UniStay

Project Submitted in Partial fulfilment of the Requirement for the Award of the Degree of

Master of Computer Applications

Semester IV

Jan - May, 2024

Under the guidance of

Dr. Deepak Abhyankar

Submitted By

Aditya Kumar Tiwari (2211305)

Shubhangi Shahi (2211365)

School of Computer Science & IT

Devi Ahilya Vishwavidyalaya, Indore, M.P.

2024

School of Computer Science & IT

Devi Ahilya Vishwavidyalaya, Indore, M.P.

DECLARATION

I hereby declare that the project titled "UniStay" submitted by me for the partial fulfilment of the requirement for the award of *Master of Computer Applications* to *School of Computer Science & IT, Devi Ahilya Vishwavidyalaya, Indore,* comprises my own work and due acknowledgement has been made in text to all other material used.

Signature of Student:

Date:

Place:

School of Computer Science & IT

Devi Ahilya Vishwavidyalaya, Indore, M.P.

CERTIFICATE FROM GUIDE

It is to certify that dissertation on "UniStay," submitted by Mr/Ms *Aditya Kumar Tiwari* to the *School of Computer Science & IT, DAVV, Indore* has been completed under my supervision and the work is carried out and presented in a manner required for its acceptance in partial fulfilment for the award of the degree of *Master of Computer Applications*.

Project Guide Signature:

Name:

Date:

School of Computer Science & IT

Devi Ahilya Vishwavidyalaya, Indore, M.P.

CERTIFICATE

It is to certify that we have examined the dissertation on "UniStay," submitted by Mr./Ms./Mrs. *Aditya Kumar Tiwari* to the *School of Computer Science & IT, DAVV,* Indore and hereby accord our approval of it as a study carried out and presented in a manner required for its acceptance in partial fulfilment for the award of the degree of *Master of Computer Applications*.

Internal Examiner

Signature: _____

Name: _____

Date: _____

External Examiner

Signature: _____

Name: _____

Date: _____

School of Computer Science & IT

Devi Ahilya Vishwavidyalaya, Indore, M.P.

ACKNOWLEDGEMENT

I extend my heartfelt gratitude to *Dr. Deepak Abhyankar* for his unwavering support and invaluable guidance throughout my dissertation. His expertise and supervision were instrumental in shaping this study. I am also thankful to the Head of Department, *Dr. Sanjay Tanwani*, and the class coordinator, *Prof. Tarjani Sevak*, for their support and encouragement.

Additionally, I am grateful to the faculty of the School of Computer Science and IT, Devi Ahilya Vishwavidyalaya, for fostering a conducive learning environment. Special appreciation goes to the respondents and participants whose contributions enriched this research.

Lastly, I express my gratitude to all individuals who assisted me in this academic journey. Your support was invaluable.

Thank you,

Aditya Kumar Tiwari

May 1, 2024

ABSTRACT

The UniStay project presents a web application designed to streamline the process of finding suitable accommodations for students. It offers a user-friendly platform with advanced search filters and map integration, enabling students to explore various options based on their preferences. Real-time availability updates, user reviews, and a secure booking system enhance the reliability of the platform. Mobile responsiveness, security measures, and legal compliance are prioritized, along with a feedback mechanism for continuous improvement. Key features include user registration, comprehensive accommodation listings, search options, map integration, user reviews, real-time availability status, a secure booking system, a notification system, user support, and scalability. The technological stack encompasses Java for backend development, Bootstrap for frontend, MySQL for the database, and GitHub for version control. Backend functionalities include user authentication, database integration, accommodation CRUD operations, search, and filter, matching algorithm, booking system, reviews and ratings, and messaging system. Frontend aspects focus on responsive design, user dashboard, accommodation listings, search interface, booking workflow, review section, messaging interface, and admin panel. General considerations cover security, scalability, performance, error handling, documentation, testing, compliance, and deployment. UniStay aims to provide a centralized, transparent, and secure platform catering to the needs of students and accommodation providers, fostering a supportive community, and ensuring user satisfaction through ongoing collaboration and adherence to best practices.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

The UniStay project introduces a comprehensive web application tailored to meet the accommodation needs of students. With a focus on user-friendliness, advanced features, and security, UniStay aims to simplify the process of finding suitable accommodations such as rooms, PGs, and hostels.

Key Features:

- User registration and profile creation facilitate personalized searches based on preferences.
- Accommodation listings offer detailed information, including rent, amenities, and location, with visual aids like images and virtual tours.
- Advanced search and filter options allow users to refine their search based on various criteria.
- Map integration provides location context, helping users assess proximity to educational institutions and amenities.
- User reviews and ratings ensure transparency and reliability, supported by a verification system.
- Real-time availability updates and a secure booking system streamline the reservation process.
- A notification system keeps users informed about new listings, price changes, and deadlines.
- User support is provided through a help center or chat support.

Technological Stack:

- Backend development utilizes Java for robust infrastructure.
- Frontend design employs Bootstrap (HTML, CSS, and JavaScript) for a visually appealing and responsive interface.
- MySQL serves as the relational database for efficient data management.
- GitHub facilitates version control.

Conclusion:

UniStay is committed to offering a centralized, secure, and user-centric platform for students and accommodation providers. By prioritizing mobile responsiveness, security measures, and legal compliance, UniStay ensures a

seamless and trustworthy experience. With ongoing collaboration, adherence to best practices, and a focus on user satisfaction, UniStay aims to foster a supportive community and become the go-to solution for accommodation search needs.

CHAPTER I

INTRODUCTION

"UniStay" is a Java-based web application that helps students identify acceptable lodging alternatives. With a rising need for centralized platforms that ease the search process, the goal of this project is to design a user-friendly interface that allows students to quickly explore various housing options such as rooms, PGs, and hostels according to their tastes and needs. The development process consists of gathering requirements, designing the interface, selecting a Java framework, implementing frontend and backend logic, integrating with a database, ensuring user authentication and authorization, enabling accommodation listings with search, and filtering functionalities, incorporating reviews and ratings features, implementing a messaging system, ensuring responsive design, thorough testing, deployment, and ongoing maintenance and updates. The application strives to provide a full solution.

