

Heaven's Light is Our Guide



Travel and Tour Booking Website

By

Nafisha Nawar

(2110026)

Software Development Project- II (ECE-3100)

Examination Committee: Prof. Dr. Md. Anwar Hossain (Head)
Asst. Prof. Hafsa Binte Kibria
(Supervisor)

Rajshahi University of Engineering & Technology

Faculty of Electrical & Computer Engineering

Department of Electrical & Computer Engineering

Rajshahi-6204

September 2025

Acknowledgment

I am profoundly grateful to my respected supervisor, Assistant Professor Hafsa Binte Kibria, Department of Electrical and Computer Engineering, RUET, for her continuous guidance, encouragement, and valuable suggestions throughout the development of this project. Her insightful feedback, constructive criticism, and kind support have been effective in shaping this work and ensuring its successful completion by making me push my limits in terms of learning and enhancing my skills regarding this project.

I would also like to extend my thanks to the Head of the Department Prof. Dr. Md. Anwar Hossain and all faculty members, for their cooperation and assistance whenever needed.

Table of Contents

Content	Page
Abstract	1
Chapter 1: Introduction	
1.1 Introduction	2
1.2 Problem Statement	2
1.3 Project Objectives & Outcomes (linked to COs)	3
1.4 Significance of the Study	3
1.5 Scope of the Project	3
Chapter 2: Requirements & Tools	
2.1 Introduction	4
2.2 Software Requirements	4
2.3 Programming Languages & Frameworks	4-5
2.4 Development Tools (IDE, Version Control, Libraries)	5
2.5 System Architecture Overview	5
Chapter 3: Database Design	
3.1 Database Requirements	6
3.2 Entity Relation Diagram (ERD)	6-7
3.3 Normalization	7
3.4 Table Structures	8-9
3.5 Database Integration with Backend	9
Chapter 4: System Design	
4.1 Use Case Diagram	10
4.2 Class Diagram	10
4.3 User Interface (UI/UX) Design	11-12
Chapter 5: System Implementation	
5.1 Introduction	13
5.2 Module-wise Implementation	13
5.3 Backend Feature Implementation	13
5.4 Code Quality & Structure	14
5.5 Security Features	14
Chapter 6: Testing & Debugging	
6.1 Testing Strategy (Unit, Integration, System)	15
6.2 Test Cases & Results	15
6.3 Debugging & Error Handling	15
6.4 Performance Evaluation	15
6.5 Limitations & Future Improvements	16
Chapter 7: Project Management & Participation	
7.1 Development Methodology (Agile, Waterfall, etc.)	17
7.2 Timeline	17
7.3 Team roles and contribution	17
7.4 Attendance & Participation	17
Chapter 8: Documentation & Report	
8.1 Objectives & Design Rationale	18
8.2 Screenshots & Results	18-21

8.3 Conclusion	22
Chapter 9: References	23
Chapter 10: Appendices	
Appendix A: Source Code / GitHub Link	24

List of Figures

Content	Name	Page
Figure-01	ER Diagram (attributes)	6
Figure-01	ER Diagram (relations)	7
Figure-03	Normalization	7
Figure-04	Use Case Diagram	10
Figure-05	Class Diagram	10
Figure-6.1	UI Design Demo (user panel)	11
Figure-6.2	UI Design Demo (2) (user panel)	11
Figure-6.3	UI Design Demo (1) (admin panel)	12
Figure-6.4	UI Design Demo (2) (admin panel)	12
Figure-6.5	UI Design Demo (3) (admin panel)	12
Figure-6.6	UI Design Demo (4) (admin panel)	12

List of Tables

Content	Name	Page
Table-01	users	8
Table-02	hotels	8
Table-03	book	9

Abstract

This project presents the design and implementation of a Travel and Tour Booking Website that simplifies hotel booking and travel planning for users and travel agencies. The system enables users to search hotels by city, select hotels based on number of people and travel dates, and confirm bookings through an intuitive interface. A secure login/signup system ensures proper account management, while the admin panel provides tools for managing hotels, editing hotel details, and verifying user bookings.

The website was developed using HTML, CSS, JavaScript, PHP, and MySQL, ensuring a responsive frontend and robust backend. The database efficiently stores user accounts, hotel information, and booking records. Security features such as password hashing and input validation are included to protect user data.

The outcome is a functional, user-friendly, and efficient hotel booking platform that can be adopted by travel agencies to manage their operations digitally and help users book hotels easily. The system also has the potential to serve as a medium for connecting local users with overseas hotels.

Chapter 1: Introduction

1.1 Introduction:

Tourism and travel are essential parts of today's global economy. With the rapid expansion of the internet, most people prefer online booking systems rather than traditional manual methods. A well-designed hotel booking website can reduce delays, prevent miscommunication, and make the booking process easier for both users and travel agencies.

This project focuses on developing a Travel and Tour Booking Website that allows users to search and book hotels according to their needs. It also provides travel agencies with an admin dashboard to manage hotels and bookings effectively. For further approaches, this website can be adapted by any travel agency for their overseas trip booking which will provide users a wider range of choice and personalized experience.

This website can also be used without the association with any travel agency. Rather it can provide a centralized system where user can book hotels from all around the world for their solo or family trips according to their need and without the interference of any third party.[1]

1.2 Problem Statement:

Traditional hotel booking systems in many travel agencies still rely on manual processes, such as phone calls or physical visits. These methods are time-consuming, error-prone, and often inconvenient for customers, especially those from abroad. Travel agencies also face difficulties in verifying bookings and managing large amounts of hotel data manually.

In another case, most people who do not want to plan their trip through any agency, rather prefer to do it on their own, there are very few centralized hotels booking system to favor that. They have to individually book hotels from each and every destination which can be very time-consuming and difficult.

To overcome these issues, a web-based Travel and Tour Booking Website is required. And this website was built around to overcome these issues.

1.3 Project Objectives & Outcomes:

Objectives:

- To design and implement a hotel booking website with a user-friendly interface.
- To enable users to search hotels by city, date, and number of people.
- To provide a secure login/signup system for both users and administrators.
- To create an admin panel for managing hotels, editing hotel details, and verifying bookings.
- To assist travel agencies or individuals in managing customers and bookings more efficiently.

