SYSTEM REQUIREMENTS

P02: TRAVEL PLANNER

STUDENT ID	NAME
25100211	HASAN MALIK
25100181	SAAD ILYAS
25100017	M. UMER JAMIL
25100016	ABDUL AHAD
25100257	ANIQA AQEEL

TABLE OF CONTENTS

1. Introduction

In today's fast-paced and interconnected world, efficient travel planning is crucial for both personal and professional endeavors. The proliferation of travel options has made the process of finding the best deals on flights and hotels increasingly complex. To address this challenge, our project aims to develop a sophisticated meta search engine designed to streamline travel planning by aggregating data from a diverse range of travel sites.

The core objective of this project is to create a comprehensive platform that consolidates flight and hotel information from multiple sources, including major airlines, budget carriers, and various accommodation providers. The platform will offer users a seamless and efficient search by leveraging advanced algorithms and real-time data integration,experience. The goal is to simplify the decision-making process, saving users both time and money while ensuring they receive the most up-to-date and relevant travel options available.

Our target users include a wide range of travelers—from business professionals seeking cost-effective and convenient travel solutions to leisure travelers looking for the best deals on vacations. The platform is designed to cater to both novice and experienced users by providing an intuitive interface that makes navigation straightforward and accessible. Users will benefit from enhanced decision-making capabilities through detailed information, including user reviews and ratings, which will guide them in selecting options that best meet their needs and preferences.

In addition to its user-centric design, the platform will integrate with Amadeus API services to enhance its capabilities for flight and hotel data aggregation. This integration will enable the meta search engine to access a vast network of travel information, ensuring comprehensive and accurate results.

Overall, this project aspires to revolutionize the travel planning experience by providing a powerful, all-in-one solution that addresses the complexities of finding the best travel deals. By offering a consolidated view of available options and empowering users with critical information, our meta search engine will set a new standard in travel planning efficiency and user satisfaction.

2. System Actors

Actor Name	Description
Traveler	This is the primary user of the Travel Planner site. This user searches for flights, and hotels. Moreover, Traveler also compare different options.
System Administrator	System Administrator will maintain the system, which includes managing user accounts, ensuring smooth operational running of the website. Look out for any glitches or unexpected errors, and update the website.
Amadeus API	Amadeus API supplies the data for hotels and flights, a key component of our meta-search engine that aggregates the information.
Travel Service Providers	These service providers will give information through the Amadeus API. These providers will make sure that the information being sent to the website is fully correct and updated.

Currency Converter	This API is responsible for providing us with real-time currency exchange rates to display prices in the traveler's preferred currency.
Weather API	This API is responsible for providing us with real-time weather updates and forecasts. Travelers will keep the weather into account while making their travel decisions.

3. Functional Requirements

Requirements	
Sr#	**Requirement**
1	As a new ==user== (traveler or admin), I want to sign up for an account by providing my email, password, and basic information so that I can access personalized features of the travel planner.

2	As a registered user (traveler or admin), I want to sign in to my account using my email and password so that I can access my saved preferences and booking history.
3	As a user (traveler or admin) I want to reset my password in case I want to change it or forget it.
4	As a traveler, I want to search for flights across multiple airlines and travel sites so that I can find the best deals for my trip.
6	As a traveler, I want to search for hotels from various accommodation providers so that I can compare prices and amenities easily.
7	As a traveler, I want to filter search results based on price, travel time, and other preferences so that I can quickly find options that meet my needs.
8	As a traveler, I want to view user reviews and ratings for flights and hotels so that I can make informed decisions about my bookings.
9	As a business traveler, I want to sort results based on convenience factors (such as direct flights or proximity to business districts) so that I can optimize my work-related travel.
10	As a leisure traveler, I want to see package deals that combine flights and hotels so that I can potentially save money on my vacation.

11	As a ==traveler==, I want to filter accommodation options by amenities (e.g., free Wi-Fi, pool, gym) so that I can find hotels that meet my specific needs.
12	As a ==traveler==, I want to read and submit reviews for hotels and airlines I've used so that I can share my experiences with other travelers.
13	As a ==traveler==, I want to access a currency converter within the platform so that I can easily understand prices in my preferred currency.
14	As a ==traveler==, I want to access and download my booking confirmations and itineraries so that I have all my travel information in one place.
15	As a ==traveler==, I want to see the weather forecast for the place I am visiting so that I can make a better decision about my trip.
16	As a ==system administrator==, I want to monitor system performance and uptime so that I can ensure a smooth user experience and address any issues promptly.
17	As a ==system administrator==, I want to update or change the design of the website.
18	As a ==system administrator==, I want to integrate and manage connections with multiple travel sites and the Amadeus API so that the platform can provide comprehensive and up-to-date results.