1.1 Aim:

The goal of this project is to provide a centralized platform that will streamline the accommodation search process for students by providing up-to-date information and detailed data. It prioritizes security and legal compliance, improves communication and decision-making, integrates maps, and is scalable. The platform has a marketing plan to draw users and a feedback system for enhancements. Its goal is to create a vibrant community that caters to the requirements of students while also helping lodging providers.

1.2. Scope:

UniStay is a web application designed to help students find suitable accommodation options. It streamlines the search process and provides a centralized platform for both students and accommodation providers. Key components include:

1.2.1 Accommodation Types:

UniStay covers a range of accommodation types, including rooms, paying guest accommodations (PGs), and hostels, ensuring flexibility to meet the diverse needs of students.

1.2.2 User registration and profiles:

The platform allows users to register and create profiles, enabling a personalized and tailored search experience based on individual preferences and requirements.

1.2.3 Accommodation Listings:

UniStay features a comprehensive database of accommodation options with detailed information, including rent, amenities, and location, providing students with a wide array of choices.

1.2.4 Search and Filter Functionality:

Advanced search filters empower users to refine their search based on factors such as location, budget, amenities, and type of accommodation, facilitating an efficient and targeted search process.

1.2.5 Real-Time Availability Updates:

Accommodation providers can update real-time availability status, ensuring that students have access to the latest and most accurate information on the availability of accommodations.

1.2.6 Map Integration:

The integration of maps allows users to visualize the exact location of accommodations, aiding in the assessment of proximity to educational institutions, public transportation, and essential services.

1.2.7 User reviews and ratings:

UniStay includes a user review and rating system to provide insights into the experiences of previous tenants, enhancing the decision-making process for prospective tenants.

1.2.8 Secure Booking System:

The platform incorporates a secure and user-friendly booking system, enabling students to reserve accommodations directly through UniStay.

1.2.9 Messaging System:

UniStay features a communication platform that facilitates interaction between students and accommodation providers, allowing for inquiries, clarifications, and other communication related to the accommodation.

1.2.10 Notification System:

A notification system keeps users informed about new listings, price changes, and other relevant updates, enhancing user engagement.

1.2.11 Feedback Mechanism:

UniStay includes a feedback mechanism to collect user opinions and suggestions, fostering continuous improvement and enhancing the overall user experience.

1.2.12 Mobile Responsiveness:

The platform is designed to be responsive across various devices, ensuring accessibility and usability on smartphones, tablets, and desktops.

1.2.13 Security Measures:

Robust security measures are implemented to protect user data, payment information, and the integrity of the UniStay platform.

1.2.14 Legal Compliance:

UniStay ensures compliance with local laws and regulations related to accommodation services.

1.2.15 Scalability:

The UniStay platform is designed with scalability in mind to accommodate potential growth in the number of users and accommodation listings.

CHAPTER II

ANALYSIS

2.1 Problem Statement:

Despite the increasing need for convenient and affordable housing options for students, there is still a lack of centralized and user-friendly platforms that effectively connect tenants and landlords. Existing solutions often fall short in addressing the unique needs of student renters, resulting in fragmented and inefficient rental experiences. Therefore, there is a clear necessity for a comprehensive and dynamic room rental platform specifically tailored to students. This platform would offer a seamless interface for both tenants and landlords to list, search, and secure rental accommodations, while also providing essential features such as roommate matching, secure payment processing, and transparent communication channels.

2.2 Project Objectives:

A project objective is a measurable, achievable, relevant, and time-bound goal set within a defined period, serving as a guiding principle for the project, ensuring alignment and focus throughout the project lifecycle, and measuring its success.

2.2.1 Developing a User-Friendly Interface:

- Design an intuitive and easy-to-navigate interface that caters to both tenants and landlords, ensuring seamless navigation throughout the platform.

2.2.2 Enable Property Listing and Searching:

- Empower landlords to showcase their rental properties with comprehensive descriptions, high-quality images, and a list of amenities to attract potential tenants.
- Enable students to effortlessly search for rental properties based on various filters such as location, price range, amenities, and other relevant criteria.

2.2.3 Enable Secure Payment Processing:

- Incorporate a reliable and secure payment processing system that allows for hassle-free rental payments, security deposits, and other financial transactions.

- Utilize trusted payment gateways and encryption technologies to protect users' financial data and ensure secure transactions.

2.2.4 Ensure Data Privacy and Security:

- Prioritize user data privacy by implementing stringent data protection measures and adhering to relevant privacy regulations.
- Employ robust security protocols, including encryption and authentication mechanisms, to safeguard users' personal information and financial data from unauthorized access or breaches.

2.2.5 Provide Transparent Rental Information:

- Ensure transparency in rental information by providing detailed property listings with accurate pricing, lease terms, and property details.
- Offer tools for landlords to update their listings regularly and respond promptly to inquiries, fostering trust and transparency in the rental process.

2.3 Scope and Limitations:

2.3.1 Scope:

Functionalities:

- The platform will support landlords in listing their rental properties, providing comprehensive details such as property name, description, location, price, and amenities. Students will be empowered to search for rental properties based on their preferences, accessing property details, and engaging in communication with landlords.

User Interface:

- An intuitive and user-friendly interface will be the hallmark of the platform, catering to both tenants and landlords.

Data Privacy and Security:

- Stringent measures will safeguard users' personal information and financial data, complying with relevant data protection regulations and standards. The platform will prioritize confidentiality and integrity, ensuring user data remains secure.

2.3.2 Limitations:

Geographical Limitation:

- Initial focus on a specific geographical area or region may restrict the availability of rental properties and users outside of that area. Expansion plans may include consideration for other regions in future iterations.

Resource Constraints:

- Resource constraints, encompassing time, budget, and personnel limitations, may impact development efforts. Priority will be given to essential features and functionalities within the available resources.