Outcomes:

- A functional and tested hotel booking website.
- A working admin panel that supports hotel management and booking verification.
- A database system that securely stores user, hotel, and booking information.

1.4 Significance of the Study:

This study is significant because it provides advantages to both travel agencies and customers. For users, the system offers an easy way to browse available hotels, check details, and complete bookings without hassle. This reduces time, effort, and dependency on traditional manual booking methods.

For travel agencies, the platform reduces workload, minimizes errors, and ensures all bookings are stored and verified digitally. Additionally, the system can serve as a medium for connecting local users with overseas hotels, creating opportunities for agencies to expand their services globally and reach more customers.

1.5 Scope of the Project:

The project mainly focuses on hotel search and booking functionality with separate modules for users and administrators. Users can create accounts, log in, search for hotels based on city, date, and number of people, and make bookings accordingly. The administrator has access to a dashboard that allows hotel management, editing hotel details, and verifying customer bookings. The system uses a database to store all records securely and supports smooth interaction between the user interface and backend.

However, the project does not include advanced features such as an online payment gateway or complete travel package booking. It does not provide the users the freedom to choose specific rooms or amenities. The database is limitedly connected and does not have the access to the specific hotel's database. It is limited to demonstrating hotel booking and management functions for a prototype version.[2]

Chapter 2: Requirements & Tools

2.1 Introduction:

Every software project requires proper planning of hardware, software, programming languages, and development tools to ensure smooth implementation. Since this project is a web-based hotel booking system, both client-side (frontend) and server-side (backend) technologies were used. This chapter describes the requirements and tools that were essential for building and running the system successfully.[3]

2.2 Software Requirements:

The project required different types of software, including an operating system, a web server, and a database server. The major software requirements include:

- Operating System: Windows 11
- Local Server: XAMPP (Apache + MySQL + PHP)
- Database: MySQL, CodeIgniter.[4]
- Code Editor: Visual Studio Code
- Browser: Google Chrome for testing

2.3 Programming Languages & Frameworks:

The project was built using a combination of frontend and backend technologies. Each language and framework played an important role in ensuring that the system is interactive, functional, and user-friendly. A detailed explanation is given below:[5]

- **HTML5 & CSS3**

HTML5 was used as the foundation for structuring the web pages, defining elements such as forms, tables, buttons, and navigation menus. It provided the skeleton of the website. CSS3 was applied to style the pages, making them visually appealing through layouts, colors, fonts, and responsive design. Together, HTML and CSS ensured that the website had a clear structure and professional appearance.

- **JavaScript**

JavaScript was used on the client side to add interactivity and enhance user experience. It allowed dynamic features such as form validation, interactive hotel search filters, and updating page content. JavaScript also handled real-time updates and improved the responsiveness of the website, making the system smoother for users.

- **PHP**

PHP served as the main backend scripting language. It processed user requests such as login, signup, hotel booking, and hotel management. It also connected with the MySQL database to fetch or store data. For example, when a user searched for a hotel by city, PHP handled the query and retrieved matching results from the database. PHP was also used in the admin panel to manage hotels and verify bookings.

- **SQL**

SQL (Structured Query Language) was used to design and manage the database. It created tables for users, hotels, and bookings and maintained relationships among them. SQL queries were written for inserting new records (e.g., new hotel or booking), updating data (e.g., editing hotel information), and retrieving stored records (e.g., showing user bookings). SQL ensured data consistency and security within the system.

2.4 Development Tools

In addition to the programming languages, some tools and libraries were used to simplify the development process:

Visual Studio Code (VS Code):

As the main integrated development environment (IDE).[6]

XAMPP Control Panel:

To run Apache and MySQL servers locally.

phpMyAdmin:

For managing the database through a web interface.

2.5 System Architecture Overview

The system follows a client-server architecture. The client (user) interacts with the system through a browser, sending requests such as searching for hotels or booking a trip. These requests are processed by the server (PHP), which communicates with the MySQL database to fetch or update data. The server then returns the response to the client, which is displayed on the website.

In simple terms:

User → Browser → Server (PHP) → Database (MySQL) → Response Back to User

This architecture ensures proper interaction between frontend and backend components and allows smooth management of hotels, users, and bookings

Chapter 3: Database Design

3.1 Database Requirements:

The database is a core part of the tour booking website, since it manages and organizes all the information about users, hotels, and bookings. It is designed to handle user registrations, store hotel details, and maintain booking records in a reliable and efficient way. The system must allow administrators to add, edit, or remove hotel information while keeping the data consistent. At the same time, it must ensure secure storage of sensitive user data such as passwords and prevent incomplete booking information.[7]

3.2 Entity Relationship Diagram (ERD):

The database consists mainly of three entities: User, Hotel, and Booking. A user can make multiple bookings, while each booking is linked to a specific hotel. Similarly, a hotel can have multiple bookings associated with it. This relationship creates a structure where the User and Hotel entities connect through the Booking entity. The ERD clearly illustrates these connections, showing how data flows between the different parts of the system.[8]

users		
int	id	PK
string	name	
string	mobile	
string	password	
string	gender	
date	timestamp	
hotels		
int	id	PK
string	name	
text	description	
string	city	
decimal	rate	
decimal	mrp	
decimal	discount	
decimal	location	
decimal	lat	
string	log	
text	services	
text	food	
text	poster	
text	roomaddhotelimages	
date	timestamp	
book		
int	id	PK
int	hotelID	FK
string	hotelName	FK
date	startDate	
date	endDate	
int	userID	
decimal	price	
int	peopleValue	
int	nights	
decimal	discount	
text	bookingName	
text	bookingEmail	
text	bookingPhone	
text	type	
date	create_at	

Figure-01: ER Diagram (attributes)

