reverse
from django.shortcuts import render
from gemini import Gemini
import re

# Create your views here.
def booking(request):
    # Cookies placed here
    if request.method == "POST":
        budget = int(request.POST.get("budget"))
        travel_style = request.POST.get("travel_style")
        travel_group = request.POST.get("travel_group")
        destination = request.POST.get("destination")
        coo = request.POST.get("coo")
        date_start = request.POST.get("date_start")
        date_end = request.POST.get("date_end")

        # A_budget = budget // 2
        # F_budget = round(budget * 0.2)
        # H_budget = round(budget * 0.3)

        urls_to_remove = [
            r"http://googleusercontent.com/tool_disclaimer_content/1",
            r"http://googleusercontent.com/hotel_content/0",
            r"http://googleusercontent.com/right_content/0",
            r"\[Image of [ ]\]+"
        ]
        url_pattern = r"|\.".join(urls_to_remove)

        # cookie inserted here
        cookies = {
            "NID": "AcjejMDt1f1Zq2eId",
            "SSID": "du41scps6lw-AM1",
            "APISID": "7DvrmP4y8t1TQcAb3L3mVP613pCUM13",
            "SAPISID": "OvdIgsoTMh0cfWnJ/AVve798w-wJ7yoS8P",
            "__Secure-1PAPISID": "OvdIgsoTMh0cfWnJ/AVve798w-wJ7yoS8P",
            "__Secure-3PAPISID": "OvdIgsoTMh0cfWnJ/AVve798w-wJ7yoS8P",
            "SEARCH_SAMESITE": "OgImZs0",
            "AEC": "AQTFBhxpxSKMHDOLFqgxkmeLubtz3cApHodr-9dsn6_Ic0QubnyA-px4hGBM",
            "go": "941_1_943748756_1/16402824",
            "SID": "g_0001A1gIRVb07JFaAA1G002o3UcLeMnjrls-",
            "CTEV140uvvp3R4g9XqhtsU_1dFeI2dkEY1wACgyKAR5SARASFQHx2M1hsHymreTcDX8xrNMyGoBoVAUF8yJcRoYpdRtQ5c6AoN-D7R52U0676",
            "__Secure-1PSID": "g_0001A1gIRVb07JFaAA1G002o3UcLeMnjrls-CteV140uvvp3R4g20Tq-b-wBOnqvfb127TzhwACgYKAuXASARASFQHx2M1XNwXoMHUjixuKgCs_xoAUFBqTwPu4sWwKf9fWz5Gghb0076",
            "__Secure-3PSID": "g_0001A1gIRVb07JFaAA1G002o3UcLeMnjrls-",
            "CTEV140uvvp3R4gpbUtrIuvgJNlW50vEf8sgpACgYKAuXASARASFQHx2M1XFRU_uLpt10Y0jv1k2JBvAUFBuKp1cAB9XMPg-0vCk-o0vU0076",
            "NID": "535",
            "__Secure-1PISID": "sidts-CjEB3EgAEqtGfzLbUjgu3J_F18TxTpL-jWSPIM40uHpgSmv1UoEEIG9pNiJRFw1VB0pEAA",
            "__Secure-3PISID": "sidts-CjEB3EgAEqtGfzLbUjgu3J_F18TxTpL-jWSPIM40uHpgSmv1UoEEIG9pNiJRFw1VB0pEAA",
            "SIDCO": "AKEuXzX_4Ww4D4BqfqlTnsAxVvCaDpA1EXS2_WB_FF3djuq1kxhV8ZVaAqj14yx10121zA",
            "__Secure-1PSIDCO": "AKEuXzVLTxJ0fJVL8TPLcbobuIv6wKL5_w7niwY_n7SH0KzI37znLHzBm0uvs10_qP43_Bg",
            "__Secure-3PSIDCO": "AKEuXzVLTxJ0fJVL8TPLcbobuIv6wKL5_w7niwY_n7SH0KzI37znLHzBm0uvs10_qP43_Bg",
            "go_wCS7kJ502Z": "GS1.1.1719059446.11.1.1719059452.0.0.0"
        }
        client = Gemini(cookies=cookies)
        Activities = client.generate_content(
            f'''Hello gemini, I'm travelling to {destination},
            from {date_start} to {date_end}, my country of origin is {coo}(use it to conclude the currency and show the
            current exchange rate) and my budget is {budget}.
            my travel group is {travel_group}, and my travel style is {travel_style}. I need you to act as a travel
            advisor and reply only
            with an itinerary that contains a list of activities about what to do each day and how much will be spent per
            activity
            (include internal mode of transportation and its costs)
            without giving additional tips or anything in the end. just a plain itinerary with no intros or outros.
            (format it in bullet points for each day e.g day1... day2... etc) also exclude flight and accommodation
            expenses.
            taking in consideration the data I told you about above"""
        )
        activities = re.sub(url_pattern, "", Activities.text)

        Flights = client.generate_content(
            f'''Hello gemini, I'm travelling to {destination} from {date_start} to {date_end}.
            my country of origin is {coo}, can you show me a list of all available flights (in bullet points).
            Reply only in bullet points with no intros or outros"""
        )

        flight = re.sub(url_pattern, "", Flights.text)