Technical Limitations:

- Technical challenges such as network connectivity, server capacity, and software compatibility may affect performance and scalability. Scalability will be a key consideration as the platform expands in terms of users and rental properties.

2.4 Requirement Analysis:

One of the most important stages of the software development lifecycle is requirement analysis, or requirements gathering, which entails identifying, recording, evaluating, and ranking the needs and expectations of stakeholders for a software system. Clearly defining the functions and characteristics that the system should have been the aim.

2.4.1 Functional Requirements:

User Registration and Authentication:

- Tenants and landlords should be able to register for an account with the platform and authenticate themselves securely.
- Password hashing and salting should be implemented for secure storage of user credentials.

Property Listing:

- Landlords should be able to list their rental properties on the platform, providing detailed information such as property name, description, location, price, and amenities.

- The ability to upload multiple images for each property listing should be supported.

Property Searching:

- Tenants should be able to search for rental properties based on location, price range, property type, and other relevant criteria.
- Advanced search filters, such as amenities, availability dates, and distance from campus, may be included.

2.4.2 Non-Functional Requirements:

Usability:

- The platform should have an intuitive and user-friendly interface, accessible from both desktop and mobile devices.
- Navigation should be straightforward, with clear instructions and guidance provided throughout the user journey.

Performance:

- The platform should be responsive and able to handle concurrent user interactions efficiently, without significant delays or downtime.
- Page load times and response times should be optimized to enhance the user experience.

Security:

- Robust data privacy and security measures should be implemented to protect users' personal information, financial data, and communication channels.
- Secure sockets layer (SSL) encryption should be used to secure data transmission over the internet.

Scalability:

- The platform should be designed to accommodate growth in terms of users and rental properties over time.
- Scalability considerations should be considered for database architecture, server infrastructure, and application design.

2.5 Technology Stack:

Technology Stack refers to all the programming languages, frameworks, libraries, tools, and infrastructure needed to create and manage a software application or system.

2.5.1 Frontend Technologies:

- HTML5 and CSS3
- JavaScript
- Bootstrap

2.5.2 Backend Technologies:

- Java
- Java Server Pages (JSP)
- Servlets

2.5.3 Database:

- MySQL (8.0.36)

2.5.4 Server Environment:

- Apache Tomcat (10.1)
- Java EE

2.6 Feasibility Study:

A feasibility study, which is carried out to evaluate the viability of a proposed project or initiative, is a crucial phase in the project development process. It highlights possible risks and obstacles and assists stakeholders in deciding if the project is worthwhile.

2.6.1 Technical Feasibility:

- Availability of Technology: Java, JSP, and related technologies are widely used and well-supported for web application development. The availability of frameworks like Spring and libraries like Hibernate simplifies backend development.

- Scalability: The chosen technology stack, including servlets and relational databases, supports scalability to accommodate growing user bases and property listings.

2.6.2 Economic Feasibility:

- Cost of Development: While the development cost may vary based on factors like team size, expertise, and development time, using open-source technologies and cloud services can help minimize expenses.
- Revenue Generation: The platform can generate revenue through various channels such as subscription fees for landlords, advertising, and commission on successful rental transactions.

2.6.3 Operational Feasibility:

- User Adoption: A user-friendly interface, robust features, and effective marketing strategies can drive user adoption among tenants and landlords.
- Maintenance: Regular maintenance and updates are essential to keep the platform secure, functional, and competitive. Implementing automated processes and monitoring tools can streamline maintenance efforts.

CHAPTER III

PROJECT PLANNING

3.1 Project Scope:

UniStay is a web application designed to help students find suitable accommodation options. It streamlines the search process and provides a centralized platform for both students and accommodation providers. Key components include:

3.1.1 User Roles:

- Tenants: Students or individuals looking for rental accommodations.
- Landlords: Property owners or managers listing rental properties.

3.1.2 Core Features:

User Registration and Authentication:

- Tenants and landlords can register for accounts with the platform.
- Secure authentication mechanisms ensure user privacy and security.

Property Listing and Searching:

- Landlords can list rental properties with details such as property name, description, location, price, amenities, and images.
- Tenants can search for rental properties based on location, price range, property type, and amenities.

3.1.3 Additional Features:

- Advanced Search Filters: Additional search filters such as availability dates, and distance from campus enhance the search experience.
- Tenant Reviews and Ratings: Tenants can leave reviews and ratings for rental properties and landlords, providing valuable feedback for other users.
- Responsive Design: The platform is responsive and accessible across various devices, including desktops, laptops, tablets, and smartphones.

3.1.4 Out of Scope:

- Property Management: Full property management features, such as maintenance requests and tenant management, are out of scope for this project.
- Legal Advice: The platform does not provide legal advice or mediate legal disputes between tenants and landlords.
- Real-Time Property Availability: Real-time availability tracking for rental properties may be considered for future iterations but is not included in the initial scope.

3.2 Project Goals and Objectives:

To steer the project in the right direction and guarantee its success, well-defined project goals and objectives are important.

3.2.1 Develop a User-Friendly Platform:

- Objective 1: Design intuitive and responsive UI for tenants and landlords.
- Objective 2: Ensure accessibility across various devices (desktops, laptops, tablets, smartphones).
- Objective 3: Implement streamlined user experience from registration to property search and communication.

3.2.2 Facilitate Rental Property Listings and Searches:

- Objective 1: Enable landlords to list properties with comprehensive details (name, description, location, price, amenities, images).
- Objective 2: Allow tenants to search properties based on criteria (location, price range, property type, amenities).
- Objective 3: Provide advanced search filters and sorting options for an enhanced search experience.

3.2.3 Ensure User Satisfaction and Engagement:

- Objective 1: Gather user feedback via surveys, reviews, and usability testing to pinpoint improvement areas.
- Objective 2: Continuously iterate and enhance the platform based on user feedback and market trends.

- Objective 3: Foster community engagement by incorporating features like tenant reviews, ratings, and forums.