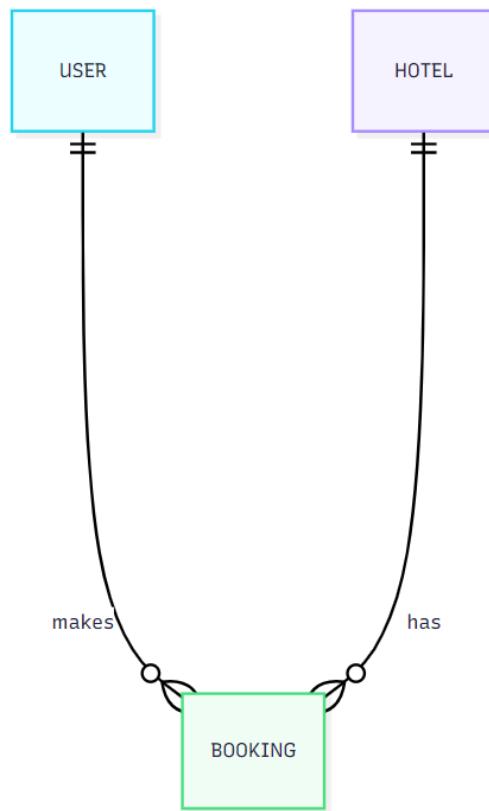


Figure-02: ER Diagram (relations)

3.3 Normalization:

To ensure efficiency and remove redundancy, the database was normalized up to the third normal form. In the first stage, repeating groups were eliminated so that every field contained only atomic values. In the second stage, all non-key attributes were made dependent only on the primary key, which avoided partial dependencies. Finally, in the third stage, transitive dependencies were removed so that attributes depended directly on the primary key and not on other non-key fields.

Showing rows 0 - 0 (1 total, Query took 0.0003 seconds.) [id: 2... - 2...]																																																		
SELECT b.* , h.* FROM `book` AS b INNER JOIN hotels AS h ON b.hotelID = h.id WHERE b.userID = '3' ORDER BY b.id DESC;																																																		
<input type="checkbox"/> Profiling Edit inline Edit Explain SQL Create PHP code Refresh																																																		
<input type="checkbox"/> Show all Number of rows: 25 Filter rows: Search this table																																																		
<input type="checkbox"/> Extra options																																																		
<table border="1"> <thead> <tr> <th id<="" th=""><th>hotelID</th><th>hotelName</th><th>startDate</th><th>endDate</th><th>userID</th><th>price</th><th>peopleValue</th><th>discount</th><th>bookingName</th><th>bookingEmail</th><th>bookingPhone</th><th>createAt</th><th>id</th><th>name</th><th>description</th><th>city</th><th>rate</th></th></tr> </thead> <tbody> <tr> <td>2</td><td>18</td><td>Comfort Suites</td><td>2025-09-08</td><td>2025-09-10</td><td>3</td><td>5600</td><td>3</td><td>8</td><td>Nafisha Nawar</td><td>nawarnafisha17@gmail.com</td><td>1234</td><td>2025-09-09 00:24:00</td><td>18</td><td>Affordable family-friendly hotel</td><td>Canada</td><td>4.30</td></tr> </tbody> </table>																<th>hotelID</th> <th>hotelName</th> <th>startDate</th> <th>endDate</th> <th>userID</th> <th>price</th> <th>peopleValue</th> <th>discount</th> <th>bookingName</th> <th>bookingEmail</th> <th>bookingPhone</th> <th>createAt</th> <th>id</th> <th>name</th> <th>description</th> <th>city</th> <th>rate</th>	hotelID	hotelName	startDate	endDate	userID	price	peopleValue	discount	bookingName	bookingEmail	bookingPhone	createAt	id	name	description	city	rate	2	18	Comfort Suites	2025-09-08	2025-09-10	3	5600	3	8	Nafisha Nawar	nawarnafisha17@gmail.com	1234	2025-09-09 00:24:00	18	Affordable family-friendly hotel	Canada	4.30
<th>hotelID</th> <th>hotelName</th> <th>startDate</th> <th>endDate</th> <th>userID</th> <th>price</th> <th>peopleValue</th> <th>discount</th> <th>bookingName</th> <th>bookingEmail</th> <th>bookingPhone</th> <th>createAt</th> <th>id</th> <th>name</th> <th>description</th> <th>city</th> <th>rate</th>	hotelID	hotelName	startDate	endDate	userID	price	peopleValue	discount	bookingName	bookingEmail	bookingPhone	createAt	id	name	description	city	rate																																	
2	18	Comfort Suites	2025-09-08	2025-09-10	3	5600	3	8	Nafisha Nawar	nawarnafisha17@gmail.com	1234	2025-09-09 00:24:00	18	Affordable family-friendly hotel	Canada	4.30																																		
<input type="checkbox"/> Show all Number of rows: 25 Filter rows: Search this table																																																		

Figure-03: Normalization

3.4 Table Structures:

The database contains three main tables: **users**, **hotels**, and **book**.

The Users table stores information such as id, name, mobile, password (stored in a hashed format for security), and gender. The Hotels table keeps hotel-related information including hotel ID, name, city, description, price, discount etc. The book table records booking details, which include the booking ID, the ID of the user making the booking, the ID of the hotel being booked, the booking date, number of people, and the booking status (such as pending, verified, or cancelled). Together, these tables cover the essential requirements of the system and maintain clear relationships among users, hotels, and bookings.

Table-01: users

The screenshot shows the 'Structure' tab for the 'users' table in MySQL Workbench. The table has six columns: id, name, mobile, password, gender, and timestamp. The 'id' column is defined as int(255) with AUTO_INCREMENT, 'name' as varchar(255), 'mobile' as varchar(255), 'password' as varchar(255), 'gender' as enum('Male', 'Female'), and 'timestamp' as timestamp with current_timestamp(). Other columns like name, mobile, password, gender, and timestamp have utf8mb4_general_ci collation. The 'Actions' column provides options to change, drop, or view more details for each column.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(255)			No	None		AUTO_INCREMENT	Change Drop More
2	name	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
3	mobile	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
4	password	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
5	gender	enum('Male', 'Female')	utf8mb4_general_ci		No	Male			Change Drop More
6	timestamp	timestamp			No	current_timestamp()			Change Drop More