        Accommodation = client.generate_content(
            f'''Hello gemini, I'm travelling to {destination} from {date_start} to {date_end} with a budget of {budget}.
            my country of origin is {coo}(use it to conclude the currency). I need you to reply with a list of hotels and
            the price per night in bullet points.
            (No intro or outro)"""
        )

        accommodation = re.sub(url_pattern, "", Accommodation.text)

        return render(
            request,
            "travel_plan/program_output.html",
            {
                "activities": activities,
                "flight": flight,
                "accommodation": accommodation,
            },
        )
    else:
        return render(request, "travel_plan/booking.html")

```

Chapter 6

Testing

Methodology

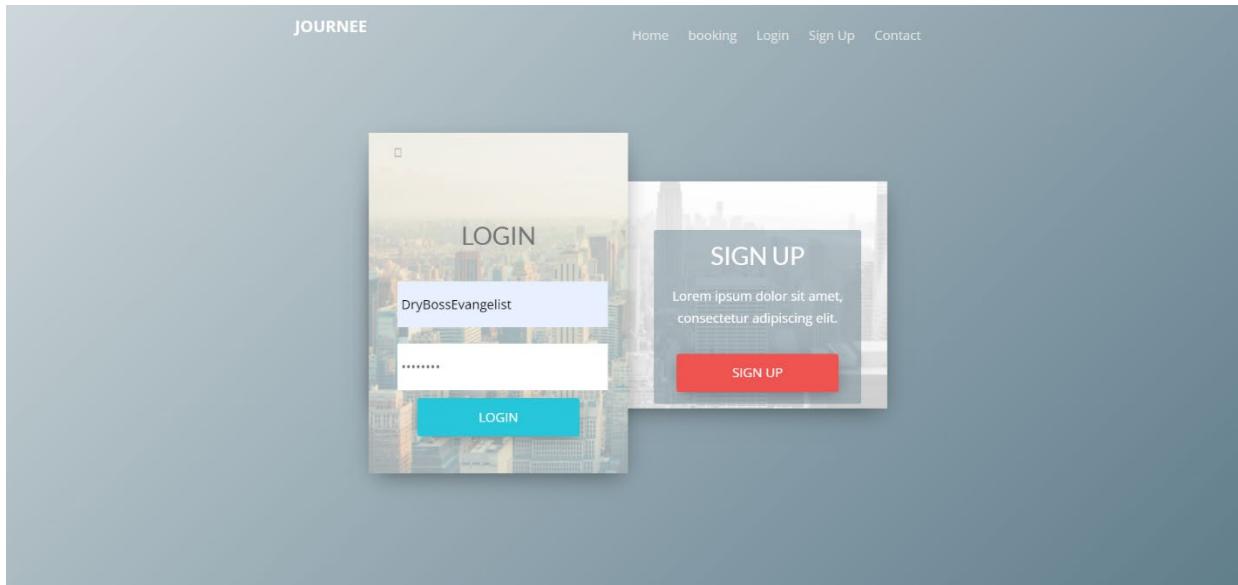
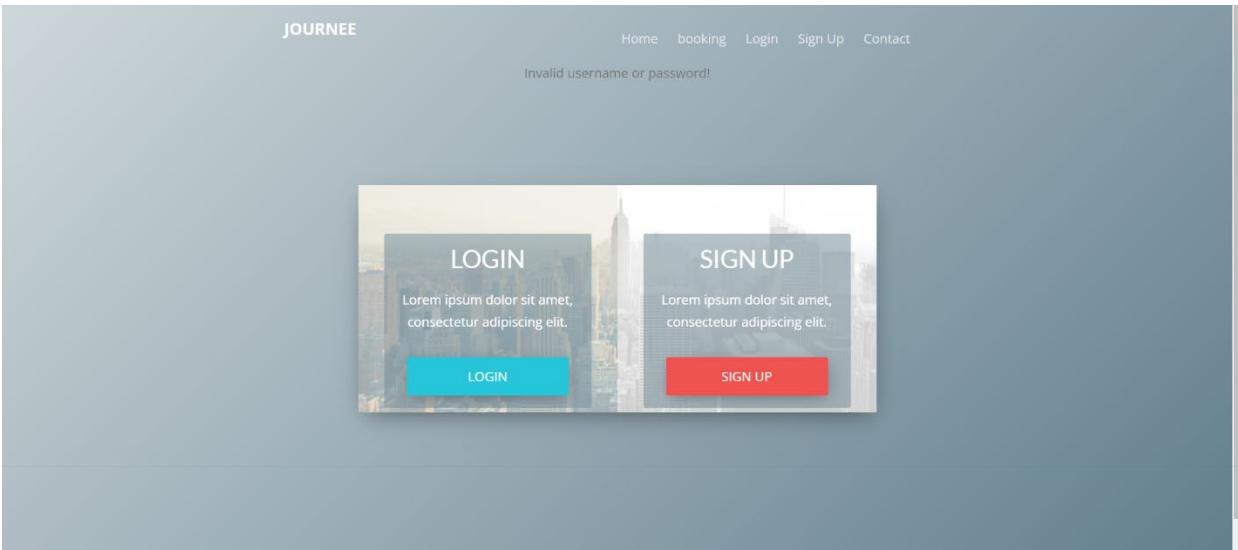
6.1 Introduction

Ensuring the quality, reliability, and functionality of any software system is paramount to its success. Testing plays a crucial role in achieving this goal, as it provides a systematic process for uncovering potential flaws, identifying areas for improvement, and ultimately verifying that the system meets the intended requirements. This chapter will delve into the testing methodology employed for the Journee project, outlining the strategies and techniques used to validate its functionality and ensure a robust and user-friendly experience. By thoroughly testing the system at different stages of development, we aimed to identify and address potential issues, ultimately delivering a product that meets the expectations of our users.

6.2 Testing methodology

The testing methodology employed for Journee adopted a comprehensive and multi-faceted approach, encompassing a variety of testing methods to ensure thorough validation. The process involved a combination of unit testing, integration testing, system testing, and user acceptance testing (UAT), each addressing a specific aspect of the system's functionality and performance. Unit testing focused on verifying the individual components of the code, ensuring that each module worked as intended. Integration testing then evaluated how different components interacted with each other, ensuring seamless data flow and communication. System testing was conducted to assess the overall functionality of the system as a whole, simulating real-world scenarios and testing the system's ability to meet the user requirements. Finally, UAT involved real users providing feedback on the system's usability, performance, and overall experience, offering invaluable insights for refining the final product.

6.3 Login validation



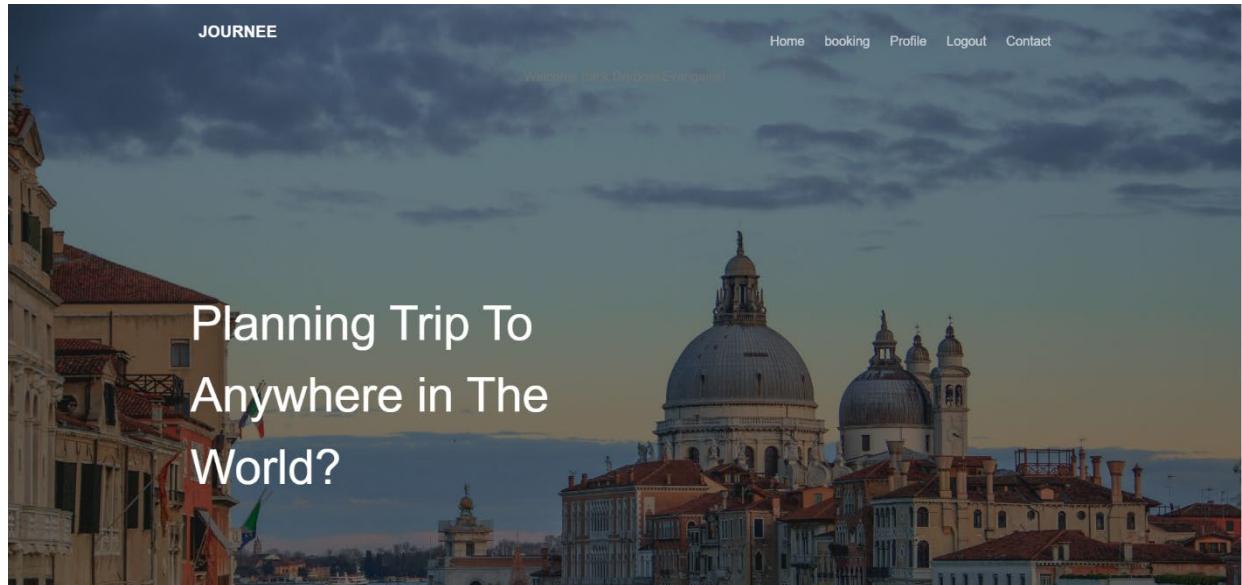
```
views.py