3.3 Work Breakdown Structure (WBS):

A Work Breakdown Structure (WBS) helps to organize and break down the project's scope into manageable tasks.

3.3.1 Project Management:

- Define project scope, goals, and objectives
- Risk management plan and mitigation strategies
- Project scheduling and resource allocation
- Progress tracking and reporting mechanisms

3.3.2 Frontend Development:

- Design user interface mock-ups and wireframes
- Implement frontend layout and navigation
- Develop user registration and authentication functionality
- Create property listing and searching interfaces
- Develop user profile management interfaces
- Ensure responsive design for various devices and screen sizes

3.3.3 Backend Development:

- Set up server environment
- Develop backend logic for user registration and authentication
- Implement APIs for property listing and searching functionalities
- Integrate database management system
- Implement data validation and security measures

3.3.4 Database Design and Management:

- Database schema design based on project requirements
- Set up and configure database server
- Implement data access layer for CRUD operations
- Optimize database performance and scalability

CHAPTER IV

SYSTEM DESIGN

The system design for UniStay involves defining the architecture, components, and interactions that constitute the platform.

4.1 Architecture:

The term "architecture" is frequently used to describe the general layout or structure of a system when it comes to software or technology. This includes a variety of elements, their interactions, and how they cooperate to accomplish functions or objectives.

4.1.1 Client-Side Architecture:

- Web-Based Interface: The system mainly relies on a web-based interface that can be accessed through typical web browsers on both desktop and mobile gadgets.
- Responsive Design: The user interface is crafted to be responsive, adjusting to different screen dimensions and resolutions to ensure a uniform experience regardless of the device being used.
- Frontend Technologies: The frontend components, such as web pages, forms, and interactive features, are built using HTML, CSS, JavaScript, and JSP (Java Server Pages).

4.1.2 Server-Side Architecture:

- Java Servlets: Servlets serve to manage HTTP requests and responses on the server side, executing business logic and communicating with the database.
- MVC Architecture: The server-side structure adheres to the Model-View-Controller (MVC) design pattern, segregating responsibilities into data manipulation (Model), presentation logic (View), and request handling (Controller).
- Service Layer: A service layer encapsulates the application's business logic, creating a separation between the presentation layer and the data access layer.

- Dependency Injection: Frameworks like Spring or CDI (Contexts and Dependency Injection) might be employed to handle dependencies and foster modularity.

4.1.3 Components:

- Presentation Layer: This layer comprises user interface elements visible to users, like web pages, forms, and interactive features for property listing, search, user registration, and profile management.
- Application Layer: This layer comprises server-side components managing user requests, executing business logic, and interfacing with the database. It encompasses servlets, controllers, and Java service classes handling various functionalities.
- Data Layer: Responsible for data storage and retrieval using an RDBMS (MySQL). It encompasses database tables, queries, and procedures for storing user profiles, property listings, and other pertinent data.

4.1.4 Security Considerations:

- Authentication and Authorization: User authentication methods like username/password is used to validate user identities and control access.
- Data Encryption: Sensitive data, including user credentials and personal information, undergoes encryption for both transmission and storage to prevent unauthorized access.
- Input Validation: Techniques for input validation is applied to prevent security risks like SQL injection.

4.2 Database Design:

An essential first step in creating a reliable and effective database system is database design. It entails building the database's architecture with data integrity, security, and performance in mind.

4.2.1 Introduction:

The database design for UniStay serves as the backbone of the entire application, providing a structured framework for storing, organizing, and managing data related to users, properties, messages, and transactions.

The primary objectives of the database design are to ensure data integrity, optimize data retrieval and manipulation operations, and support the functionality and performance requirements of the platform.

4.2.2 Entity-Relationship (ER) Model:

Following are the Entities and Attributes of the database:

- **User** (Email (PK), Username, Password, Mobilenumber, Reset_Token)
- **Owner** (Email (FK), Owner_Name, Owner_Contact)
- **Tenant** (Email (FK), Tenant_Name, Tenant_Contact)
- **Property** (PropertyID (PK), Email (FK), Property_Name, Address, RentalPrice)
- **Property Images** (Email (FK), Image (property images))
- **Profile Pictures** (Email (FK), Images (Profile Pictures)
- **Booking** (Booking_ID (PK), PropertyID (FK), Email (FK), Booking_Date, Status)

4.2.3 Relationships:

User - Owner (One-to-One):

- A user can be an owner. Each owner is associated with exactly one user.

User - Tenant (One-to-One):

- A user can be a tenant. Each tenant is associated with exactly one user.

Owner - Property (One-to-Many):

- An owner can list multiple properties for rent. Each property is associated with exactly one owner.

Tenant - Booking (One-to-Many):

- A tenant can make multiple bookings. Each booking is associated with exactly one tenant.

Property - Booking (One-to-Many):

- A property can have multiple bookings. Each booking is associated with exactly one property.

Table	Columns	Data Types	Constraints	Description
user	username	VARCHAR (100)		Stores the usernames of the users.
	mobilenumber	BIGINT		Stores Mobile number of users.
	email	VARCHAR (100)	PRIMARY KEY	Stores email of the user.
	password	VARCHAR (100)		Stores password.
	Reset_Token	VARCHAR (255)		Stores reset token to reset password and forgot password functionality.
image	id	INT AUTO_INCREMENT	PRIMARY KEY	Stores the id of images uploaded.
	image	LONGBLOB		Stores the image.
	email	VARCHAR (255)		Stores the email of the user uploading images.
admin	email	VARCHAR (255)	PRIMARY KEY	Stores email of the admin.
	password	VARCHAR (200)		Stores password of the admin.
user_roles_purposes	email	VARCHAR (255)	PRIMARY KEY	Stores email of the user.
	you_are	VARCHAR (255	NOT NULL	Stores the relation of user with the property.
	you_are_here_to	VARCHAR (255)	NOT NULL,	Stores the activity to be performed by the user.
	created at	TIMESTAMP DEFAULT CURRENT_TIMESTAMP	NOT NULL	Time when the listing was created.