Table-02: hotels

The screenshot shows the 'Structure' tab for the 'hotels' table in MySQL Workbench. The table has sixteen columns: id, name, description, city, rate, mrp, discount, location, lat, log, services, food, poster, room_addHotellImages, room, and timestamps. The 'id' column is defined as int(11) with AUTO_INCREMENT, 'name' as varchar(255), 'description' as text, 'city' as varchar(255), 'rate' as decimal(10,2), 'mrp' as decimal(10,2), 'discount' as decimal(5,2), 'location' as varchar(255), 'lat' as decimal(9,6), 'log' as decimal(9,6), 'services' as text, 'food' as text, 'poster' as text, 'room_addHotellImages' as text, 'room' as int(255), and 'timestamps' as timestamp with current_timestamp(). Other columns like name, description, city, location, services, food, poster, room, and timestamps have utf8mb4_general_ci collation. The 'Actions' column provides options to change, drop, or view more details for each column.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop
2	name	varchar(255)	utf8mb4_general_ci		No	None			Change Drop
3	description	text	utf8mb4_general_ci		No	None			Change Drop
4	city	varchar(255)	utf8mb4_general_ci		No	None			Change Drop
5	rate	decimal(10,2)			No	None			Change Drop
6	mrp	decimal(10,2)			No	None			Change Drop
7	discount	decimal(5,2)			No	None			Change Drop
8	location	varchar(255)	utf8mb4_general_ci		No	None			Change Drop
9	lat	decimal(9,6)			No	None			Change Drop
10	log	decimal(9,6)			No	None			Change Drop
11	services	text	utf8mb4_general_ci		No	None			Change Drop
12	food	text	utf8mb4_general_ci		No	None			Change Drop
13	poster	text	utf8mb4_general_ci		No	None			Change Drop
14	room_addHotellImages	text	utf8mb4_general_ci		No	None			Change Drop
15	room	int(255)			No	None			Change Drop
16	timestamps	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()	Change Drop

Table-03: book

The screenshot shows the MySQL Workbench interface with the 'book' table selected. The table has 15 columns:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	<code>id</code>	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	<code>hotellID</code>	int(11)			No	None			Change Drop More
3	<code>hotelName</code>	longtext	utf8mb4_general_ci		No	None			Change Drop More
4	<code>startDate</code>	date			No	None			Change Drop More
5	<code>endDate</code>	date			No	None			Change Drop More
6	<code>userID</code>	int(11)			No	None			Change Drop More
7	<code>price</code>	int(11)			No	None			Change Drop More
8	<code>peopleValue</code>	int(11)			No	None			Change Drop More
9	<code>nights</code>	int(11)			No	None			Change Drop More
10	<code>discount</code>	int(11)			No	None			Change Drop More
11	<code>bookingName</code>	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
12	<code>bookingEmail</code>	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
13	<code>bookingPhone</code>	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
14	<code>type</code>	varchar(255)	utf8mb4_general_ci		No	Pending			Change Drop More
15	<code>create_at</code>	timestamp			No	current_timestamp()			Change Drop More

3.5 Database Integration with Backend:

The backend of the system was connected to the database using PHP with MySQL queries. When a user signs up, their information is inserted into the ‘users’ table. When they search for hotels, PHP scripts fetch the required records from the ‘hotels’ table based on the selected city. Similarly, when a booking is made, a new record is added to the ‘book’ table with references to both the user and the hotel.

The administrator can also update hotel information or change the status of bookings directly through the admin panel, which internally updates the related database tables. This integration ensures smooth communication between the frontend, backend, and database, creating a fully functional system.

All integrated connections were made through specific routes, config and db_diver connection files.

Chapter 4: System Design

4.1 Use Case Diagram:

The use case diagram highlights the functional requirements of the system by showing the interactions between users, admin, and the website. The system consists of two major actors: the user and the admin. The user is able to create an account through the signup feature, log in to the system, search for hotels based on city, date, and number of people, and complete bookings. The administrator, on the other hand, is responsible for managing the website, adding or editing hotel details, verifying booking requests and bookings through the dashboard.

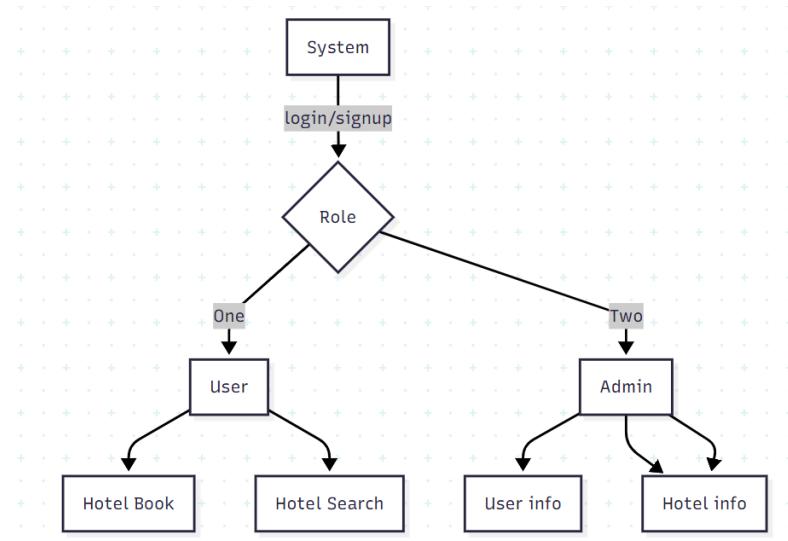


Figure-04: Use Case Diagram

4.2 Class Diagram:

The class diagram illustrates the structure of the system by representing its classes and their relationships. The main classes in the system include User, Hotel, Booking, and Admin.

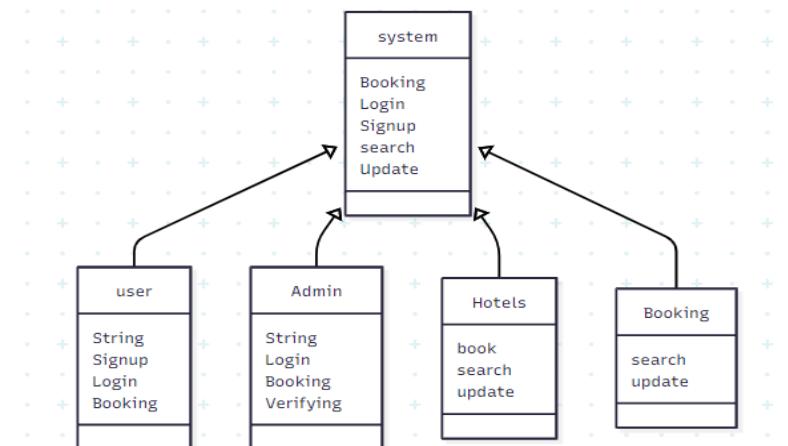


Figure-05: Class Diagram

4.3 User Interface (UI/UX) Design:

The user interface of the system has been designed to be simple, intuitive, and user-friendly. The homepage allows users to search for hotels by specifying the city, travel date, and number of people. A dedicated login and signup page ensures secure authentication for both users and administrators. [9]