from django.shortcuts import render, redirect
from django.contrib import messages
from django.contrib.auth import authenticate, login, logout, update_session_auth_hash
from django.contrib.auth.models import User
from django.contrib.auth.decorators import login_required
from django.core.mail import send_mail
from django.conf import settings


def login_view(request):
    if request.method == 'POST':
        username = request.POST['username']
        password = request.POST['password']
        user = authenticate(username=username, password=password)

        if user is not None:
            login(request, user)
            messages.success(request, 'Welcome back ' + username + '!')
            return redirect('core:home')
        else:
            messages.error(request, "Invalid username or password!")

    return render(request, 'users/auth.html')
```



6.4 Sign-up validation

The image consists of three vertically stacked screenshots of a web application interface, likely a travel booking site named "JOURNEE".

Screenshot 1: The top screenshot shows the sign-up process. On the right, a "SIGN UP" form is displayed with fields for "DryBossEvangelist" (username), "aa535399@gmail.com" (email), and two password fields. On the left, a "LOGIN" form is shown with placeholder text "Lorem ipsum dolor sit amet, consectetur adipiscing elit." and a blue "LOGIN" button.

Screenshot 2: The middle screenshot shows the same interface after attempting to sign up with an existing email. A message "Email already exists!" is displayed above the sign-up form.

Screenshot 3: The bottom screenshot shows the interface after a successful sign-up. A message "Registered successfully! Please check your email." is displayed above the sign-up form.