Table 4.1 Based on database

4.2.4 ER Diagram:

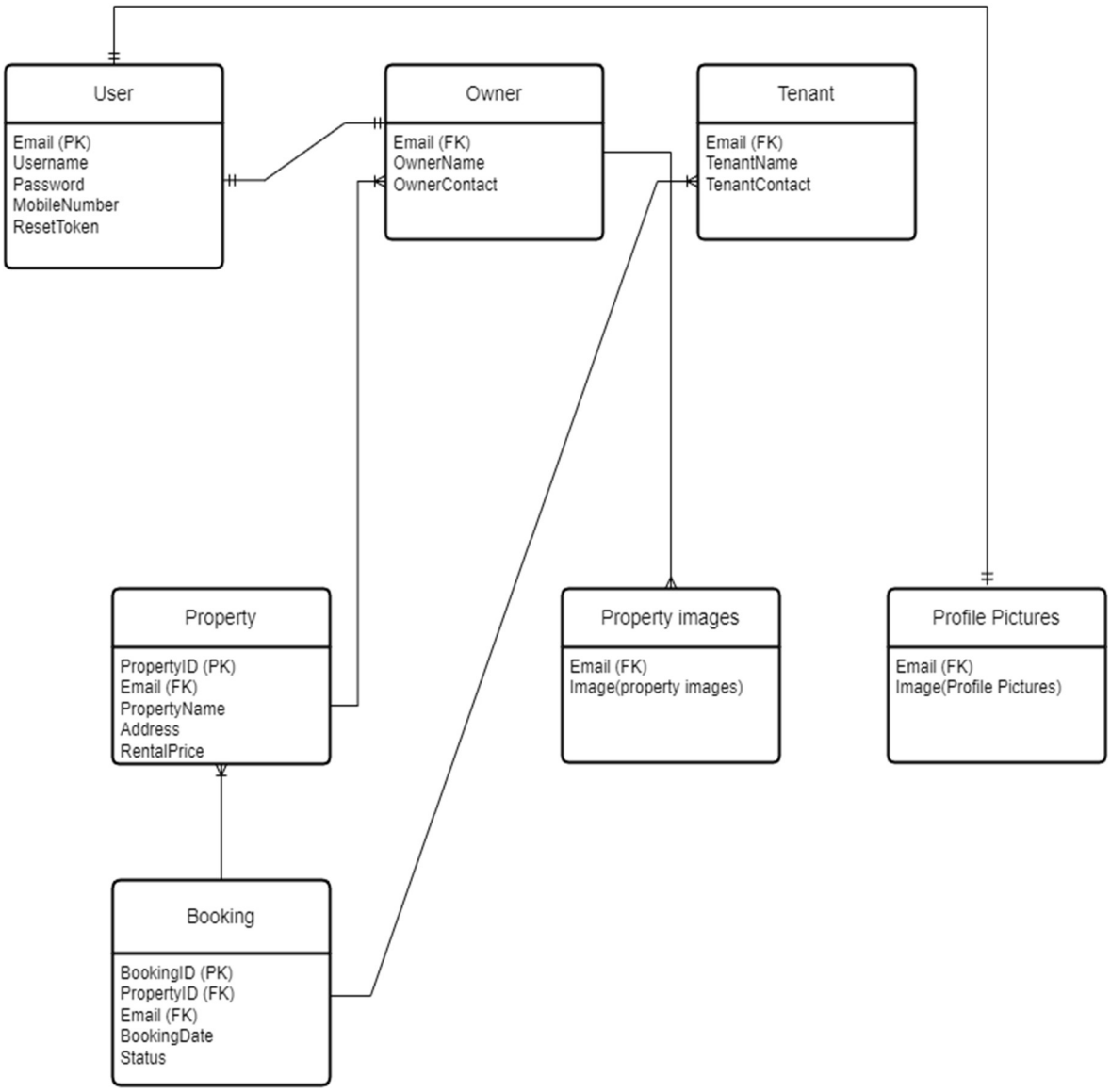

Figure 4.1 ER Diagram

4.2.5 Use case Diagram:

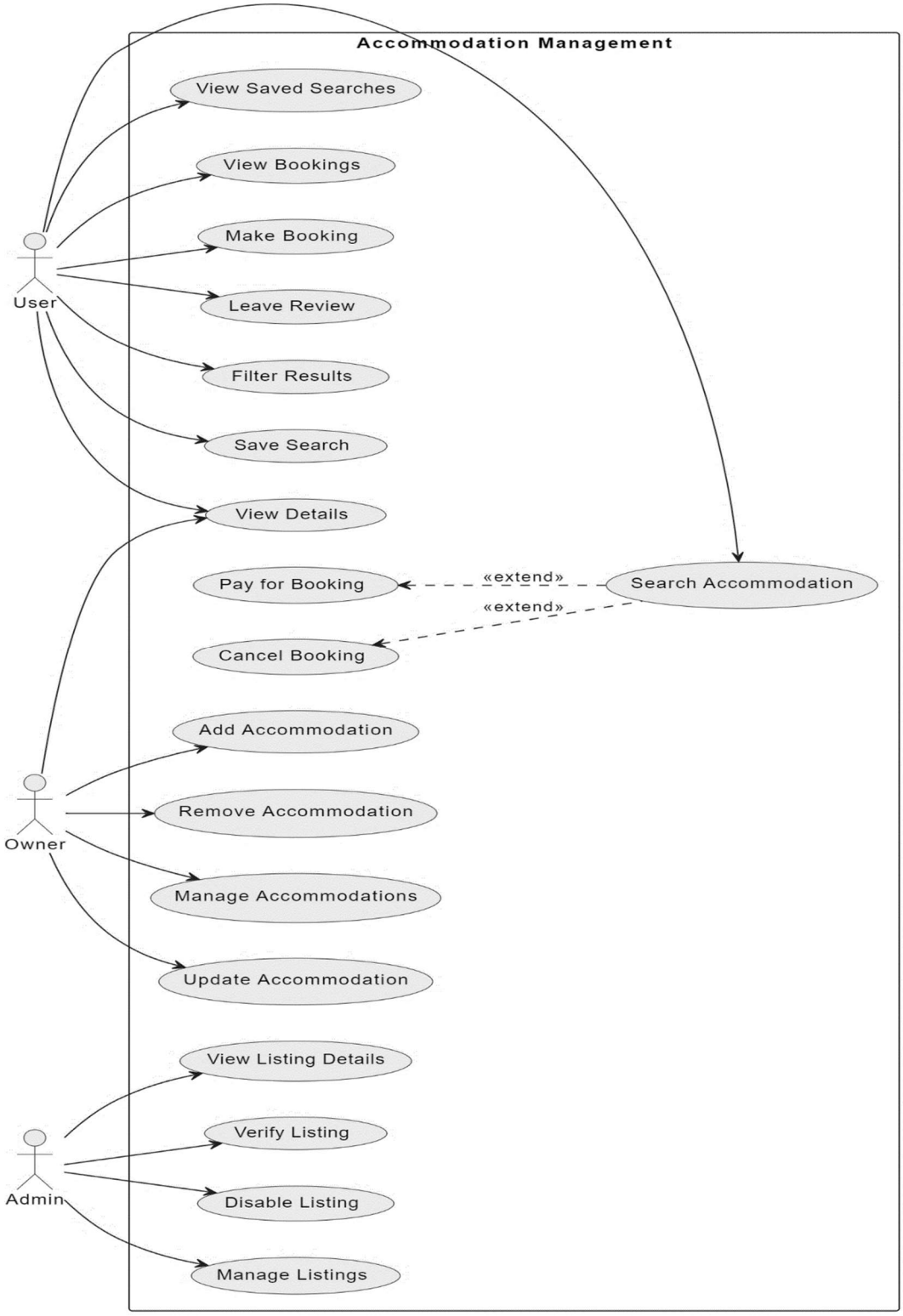

Figure 4.2 Use Case Diagram

CHAPTER V

SOFTWARE DEVELOPMENT METHODOLOGY

Keeping in mind the nature of the project and the requirements Agile software development methodology would be suitable. The reason of using it:

5.1 Iterative and Incremental Development:

- Agile methodologies advocate breaking down projects into smaller iterations or sprints, typically lasting 1-4 weeks. Each iteration focuses on delivering potentially shippable increments of the product, allowing for frequent releases and feedback from stakeholders.
- This approach enables continuous refinement and adaptation throughout the development process. By delivering incremental updates, the team can gather feedback from users and stakeholders, identify areas for improvement, and adjust priorities for subsequent iterations.
- For college projects, iterative and incremental development is beneficial as it allows students to make steady progress towards project goals while incorporating feedback and making adjustments as needed. It also helps manage project scope and complexity, ensuring that deliverables are achievable within the given timeframe.

5.2 Flexibility and Adaptability:

- Agile methodologies prioritize flexibility and adaptability to changing requirements, priorities, and market conditions. Instead of rigidly adhering to a fixed plan, agile teams embrace change and adjust their approach based on feedback and evolving needs.
- This flexibility is particularly advantageous for college projects, where requirements may evolve as students gain a deeper understanding of the problem domain or encounter unexpected challenges. Agile methodologies empower students to respond to changes effectively, iterate on their ideas, and deliver value incrementally.
- By embracing change and adapting to new requirements, students can create solutions that better align with the needs and expectations of their target audience, resulting in a more successful project outcome.

5.3 Collaboration and Communication:

- Agile methodologies emphasize collaboration and communication among team members, stakeholders, and end-users. Regular meetings, such as daily stand-ups, sprint planning sessions, and sprint reviews, facilitate transparent communication and ensure that everyone is aligned on project goals and priorities.
- In college projects, effective collaboration and communication are essential for coordinating efforts, sharing ideas, and resolving issues collaboratively. By fostering a collaborative environment, agile methodologies enable students to leverage each other's strengths, brainstorm creative solutions, and learn from one another.