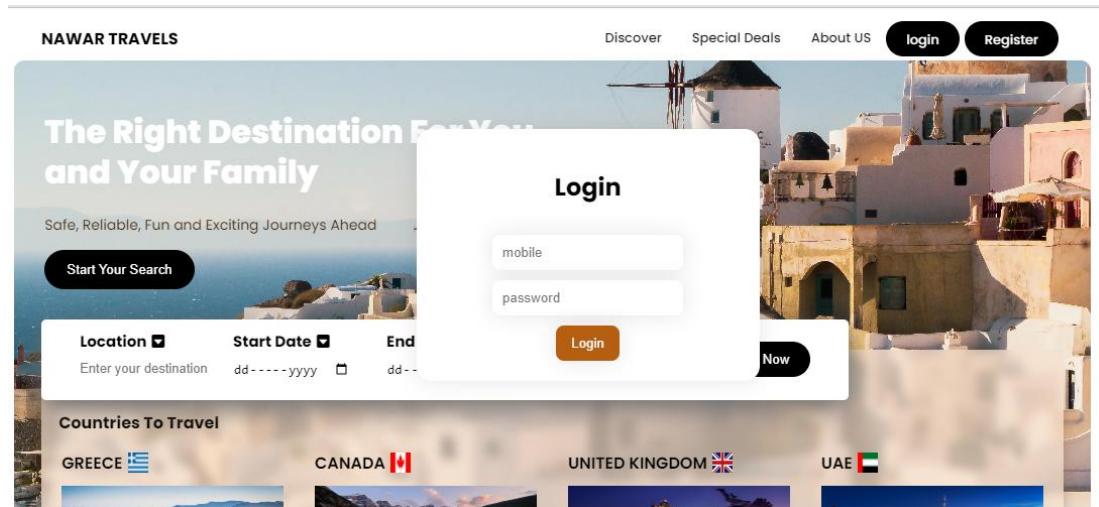


Figure-6.1: UI Design Demo (user panel)

Once logged in, the user is presented with a personal Booking Tab displaying booking history.

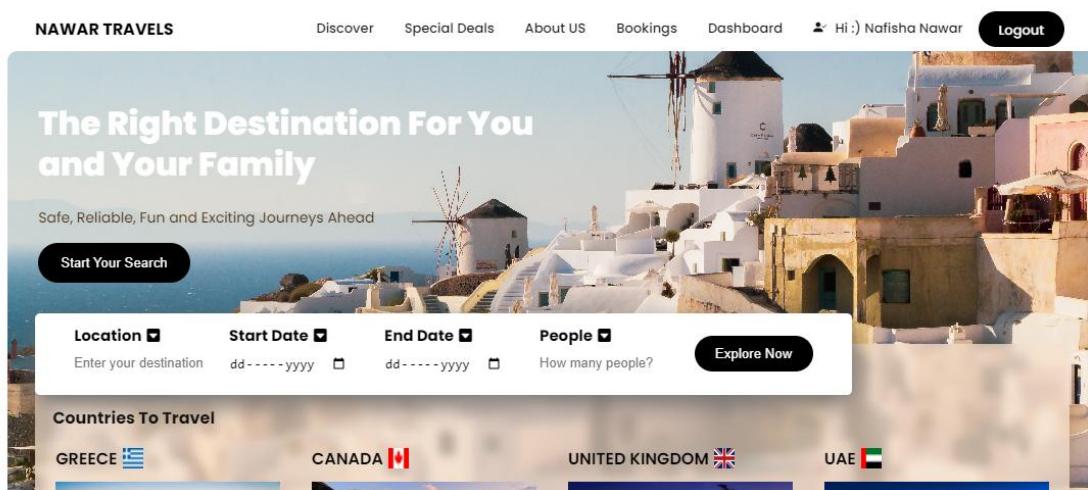


Figure-6.2: UI Design Demo (2) (user panel)

The administrator is provided with a separate dashboard, which summarizes key statistics such as the number of users, hotels, and bookings. The administrator can also access options to add or edit hotel records and verify bookings. [10]

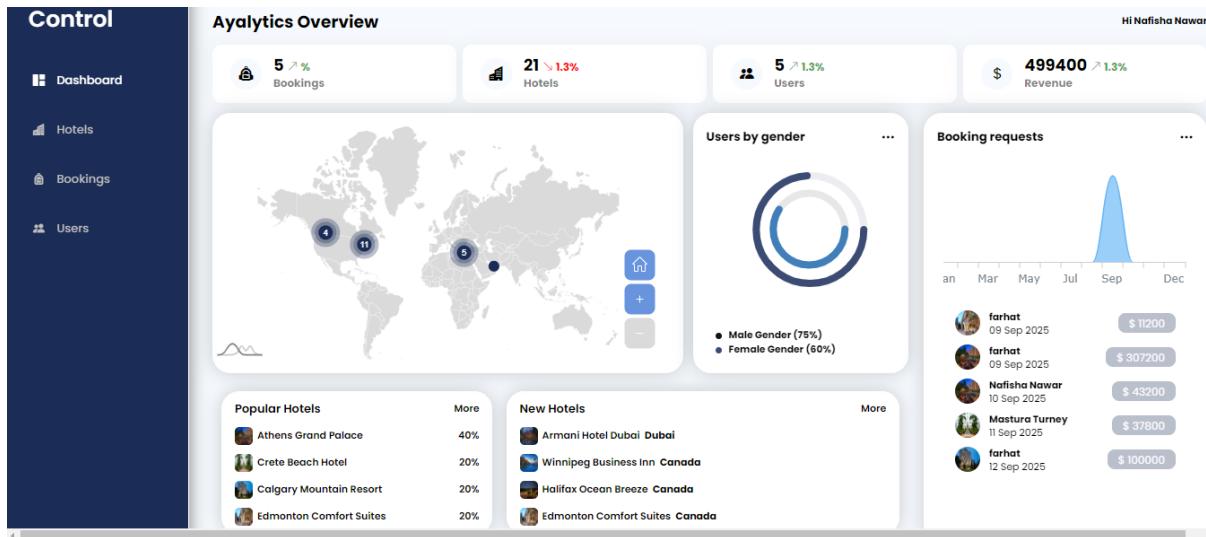


Figure-6.3: UI Design Demo (1) (admin panel)