```
views.py

from django.shortcuts import render, redirect
from django.contrib import messages
from django.contrib.auth import authenticate, login, logout, update_session_auth_hash
from django.contrib.auth.models import User
from django.contrib.auth.decorators import login_required
from django.core.mail import send_mail
from django.conf import settings

def register_view(request):
    if request.method == 'POST':
        username = request.POST['username']
        email = request.POST['email']
        password = request.POST['password']
        confirm_password = request.POST['confirm_password']

        if password != confirm_password:
            messages.error(request, "Passwords do not match!")
            return redirect('users:register')

        if User.objects.filter(email=email).exists():
            messages.error(request, "Email already exists!")
            return redirect('users:register')

        if User.objects.filter(username=username).exists():
            messages.error(request, "Username already exists!")
            return redirect('users:register')

        user = User.objects.create_user(username=username, email=email, password=password)
        user.save()

        ...
        send_mail(
            'Welcome to our site',
            'Thank you for registering on our website.',
            settings.DEFAULT_FROM_EMAIL,
            [email],
            fail_silently=False,
        )
        ...

        messages.success(request, "Registered successfully! Please check your email.")
        return redirect('users:login')

    return render(request, 'users/auth.html')
```

6.5 Main functions validations

1. Itinerary generation

Here's your program

Activities

Itinerary for Berlin Trip (Solo Traveler, Culture Enthusiast)

Day 1 (Tuesday, June 25th):

- Morning:** Check into your hotel and leave your luggage.
- Activity:** Explore Museum Island (). Spend the morning wandering through the museums on this UNESCO World Heritage Site. Tickets cost around €18 (\$20.44). Take the U-Bahn (underground train) to Museumsinsel station (Zones AB), costing €3.40 (\$3.87) for a single ticket.
- Afternoon:** Take a walking tour of the Brandenburg Gate () and surrounding area. Tours typically last 2-3 hours and cost around €15 (\$17.08).
- Evening:** Enjoy a traditional German dinner in Mitte, Berlin's historic center. Expect to spend around €20 (\$22.77) for dinner.

Estimated Costs for Day 1:

- Museum Island Tickets: €18 (\$20.44)
- U-Bahn Tickets: €3.40 (\$3.87)
- Walking Tour: €15 (\$17.08)
- Dinner: €20 (\$22.77)
- Total:** €56.80 (\$64.16)

Day 2 (Wednesday, June 26th):

- Morning:** Visit the Reichstag Building () for a free tour of the parliament building and its famous dome. Tickets can be booked online in advance. Take the U-Bahn to Bundestag station (Zones AB), costing €3.40 (\$3.87) for a single ticket.
- Afternoon:** Explore the remnants of the Berlin Wall at the East Side Gallery (), an open-air gallery featuring murals painted on the remaining sections of the wall. The

Total Estimated Costs for Activities (excluding flights and accommodation):

- Day 1: €56.80 (\$64.16)
- Day 2: €26.80 (\$30.51)
- Day 3:

Flight

- SWISS LX239 LX966 departing at 3:10 AM, arriving at 1:55 PM, priced from EGP 19,742 (round trip) will take 11 hours and 45 minutes with a layover in Zürich (ZRH) for 6 hours.
- Austrian OS866 OS233 departing at 3:40 AM, arriving at 5:45 PM, priced from EGP 20,073 (round trip) will take 15 hours and 5 minutes with a layover in Vienna (VIE) for 10 hours and 20 minutes.
- Lufthansa LH887 LH1942 departing at 3:25 AM, arriving at 3:05 PM, priced from EGP 20,277 (round trip) will take 12 hours and 40 minutes with a layover in Munich (MUC) for 7 hours and 35 minutes.
- EgyptAir MS731 departing at 11:35 AM, arriving at 2:45 PM, priced from EGP 23,891 (round trip) will take 4 hours and 10 minutes (non-stop).
- Lufthansa LH885 LH190 departing at 5:20 AM, arriving at 4:55 PM, priced from EGP 20,709 (round trip) will take 12 hours and 35 minutes with a layover in Frankfurt (FRA) for 6 hours and 55 minutes.

Accommodation

- [Berlin Marriott Hotel](#) - EGP 12,223 per night
- [Melia Berlin](#) - EGP 12,032 per night
- [Hotel NH Collection Berlin Mitte Friedrichstrasse](#) - EGP 11,839 per night
- [Grand Hyatt Berlin](#) - EGP 18,242 per night
- [The Ritz-Carlton Berlin](#) - EGP 20,482 per night

[Download Program](#)

```

views.py

from django.http import HttpResponseRedirect, HttpResponse
from django.urls import reverse
from django.shortcuts import render
from gemini import Gemini
import re

# Create your views here.
def booking(request):
    # Cookies placed here
    if request.method == "POST":
        budget = int(request.POST.get("budget"))
        travel_style = request.POST.get("travel_style")
        travel_group = request.POST.get("travel_group")
        destination = request.POST.get("destination")
        coo = request.POST.get("coo")
        date_start = request.POST.get("date_start")
        date_end = request.POST.get("date_end")