- Through open and transparent communication channels, students can clarify requirements, address concerns, and ensure that everyone is on the same page regarding project progress and objectives.

5.4 Continuous Improvement:

- Agile promotes a culture of continuous improvement through retrospective meetings held at the end of each iteration. During these meetings, team members reflect on their performance, identify successes and areas for improvement, and discuss ways to enhance processes for future iterations.
- For college projects, embracing a mind-set of continuous improvement allows students to learn from their experiences, refine their approaches, and develop their skills over time. By reflecting on their progress and seeking feedback from peers and instructors, students can iteratively enhance their project outcomes and personal development.
- Continuous improvement is integral to the learning process, enabling students to identify strengths, weaknesses, and areas for growth. By actively seeking opportunities to improve, students can maximize the value they derive from their college projects and develop valuable skills that are applicable beyond the classroom.

CHAPTER VI

SYSTEM IMPLEMENTATION

6.1 Set up Development Environment:

- Install Eclipse IDE, Apache Tomcat Server, and MySQL as the database management system to create a comprehensive development environment.
- Configure Eclipse with necessary plugins for Java development and web application deployment.
- Install MySQL Workbench or another MySQL client to manage the database and execute SQL queries.

6.2 Frontend Development:

- Use HTML, CSS, JavaScript, and JSP (Java Server Pages) to develop responsive and visually appealing user interfaces.
- Design web pages with intuitive layouts, navigation menus, and forms for user registration, property listing, searching, and profile management.
- Implement client-side validation using JavaScript to ensure data integrity and provide real-time feedback to users.

6.3 Backend Development:

- Develop backend logic using Java programming language and servlets to handle HTTP requests and responses.
- Implement server-side functionalities for user authentication, authorization, and session management to ensure secure access to the application.
- Create service classes to encapsulate business logic and facilitate code reuse across multiple servlets.
- Use JDBC (Java Database Connectivity) frameworks to interact with the MySQL database, executing SQL queries and processing query results.