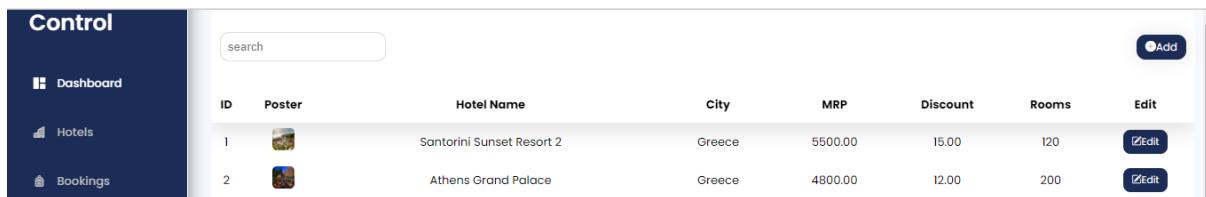


Figure-6.4: UI Design Demo (2) (admin panel)

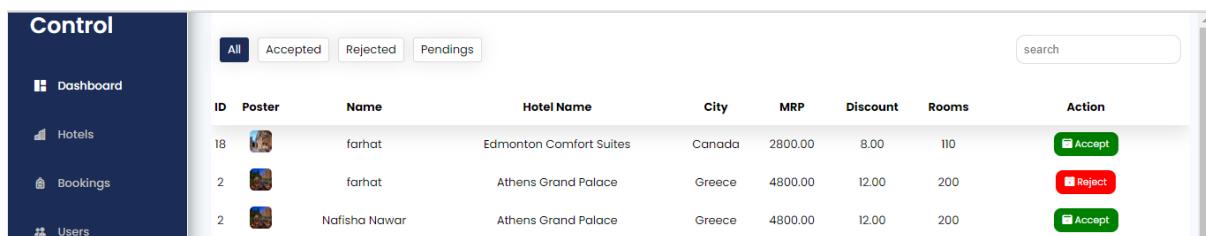


Figure-6.5: UI Design Demo (3) (admin panel)

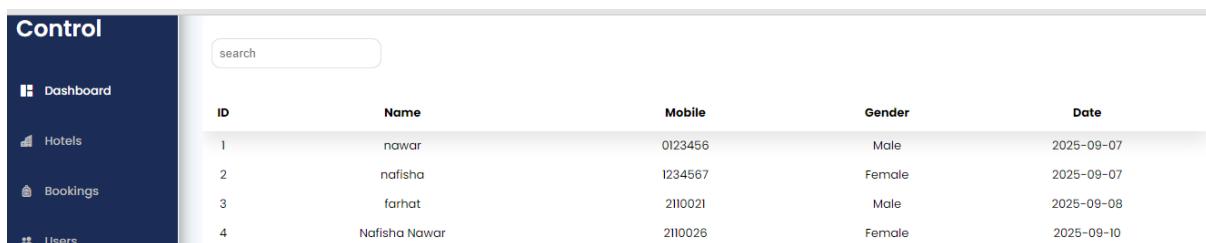


Figure-6.6: UI Design Demo (4) (admin panel)

The interface follows responsive design principles so that it can be accessed from different devices, ensuring convenience for both users and administrators.

Chapter 5: System Implementation

5.1 Introduction

This chapter describes the implementation process of the system, focusing on how the different modules were developed and integrated. The implementation covers both the frontend design and backend logic, along with the connection to the database.

5.2 Module-wise Implementation.

The system was implemented in separate modules, each responsible for a particular feature. Among them, the Authentication Module is one of the most important, since it allows users to create an account and securely log in before accessing other features of the website.

For the signup functionality, the system collects user details such as name, mobile number, password, and gender through the signup form. When the user clicks the signup button, an event listener captures the input values and stores them in a **FormData** object. This data is then sent to the backend using the fetch API with the **HTTP POST** method. The backend processes the request, stores the information in the database, and returns a **JSON** response. Once the response is received, the page reloads to reflect the successful signup.

The login module follows a similar implementation pattern. When a user enters their mobile number and password, these values are also appended to a **FormData** object and sent to the backend through a **POST** request. The backend validates the credentials against the stored records in the database. If the information is correct, the backend responds with a success message, after which the system reloads the page and grants the user access to their account dashboard. If the credentials are invalid, an error message is returned instead.

This modular approach ensures a clear separation of responsibilities: the frontend is responsible for capturing input data and sending requests, while the backend handles authentication and database interaction. By reloading the page after successful signup or login, the system maintains consistency and provides immediate feedback to the user.

5.3 Backend Feature Implementation

The backend of the system was implemented using PHP in combination with a MySQL database. The login system verifies user credentials against stored records in the database, while session management ensures that users remain authenticated during their activity. SQL queries are used to retrieve hotel data and insert new booking records. For administrative tasks, queries were developed to insert and update hotel records and to modify booking statuses after verification.

5.4 Code Quality and Structure

The system was developed with emphasis on code clarity and maintainability. The implementation follows a modular approach similar to the Model-View-Controller (MVC) design pattern, separating the presentation layer, business logic, and database operations. Functions were written to handle recurring operations such as input validation and database queries. Comments were included throughout the source code to improve readability and assist in future modifications.

All Frameworks were implemented in one single file but were separated for different modules. Entire coding was completed in one php file for each module. This ensured better obtainability of code.

5.5 Security Features

Security measures were included in the design to protect user data and system integrity. User passwords are hashed before being stored in the database. These features enhance the reliability of the system and safeguard sensitive information but is certainly not enough. The website not properly developed in security aspects. Rather than more focused on implementing more features.

Chapter 6: Testing and Debugging

6.1 Testing Strategy

Testing was conducted at various stages of development to ensure the correctness and reliability of the system. Unit testing was performed to evaluate individual functions such as login verification and hotel search. As each feature were added, each time it had been tested manually.

Integration testing was used to confirm that different modules communicated correctly, particularly between the frontend and the backend database. Finally, system testing was carried out to evaluate the complete system under real usage scenarios.