        # A_budget = budget // 2
        # F_budget = round(budget * 0.2)
        # H_budget = round(budget * 0.3)

        urls_to_remove = [
            r"http://googleusercontent.com/tool_disclaimer_content/1",
            r"http://googleusercontent.com/hotel_content/0",
            r"http://googleusercontent.com/flight_content/0",
            r"\[Image of [!]\]+",
        ]
        url_pattern = r"|\".join(urls_to_remove)"

        # cookie inserted here
        cookies = [
            "NID": "AcjemDtiT12qzeld",
            "SSID": "x1v41scpsb6lw-AX1",
            "APISID": "7DwmpJdy81t7QcAb3L3mVP513pCUH13",
            "SAPISID": "OvdIgsoTMnqcfWnJ/Ayve798v-wJ7yo8SP",
            "__Secure-1PAPISID": "OvdIgsoTMnqcfWnJ/Ayve798v-wJ7yo8SP",
            "__Secure-3PAPISID": "OvdIgsoTMnqcfWnJ/Ayve798v-wJ7yo8SP",
            "SEARCH_SAMESITE": "GogImZsB",
            "AEC": "AQTFBhvpx8KMHDLFqgxkmJuhtz2sA9Hds-8s00u8nyA-pxAhGBM",
            "ga": "GA1.1.843748736.176402824",
            "GID": "g_d0001A161RWb7JFdAA16902633UjleWmjrs",
            "CtEV1L40uvap3R4g9XqghU_1dfe12dkEY1wAcgYKATk5ARASQf0GX2MhHymneIcDX8xrN4ynGcBoVAUF8yKrRoYpdRtQ5c0AcN-D7R52U6676",
            "__Secure-1PSID": "g_d0001A161RWb7JFdAA16902633UjleWmjrs-CtEV1L40uvap3R4g201q6",
            "wBOnqvfb127TzIxw0gjKAxsXARASF040X2M18Nm0cmH-JU1j0xuksCz_xoAUFBp4sJdWKf9fwZ5Gh0b0076",
            "__Secure-3PSID": "g_d0001A161RWb7JFdAA16902633UjleWmjrs",
            "CtEV1L40uvap3R4qplUrT1gvQW8D0vef8sgACgYKASYSARASFOHGX2M1XFRU_uhptIQYDj1k2JBsVAUF8yKp1cAB0XMPg-oVfCk-o0vU8078",
            "NID": "515",
            "__Secure-1PSIDTS": "sidts-CjEB3EgAEqt0FzLB0g2u3J_F18TxHTpL-jWSPjM40yHpgSmv1UqEEI0pNjJRfw1VB0pEAA",
            "__Secure-3PSIDTS": "sidts-CjEB3EgAEqtGFLm0u1u3J_F18TxHTpL-jWSPjM40yHpgSmv1UqEEI0pNjJRfw1VB0pEAA",
            "SIDCO": "AKEqx2X_u4Wq4D8qfqtqDTnsAXDvCaqj1EXs2_Mh_FPD3uyg1KGw8ZV0Asqj14yo1012zrA",
            "__Secure-1PSIDCO": "AKEqx2VL7x1GFJvL8TPLcc0uIv5wKL5-wtNiW_y7SH0KzI3Y7enLHzb0uowS10_p43_5g",
            "__Secure-3PSIDCO": "AKEqx2VL7x1GFJvL8TPLcc0uIv5wKL5-wtNiW_y7SH0KzI3Y7enLHzb0uowS10_p43_5g",
            ".ga_WCS7k_J50Z": "GS1.1.1719059446.11.1.1719059452.0.0.0",
        ]
        url_pattern = r"|\".join(url_pattern)"

        client = Gemini(cookies=cookies)
        Activities = client.generate_content(
            f"""Hello gemini, I'm travelling to {destination},
            from {date_start} to {date_end}, my country of origin is {coo}(use it to conclude the currency and show the
            current exchange rate) and my budget is {budget}.
            my travel group is {travel_group}, and my travel style is {travel_style}. I need you to act as a travel
            advisor and reply only
            with an itinerary that contains a list of activities about what to do each day and how much will be spent per
            activity
            (include internal mode of transportation and its costs)
            without giving additional tips or anything in the end. just a plain itinerary with no intros or outros.
            (format it in bullet points for each day e.g day1... day2... etc) also exclude flight and accommodation
            expenses.
            taking in consideration the data I told you about above"""
        )

        activities = re.sub(url_pattern, "", Activities.text)

        Flights = client.generate_content(
            f"""Hello gemini, I'm travelling to {destination} from {date_start} to {date_end}.
            my country of origin is {coo}.can you show me a list of all available flights (in bullet points).
            Reply only in bullet points with no intros or outros"""
        )

        flight = re.sub(url_pattern, "", Flights.text)