6.4 Database Design and Implementation:

- Design the database schema based on the project requirements, identifying entities, attributes, relationships, and constraints.
- Create tables for entities such as users, properties, and transactions, defining appropriate data types, primary keys, foreign keys, and constraints.

- Implement database operations for CRUD (Create, Read, Update, Delete) operations using JDBC, including prepared statements and transactions for data manipulation and retrieval.

CHAPTER VII

SYSTEM TESTING

7.1 Functional Testing:

Functional testing is a crucial aspect of software testing, focused on verifying that a software application functions correctly according to its specifications. It evaluates the system's behaviour against the expected behaviour, ensuring that it meets the requirements outlined by the stakeholders.

7.1.1 User Registration:

- Verify that the user registration process functions correctly, allowing users to sign up with valid information. Test for validation checks to ensure that users cannot register with invalid or incomplete data. Verify that user data is stored accurately in the database upon successful registration.

7.1.2 Property Listing and Searching:

- Test the property listing functionality to ensure that users can add, edit, and delete property listings as expected. Verify that search functionality works correctly, allowing users to find properties based on various criteria such as location, price range, and amenities.

7.1.3 User Profile Management:

- Verify that users can update their profiles, change settings, and manage their account information effectively. Test for functionalities such as updating personal details, changing passwords, and modifying notification preferences. Ensure that changes made by users are reflected accurately in the database.

7.2 Usability Testing:

Usability testing is a critical aspect of software testing focused on evaluating a product by testing it with representative users. Its primary goal is to ensure that the product is easy to use, intuitive, and provides a satisfactory user experience.

7.2.1 User Interface:

- Evaluate the user interface for its usability, including layout, design, and visual elements. Test for consistency in design across different pages and ensure that UI components are intuitive and easy to understand. Gather feedback from users to identify any usability issues or areas for improvement.

7.2.2 Navigation:

- Test the navigation flow of the platform to ensure that users can easily move between different sections and perform desired actions without confusion. Check for the availability of navigation menus, buttons, and links to guide users to relevant pages. Ensure that users can access essential features and functionalities without encountering obstacles.

7.3 Compatibility Testing:

Compatibility testing is a type of software testing that ensures a particular software application or system is compatible with various computing environments, software platforms, devices, browsers, networks, and operating systems. The goal is to verify that the software behaves as expected across different configurations and setups.

7.3.1 Browser Compatibility:

- Test the platform on different web browsers such as Chrome, Firefox, Safari, and Edge to ensure consistent performance and appearance across platforms. Check for compatibility issues related to rendering, layout, and functionality, and address any discrepancies found.

7.3.2 Device Compatibility:

- Evaluate the platform's compatibility with various devices, including desktops, laptops, tablets, and smartphones, running different operating systems such as Windows, macOS, iOS, and Android. Test for responsiveness and adaptability to different screen sizes and resolutions. Ensure that users can access and use the platform seamlessly across different devices and platforms.