All of the testing measures were carried out separately and manually while developing the system.

6.2 Test Cases and Results

Several test cases were executed to verify the accuracy of system functions. The login process was tested with both correct and incorrect credentials. The system successfully redirected valid users to their dashboard and displayed appropriate error messages for invalid attempts. The hotel search module was tested by entering different cities and dates, which correctly displayed matching hotels. The booking process was tested to confirm that reservations were stored in the database and displayed in the user dashboard. In all cases, the expected results matched the actual system behavior.

6.3 Debugging and Error Handling

During implementation, several issues were identified and resolved. One recurring problem was related to session persistence, where users were not being retained in the system after login. This was corrected by modifying session handling logic. SQL errors in booking queries were debugged by carefully analyzing query structures. Additional validation was added to forms to prevent users from submitting empty or invalid fields, thereby improving system stability.

Frontend issues were inspected and resolved. For Backend, each query was run in the database before implementing on the project. JavaScript errors were handled using try-catch exception handling method and errors were consoled in the inspect window.

6.4 Performance Evaluation

The performance of the system was measured in terms of Basic and relevant features, Working modules, Frontend Design and mostly Backend works. The behavior of the system was observed and its's response to each features contributed to the proper evaluation.

6.5 Limitations and Future Improvements

Although the system is functional, some limitations remain. At present, the system does not include an online payment gateway. Future enhancements could integrate secure payment systems to complete the booking cycle.

Currently, the website is limited to hotel bookings only, but it could be extended to include flight reservations and full travel packages. A mobile application could also be developed in the future to improve accessibility for users.

The admin panel is confined to just one administrator. Multiple access of admins in future could be an additional feature.

The system could be further developed in security aspects to prevent user data and booking data leakage and provide a safe and reliable system. Such as: In login-registration, verification through emails or security code, after booking a trip, the user receiving a confirmation email, notifying users for their upcoming trips.

There are further scope for the project to be integrated with database of each hotels included for better managing of the booking and letting users have a freedom in terms of room booking and additional facilities.

Chapter 7: Project Management and Participation

7.1 Development Methodology

The system was developed following the Agile methodology. The project was divided into several features which can be called sprints, each focusing on a particular set of tasks.

The first sprint concentrated on UI design and Style of the ‘Discover’ page. After designing a functioning front-end, the authentication system or login/signup was designed as 2nd sprint. The 2nd sprint solely focused on the backend aspect of the entire authentication system.

The 3rd sprint developed the hotel search and booking modules. Frontend design for this module was done alongside backend development.

The 4th sprint focused on the administrative dashboard and hotel management features. The final sprint included comprehensive testing and deployment. This iterative approach allowed continuous refinement and adaptation of requirements throughout development.

There was no scope for proper user feedback. Although it was somewhat implemented by respected supervisor after each feature were added.

7.2 Timeline

The project was implemented over a period of 14 weeks. During the first two weeks, requirement analysis and database design were completed. Weeks three and five were dedicated to frontend design and implementation. Backend development and database integration were carried out in weeks six and nine. The administrator panel and dashboard were implemented in week ten to eleven. The final weeks were allocated for testing, debugging, and final documentation.

7.3 Team Roles and Contributions

The development of the project involved only one contributor as it was an individual project. while the respected project supervisor coordinated the tasks, ensured adherence to the timeline, and reviewed deliverables.

7.4 Attendance and Participation

Since the project was carried out individually, attendance was monitored through periodic check-ins with assigned supervisor. Approximately every two to three weeks, the supervisor scheduled review sessions to evaluate the progress of the system. During these sessions, the current state of the project was demonstrated, and the supervisor provided constructive feedback on both the functionality and design. Suggestions were given for additional features that could improve the usability of the system.

Chapter 8: Documentation and Report

8.1 Objectives and Design Rationale

The objective of the documentation is to explain the system design, implementation, and testing process in detail. The project team chose PHP and MySQL for backend development due to their wide availability and ease of integration. HTML5, JavaScript and CSS3 were selected for frontend design because it provides freedom of design and implementation from scratch. XAMPP was used as the local server environment for testing and development.[11]

8.2 Screenshots and Results:

Discover Page:

The screenshot shows the homepage of NAWAR TRAVELS. At the top, there's a navigation bar with links for 'Discover', 'Special Deals', 'About US', and buttons for 'login' and 'Register'. The main header reads 'The Right Destination For You and Your Family' with a subtext 'Safe, Reliable, Fun and Exciting Journeys Ahead'. Below this is a search bar with fields for 'Location', 'Start Date', 'End Date', and 'People', along with a 'Explore Now' button. A large image of a Greek island town with windmills serves as the background. Below the search bar, there's a section titled 'Countries To Travel' featuring four destinations: Greece (Santorini \$560), Canada (Mountain Lake \$600), United Kingdom (Tower of London \$700), and UAE (Dubai \$800). Each destination has a thumbnail image, a 'Read Now' button, and a price.

This screenshot shows a section titled 'Best Tour Pacakge Offers For You' with the subtext 'Choose Your Next Destination'. It features four travel packages: 'Niagara Falls', 'Emporio Castle', 'Grand Canal', and 'Hanalei Bay'. Each package includes a thumbnail image, a list of included services ('Air ticket, Hotel, Breakfast, Tours, Airport Transfer'), and a summary of reviews (96415, 8586) and price (\$2114, \$2648, \$2880, \$3984). There are also 'More Info' and '7 Days' buttons for each package.

Our Destination

Choose your next destination

USA

Canada

Dubai

Canada

Venice

Greece

Included: Air ticket, Hotel, Breakfast, Tours, Airport Transfer

[MORE INFO](#)



Subscribe & get special discount

All Exciting Offers in One Place.