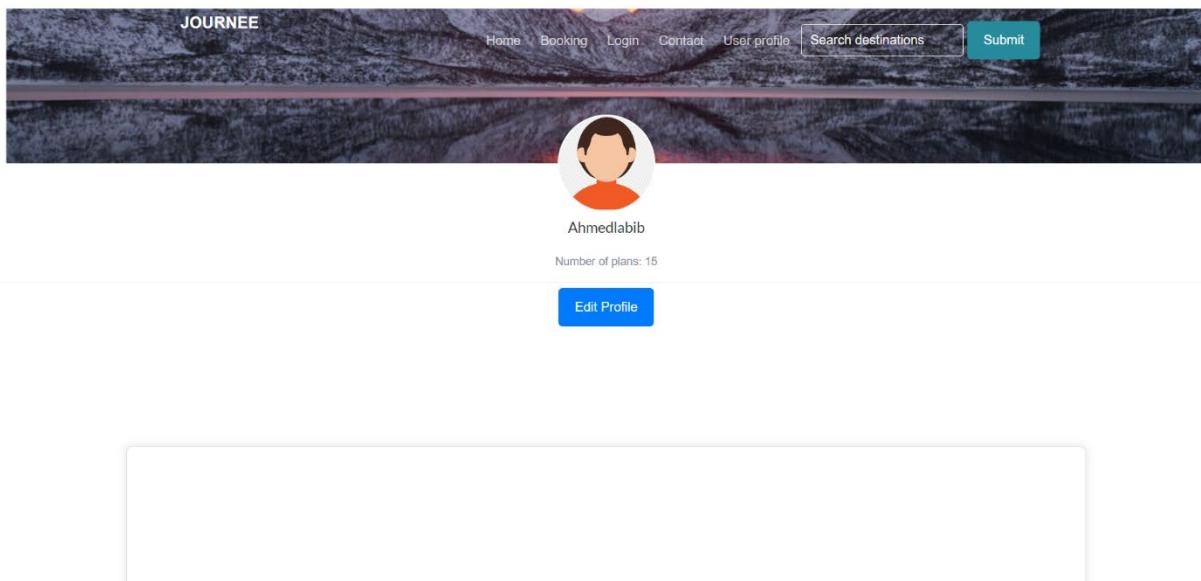
        Accommodation = client.generate_content(
            f"""Hello gemini, I'm travelling to {destination} from {date_start} to {date_end} with a budget of {budget}.
            my country of origin is {coo}(use it to conclude the currency).I need you to reply with a list of hotels and
            the price per night in bullet points.
            (No intro or outro)"""
        )

        accommodation = re.sub(url_pattern, "", Accommodation.text)

        return render(
            request,
            "travel_plan/program_output.html",
            {
                "activities": activities,
                "flight": flight,
                "accommodation": accommodation,
            },
        )
    else:
        return render(request, "travel_plan/booking.html")

```

2. View profile



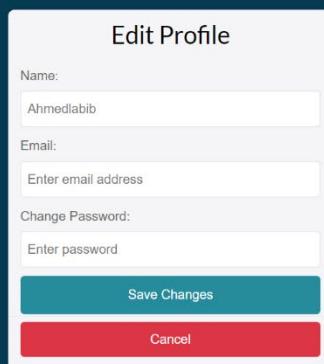
```
views.py

from django.shortcuts import render, redirect
from django.contrib import messages
from django.contrib.auth import authenticate, login, logout, update_session_auth_hash
from django.contrib.auth.models import User
from django.contrib.auth.decorators import login_required
from django.core.mail import send_mail
from django.conf import settings

@login_required
def view_profile(request):

    context = {
        'username': request.user.username,
        'email': request.user.email,
    }
    return render(request, 'users/user_profile.html', context)
```

3.Edit profile



Edit Profile

Name:
Ahmedlabib

Email:
Enter email address

Change Password:
Enter password

Save Changes

Cancel

```
views.py

from django.shortcuts import render, redirect
from django.contrib import messages
from django.contrib.auth import authenticate, login, logout, update_session_auth_hash
from django.contrib.auth.models import User
from django.contrib.auth.decorators import login_required
from django.core.mail import send_mail
from django.conf import settings

@login_required
def edit_profile(request):
    if request.method == 'POST':

        username = request.POST['username']
        email = request.POST['email']
        password = request.POST['password']

        user = request.user
        user.username = username
        user.email = email

        if password:
            update_session_auth_hash(request, user) # Prevents logout after password change
            user.save()

        messages.success(request, 'Your profile has been updated successfully.')
        return redirect('users:view_profile')

    return render(request, 'users/edit_profile.html')
```

4.Payment

Billing Address		Payment	
Full Name:		Card Accepted:	
<input type="text" value="Enter Your Full Name"/>			
Email:		Name :	
<input type="text" value="Enter Email Address"/>		<input type="text" value="Enter Card Name"/>	
Address:		Account Number:	<input type="text" value="1111-2222-3333-4444"/>
<input type="text" value="Enter Address"/>			
City:		Exp Month:	<input type="text" value="Choose month"/>
<input type="text" value="Enter City"/>			
State:	Zip Code:	Exp Year:	
<input type="text" value="Enter State"/>	<input type="text" value="123 456"/>	<input type="text" value="Choose Year"/>	
Proceed To Checkout			
Cancel			

```
views.py

from django.shortcuts import render, redirect
from django.views.generic import View
from django.conf import settings
from django.http import HttpResponseRedirect

import stripe

# Replace with your actual stripe secret key
stripe.api_key = settings.STRIPE_SECRET_KEY

class PaymentView(View):
    def get(self, request):
        # Render a form with payment details
        context = {
            'stripe_publishable_key': settings.STRIPE_PUBLISHABLE_KEY,
        }
        return render(request, 'payment/payment.html', context)

    def post(self, request):
        # Retrieve payment information from the form
        token = request.POST.get('stripeToken')
        amount = int(request.POST.get('amount')) # Convert to integer for stripe