19	As a travel site partner (Amadeus API), I want to provide real-time availability and pricing data to the meta search engine so that users can see accurate information from my site.
20	As a traveler, I want to save and organize my favorite flights and hotels in a wishlist so that I can easily compare and revisit them later.
21	As a system administrator, I want to generate and analyze usage reports and statistics so that I can track user behavior and improve the platform.
22	As a system administrator, I want to manage user accounts, including the ability to suspend or delete accounts if necessary, to maintain the integrity of the platform.

4. Non-functional Requirements / Quality Attributes

Sr#	Requirements
1	The system should not utilize more than 1 GB of memory at any time during its execution.
2	The system should not fail more than 3 times every 24 hours. In case of a failure, the system should restore to normal operations within 5 minutes of a failure.
3	The response time for search queries should be less than 2 seconds for 95% of the requests.
4	The user interface should be responsive and provide a consistent experience across different devices, including desktops, tablets, and smartphones.
5	The system should be accessible 24/7 with an uptime of 99.9% over any given month.
6	The system should ensure data integrity and consistency, particularly when handling concurrent updates to booking information.

7	The system should provide user authentication and authorization mechanisms to ensure that users can only access their own data and features.
8	The system should be compatible with the latest versions of major web browsers, including Chrome, Firefox, Safari, and Edge.
9	The system should be designed to handle up to 1 million records in the database without significant performance degradation.

5. Security Requirements

Sr#	Security Risks	Potential Losses	Controls
1	Cross-Site Scripting (XSS)	Malicious actors can inject JavaScript code into web pages, potentially compromising user data, hijacking user sessions, or redirecting users to malicious sites.	Implement input validation to sanitize all user inputs, ensuring no malicious code or HTML tags are executed. Use Content Security Policy

			(CSP) to restrict resource loading.
2	Injection (SQL, NoSQL, Command Injection)	Attackers can inject malicious queries or commands into your system through vulnerable inputs (e.g., search forms, API requests), potentially leading to unauthorized data access, data corruption, or complete database compromise.	Use parameterized queries or prepared statements for database interactions. Implement input validation and sanitization on all user inputs to prevent malicious code execution. Conduct regular security code reviews and vulnerability scanning.
3	DDOS attacks	A large volume of malicious traffic can overwhelm your servers, making your website inaccessible to legitimate users, resulting in lost revenue, reputational damage, and potential business downtime.	Use a web application firewall (WAF) and distributed content delivery network (CDN) like cloudflare to mitigate large-scale traffic. Implement rate limiting, traffic filtering, and automated bot detection systems. Regularly monitor network traffic for unusual patterns.

4	Weak Authentication	Attackers may gain unauthorized access to accounts due to weak passwords or insecure authentication methods, compromising user data.	Password Strength Enforcement: Enforce strong password policies by requiring a mix of characters and regular password updates.
5	Data Breach	Unauthorized access to user data, including personal information, payment details, and travel preferences, leading to loss of user trust and legal penalties.	Data Encryption: Implement SSL/TLS encryption for all data transmitted between users and the server to protect sensitive information. Secure Password Storage: Use secure password hashing algorithms (e.g., bcrypt) for storing user passwords. Access Control: Restrict access to sensitive data based on user roles (e.g., only administrators can access certain data).

6. Who Did What?

Name of the Team Member	Tasks done
Aniqa Aqeel	Security requirements
Hasan Malik	Introduction, Security Requirements
M. Umer Jamil	Functional Requirements
Abdul Ahad	System Actors
Saad Ilyas	Non Functional Requirements

7. Review checklist

Before submission of this deliverable, the team must perform an internal review. Each team member will review one or more sections of the deliverable.

Section Title	Reviewer Name(s)
Introduction	Aniqa Aqeel

Actors	Saad Ilyas
Functional Requirements	Abdul Ahad Bin Ali
Non-functional requirements	M. Umer Jamil
Security Requirements	Hasan Malik