Enter your email address

subscribe

Search page:

NAWAR TRAVELS

Discover Special Deals About US

[login](#)

[Register](#)

The Right Destination For You and Your Family

Safe, Reliable, Fun and Exciting Journeys Ahead

Location

greece

Start Date

14 - Sep - 2025

End Date

17 - Sep - 2025

People

4

[Explore Now](#)

[Filter](#)

[Hotels → greece](#)

Room Amenities

- Pool
- Spa
- WiFi
- AC
- Parking
- Gym
- Restaurant

Price per night

- Under \$1000
- Under \$2000



Santorini Sunset Resort 2

★★★★★

Luxury resort with a view of the Caldera

Pool Spa WiFi AC

Oia, Santorini

Very Good 4.8

120 Rooms Left

\$5500 / Night

15% Discount

[Book Now](#)



Athens Grand Palace

★★★★★

5-star hotel near Acropolis

Parking Pool Gym WiFi AC

Syntagma Square, Athens

Very Good 4.7

200 Rooms Left

\$4800 / Night

12% Discount

[Book Now](#)

Hotel booking info page:

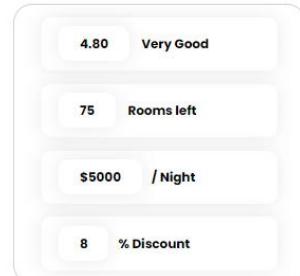
NAWAR TRAVELS

Discover Special Deals About US

[login](#)

[Register](#)

Mykonos Paradise

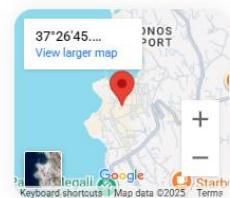


Amenities

- Pool
- Restaurant
- WiFi
- Greek
- Continental

Description

Boutique hotel with nightlife access



Booking Details

Mykonos Paradise

★★★★★ Family Friendly

Mykonos Town, Mykonos

CHECK IN Sun 14 Sept 2025 3 Nights 12 PM

CHECK OUT Wed 17 Sept 2025 12 PM

3 Nights | 4 People | 1 Room

Guest Details

Name Email Mobile Book Now

Subscribe & get special discount

All Exciting Offers in One Place.

Enter your email address subscribe

Hotel booking page:

The screenshot shows a hotel booking interface with two search results displayed:

- Vancouver Seaside Resort** (Family Friendly):
 - Check-in: Mon 15 Sept 2025, Check-out: Tue 30 Sept 2025 (15 Nights)
 - Price: \$92070
 - Discount: \$6200.00 with 15% discount
- Rhodes Heritage Inn** (Family Friendly):
 - Check-in: Sun 14 Sept 2025, Check-out: Wed 17 Sept 2025 (3 Nights)
 - Price: \$10150
 - Discount: \$3500.00 with 10% discount

Admin Dashboard:

The screenshot shows an admin dashboard with the following sections:

- Control** sidebar with links: Dashboard, Hotels, Bookings, Users.
- Analytics Overview** section:
 - Metrics: 7 Bookings (100%), 22 Hotels (100%), 6 Users (100%), \$1925900 Revenue (100%).
 - World map showing booking locations.
 - Donut chart showing User gender distribution: Male Gender (75%) and Female Gender (25%).
 - Booking requests chart showing a peak in September.
 - Popular Hotels: Athens Grand Palace (28%), Crete Beach Hotel (14%), Rhodes Heritage Inn (14%), Vancouver Seaside Resort (14%).
 - New Hotels: Nou hotel dhaka, Armani Hotel Dubai Dubai, Winnipeg Business Inn Canada, Halifax Ocean Breeze Canada.

8.3 Conclusion

The development of the Travel and Tour Booking Website successfully achieved its primary objective of creating a reliable, user-friendly, and secure online platform for hotel booking and travel planning. By integrating modules for user authentication, hotel search, booking management, and an administrative control panel, the system provides a complete solution that benefits both users and travel agencies. The use of HTML, CSS, JavaScript, PHP, and MySQL ensured a responsive frontend and a robust backend capable of handling authentication, data storage, and booking operations efficiently.

Although the current implementation is limited to hotel bookings, the system has strong potential for future expansion. Features such as online payment integration, mobile application development, and the inclusion of flight and tour package bookings could transform it into a complete travel management platform.

In conclusion, this project illustrates how web-based systems can act as a bridge between users and service providers, offering a centralized and efficient medium for hotel booking. It stands as a scalable solution that can be adapted by travel agencies and independent users alike, contributing to the growing digital transformation of the tourism and travel industry.

Chapter 9: References

- [1] “20 Example Travel Websites That Will Get You Moving.” Accessed: Sep. 13, 2025. [Online]. Available: <https://www.crazyegg.com/blog/example-travel-websites/>
- [2] “(1) Create A Responsive Tour & Travel Agency Website Design Using HTML / CSS / JAVASCRIPT | Step By Step - YouTube.” Accessed: Sep. 13, 2025. [Online]. Available: <https://www.youtube.com/watch?v=nPZp-YNPyC8&t=447s>
- [3] “(1) Introduction to HTML, CSS, JavaScript & How websites work? | Web Development Tutorials #1 - YouTube.” Accessed: Sep. 13, 2025. [Online]. Available: https://www.youtube.com/watch?v=6mbwJ2xhgzM&list=PLu0W_9lII9agiCUZYRsvtGTXdxkzPyItg
- [4] “Welcome to CodeIgniter.” Accessed: Sep. 13, 2025. [Online]. Available: <https://codeigniter.com/>
- [5] “Bootstrap Icons · Official open source SVG icon library for Bootstrap.” Accessed: Sep. 13, 2025. [Online]. Available: <https://icons.getbootstrap.com/>
- [6] “Visual Studio Code - Code Editing. Redefined.” Accessed: Sep. 13, 2025. [Online]. Available: <https://code.visualstudio.com/>
- [7] “phpMyAdmin.” Accessed: Sep. 13, 2025. [Online]. Available: <https://www.phpmyadmin.net/>
- [8] “Mermaid | Diagramming and charting tool.” Accessed: Sep. 13, 2025. [Online]. Available: <https://mermaid.js.org/>
- [9] “Free Stock Photos, Royalty Free Stock Images & Copyright Free Pictures · Pexels.” Accessed: Sep. 13, 2025. [Online]. Available: <https://www.pexels.com/>
- [10] “AnyChart is a lightweight and robust JavaScript charting library.” Accessed: Sep. 13, 2025. [Online]. Available: <https://www.anychart.com/>
- [11] “Wallpaper 4k Pictures | Download Free Images on Unsplash.” Accessed: Sep. 13, 2025. [Online]. Available: <https://unsplash.com/s/photos/wallpaper-4k>

Chapter 10: Appendices

Appendix A: Source Code / GitHub Link:

- https://github.com/NawarNafisha/software_project_2.git