        # Create a Stripe charge
        try:
            charge = stripe.Charge.create(
                amount=amount,
                currency='usd',
                description='Payment for your purchase',
                source=token,
            )

            # Handle successful payment
            # Redirect to a success page or perform other actions
            return redirect('payment:success')
        except stripe.error.CardError as e:
            # Handle card errors
            error_message = e.json_body.get('error', {}).get('message')
            return render(request, 'payment/payment.html', {'error': error_message})
        except Exception as e:
            # Handle other exceptions
            return HttpResponseRedirect(f'An error occurred: {e}')



```

Chapter 7

Conclusion &

Future work

7.1 Introduction

Our “Journee” -No pun intended- ends and concludes with This final chapter that concludes our exploration of the Journee project, summarizing the key achievements and insights gained throughout the development process. It also provides a platform for reflection on the project's significance and potential impact. We will highlight the project's contributions to the field of AI-powered travel planning, as well as discuss future opportunities for enhancing and expanding the functionality of Journee.

This chapter will serve as a comprehensive overview of the project's accomplishments, providing a valuable resource for future development efforts and inspiring further innovation in the realm of personalized travel planning.

7.2 Conclusion

This section provides a comprehensive reflection on the Journee project, outlining its key achievements, discussing the implications of our research, summarizing the development process, and highlighting the new knowledge contributed to the field of AI-powered travel planning.

A Summary of Key Arguments and Results:

The Journee project successfully demonstrated the potential of leveraging AI to create personalized travel itineraries tailored to individual preferences. We designed and implemented a web application that utilizes Google AI API to generate customized travel plans based on user-specific inputs, including destination, budget, travel style, and duration. Through careful analysis of user requirements and meticulous design of the system architecture, we built a robust and user-friendly platform that caters to the needs of diverse travelers.

A Short Discussion of the Implications of Research:

Our project's success in integrating AI technology into the travel planning process has significant implications for the future of the industry. By providing users with personalized and efficient itineraries, Journee can help to:

Simplify Trip Planning: Reduce the time and effort required to plan a trip, allowing travelers to focus on enjoying their experience.

Increase Travel Accessibility: Empower individuals with limited travel planning experience or time constraints to easily create comprehensive itineraries.

Promote Sustainable Travel: Encourage travelers to explore diverse destinations and discover unique experiences that align with their interests and budgets.

Foster Innovation: Drive the development of more sophisticated and personalized travel planning tools and services powered by AI.

Reflecting on the Research Process:

The research process involved a comprehensive approach that combined theoretical exploration, practical implementation, and continuous evaluation. We started by analyzing the existing landscape of travel planning tools and identifying the limitations of traditional methods. This analysis informed the development of our core concept, which focused on incorporating AI capabilities to generate personalized itineraries.

We designed the system architecture using UML diagrams, including class diagrams and ERDs, to ensure a robust and scalable system. We then implemented the frontend using HTML, CSS, and JavaScript, creating a user-friendly interface for inputting travel preferences and viewing generated itineraries. The backend was developed

using Python and the Django framework, which allowed us to handle user requests, interact with the database, process travel preferences, and integrate with Google Gemeni AI API for itinerary generation.

Throughout the development process, we conducted rigorous testing to ensure the system's functionality and performance. We faced challenges in optimizing and engineering AI Prompt to generate diverse and creative itineraries, and we learned valuable lessons about the importance of user feedback in refining the system's capabilities.

Show What New Knowledge You Have Contributed to Your Field:

Our research has contributed new knowledge to the field of AI-powered travel planning by:

Demonstrating the Practical Application of AI: We successfully integrated Gemeni API into a functional web application, demonstrating the feasibility and value of using AI for personalized travel planning.

Developing Comprehensive System Architecture: Our design process, incorporating UML diagrams and database mapping, provided a robust and scalable framework for building AI-powered travel planning tools.

Highlighting the Importance of User Feedback: Through continuous evaluation and user feedback, we were able to refine the system's features and enhance the user experience, demonstrating the crucial role of user input in shaping the development of AI-driven travel solutions.

Overall, the Journee project has made significant contributions to the field of AI-powered travel planning, demonstrating the potential of this technology to

revolutionize the way we plan and experience our journeys. We believe that our research will inspire further development of innovative and personalized travel solutions that will enhance the travel experience for everyone.

7.3 Future Work

While we believe our project has achieved significant milestones in personalized travel planning, we envision a future where the application can evolve even further, offering a truly immersive and enriching travel experience. Here are some exciting directions for future development:

1. Expanding the AI Capabilities:

Multimodal Itinerary Generation: Beyond just suggesting activities, accommodations, and transportation, Journee could incorporate AI-powered recommendations for immersive experiences like local food tours, cultural workshops, and off-the-beaten-path adventures.

Real-Time Itinerary Adjustment: Leveraging real-time data, such as weather forecasts, traffic conditions, and event schedules, Journee could dynamically adjust itineraries, ensuring optimal travel flow and maximizing user satisfaction.

Flexible Customization: Journee empowers users to take control of their itinerary by providing tools to add, remove, or rearrange suggested items. This allows users to personalize the plan to their exact preferences and ensure it perfectly matches their vision for their trip. Users can also save multiple alternative versions of their itinerary, enabling them to compare different options and choose the one that best suits their needs.

2. Integrating with Other Travel Services:

Booking Platform Integration: Seamless integration with online booking platforms for flights, accommodations, and activities would streamline the travel planning process, allowing users to book directly within Journee.

Real-Time Travel Updates: Journee could integrate with real-time travel information providers like Google Maps or Flightradar24 to offer up-to-the-minute travel updates, helping users navigate unexpected delays or changes.

Personalized Travel Assistant: Through voice assistant integration, Journee could become a personalized travel companion, answering questions, providing guidance, and helping users navigate their journeys effortlessly.

3. Enhancing User Experience:

AR/VR Integration: Integrating augmented and virtual reality technologies could allow users to virtually explore destinations and experience potential activities firsthand, making the planning process more engaging and informative.

Social Sharing and Collaboration: Enabling users to share their itineraries, collaborate on group trips, and discover travel inspiration from other users could foster a sense of community and encourage travel exploration.

Gamification and Rewards: Introducing gamified elements, like challenges and rewards for completing travel tasks, could add an extra layer of engagement and encourage users to actively participate in the planning process.

References

Chapter 1: Introduction

- General Travel Industry Trends:

- [**World Tourism Organization \(UNWTO\)**](#): - Provides global travel data and statistics. You can cite specific reports on tourism trends.
- [**Skift**](#) - A leading news and research platform for the travel industry. They offer insightful articles and reports on various travel trends, including the impact of AI.
- [**Statista**](#): - This is a comprehensive statistics platform that you can use to find data on the travel industry, including market sizes, trends, and consumer behavior.
- [**Phocuswright**](#): - A market research firm specializing in the travel industry, offering reports on various travel segments.

- AI in Travel:

- [**Google AI Blog**](#): - Follows the latest developments in Google's AI research and its applications in travel.
- [**MIT Technology Review**](#): - A publication covering emerging technologies, including the use of AI in travel and tourism.
- [**Forbes**](#) - A great resource for articles on the role of AI in the travel industry.
- [**McKinsey & Company**](#): - A management consulting firm that offers insights on AI and its impact on various industries, including travel.

Chapter 2: Business Plan

- **Marketing:**
 - [**Hootsuite**](#): - A platform for social media management; you can cite their blog posts or articles on travel marketing best practices.
 - [**Content Marketing Institute**](#): Provides resources on creating and executing successful content marketing strategies.
 - [**SEMrush**](#): - A tool for SEO and online marketing, providing valuable data and insights. You can cite their blog posts or reports on relevant topics.
- **SWOT Analysis:**
 - [**MindTools**](#): - A website with a good explanation of SWOT analysis and how to conduct it.
 - [**Harvard Business Review**](#): - A reputable publication that offers articles on various business topics, including SWOT analysis.
- **Legal Structure:**
 - [**SBA \(Small Business Administration\)**](#): - This website offers a lot of information on business structures, including partnerships. You can cite their guides on choosing the right structure for your business.
- **Financial Planning:**
 - [**Investopedia**](#) - - A great resource for financial planning information, including understanding balance sheets, cash flow statements, and profit margins.
 - [**The Balance**](#): - A website offering guides and articles on various financial topics, including business planning.

- [**Entrepreneur**](#): - A publication covering entrepreneurship, business development, and financial planning.

Chapter 3: System Analysis

- **UML Diagrams:**

- [**UML.org**](#) - The official website for Unified Modeling Language (UML), where you can find documentation and resources for the different diagram types.
- [**Visual Paradigm**](#): - A popular tool for creating UML diagrams; their website has helpful guides and tutorials.
- [**IBM**](#): IBM offers resources and guides on UML modeling and its applications in software development.
- [**Lucid chart**](#): is the intelligent diagramming application that brings teams together to make better decisions and build the future.

Chapter 4: System Design

- **Database Design:**

- [**SQL Tutorial**](#) - - A great resource for learning SQL and database design.
- [**MySQL**](#): - A popular database management system with a fantastic client workbench

Chapter 5: System Implementation

- **Python and Django:**

- [**Django Documentation**](#) - - The official Django documentation, a great resource for all things Django.
- [**Real Python**](#) - A website with many tutorials and articles on Python and Django development.

- [**Python.org**](https://www.python.org) - The official website for Python; their documentation and tutorials are valuable resources.
- **API Integration:**
 - [**Google Cloud Platform**](https://cloud.google.com) - - The official website for Google Cloud, which provides documentation on their AI APIs.

Chapter 7: Conclusion & Future Work

- **AI in Travel (Specific Applications):**
 - [**Skift**](https://skift.co): - A great resource for articles on the future of AI in travel, including potential applications.
 - [**VentureBeat**](https://venturebeat.com): - A technology news website that covers the latest developments in AI and travel.
 - [**TechCrunch**](https://techcrunch.com): - A leading technology news website covering AI and its impact on various industries.