CHAPTER VIII

OUTPUT FORMS & REPORTS

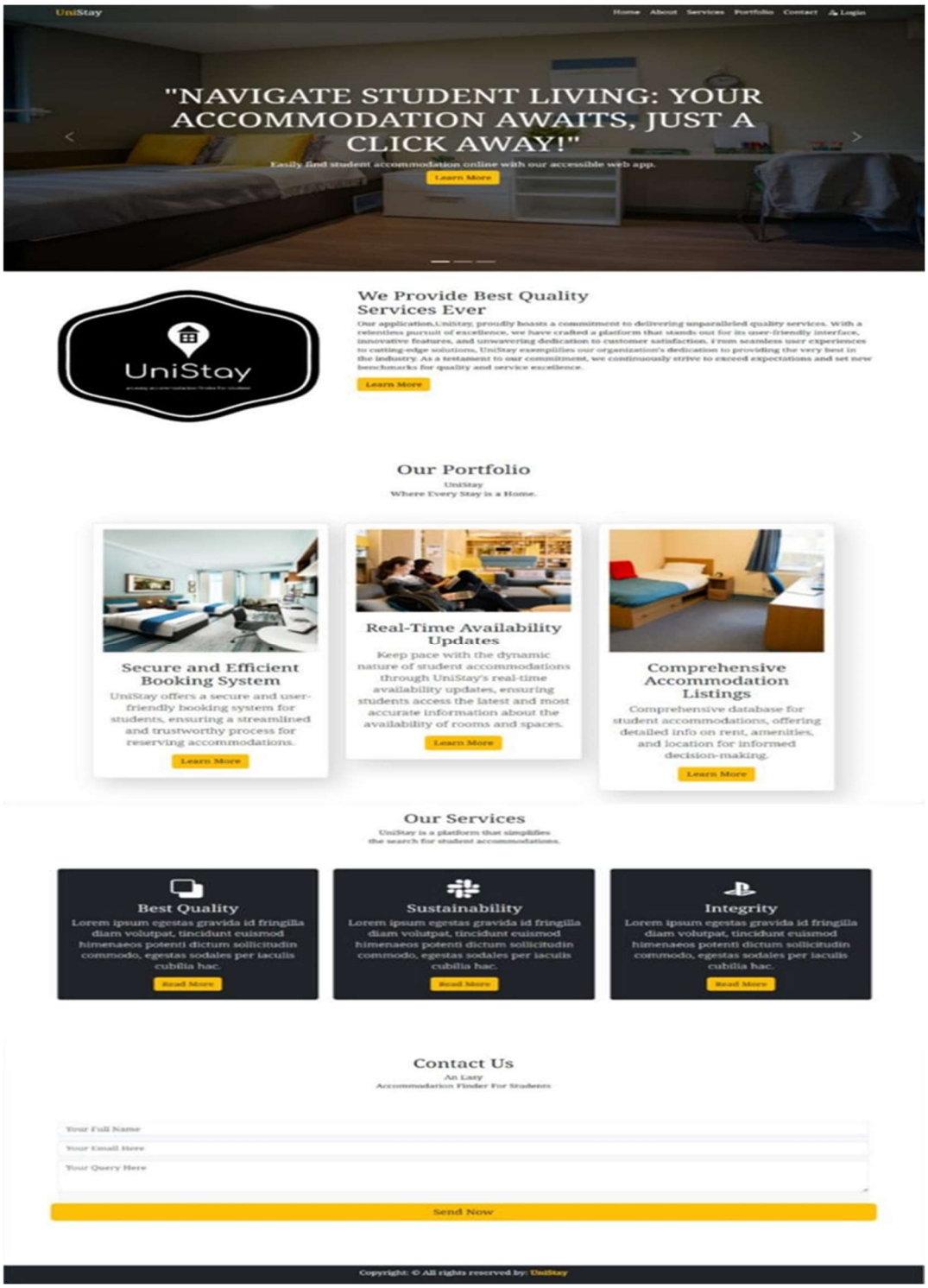

Figure 8.1 Home Page

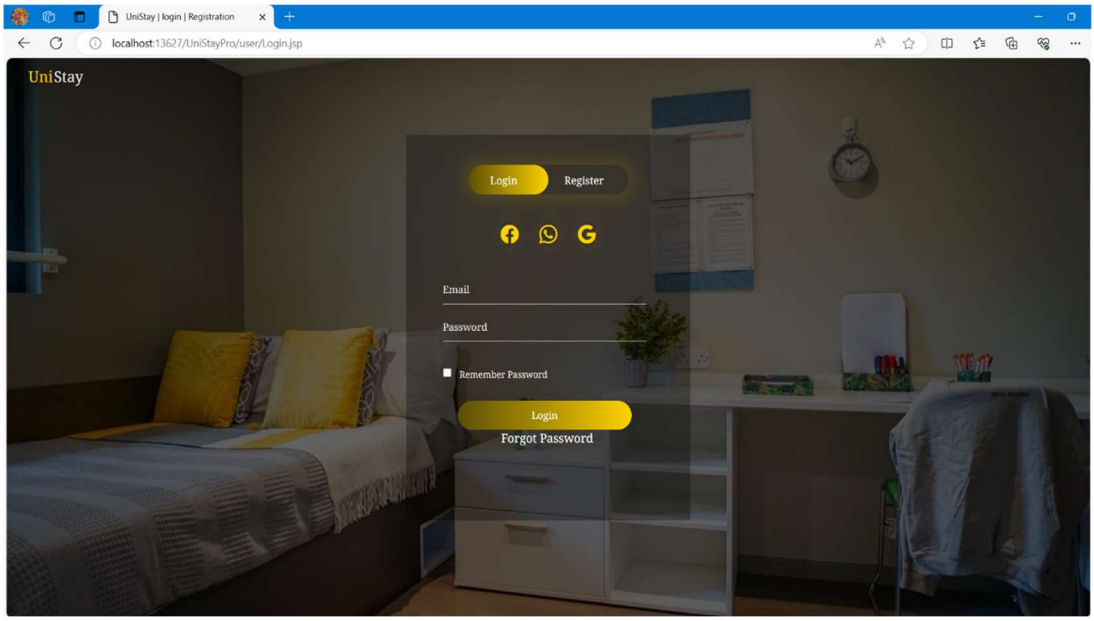

Figure 8.2 Login Page

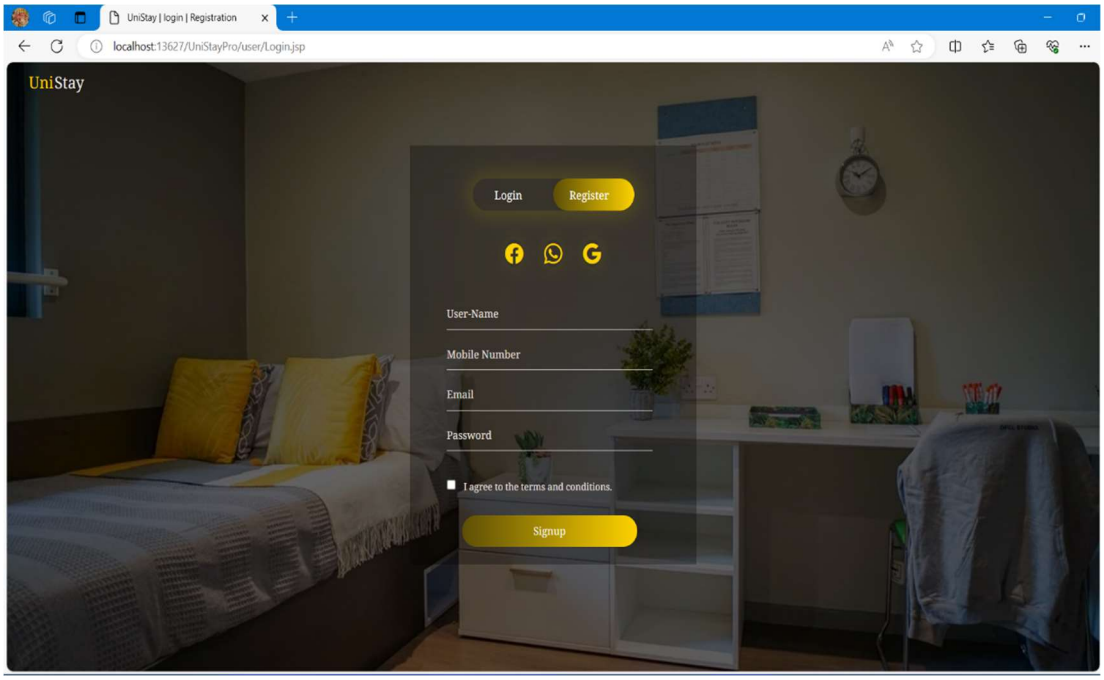

Figure 8.3 Signup Page

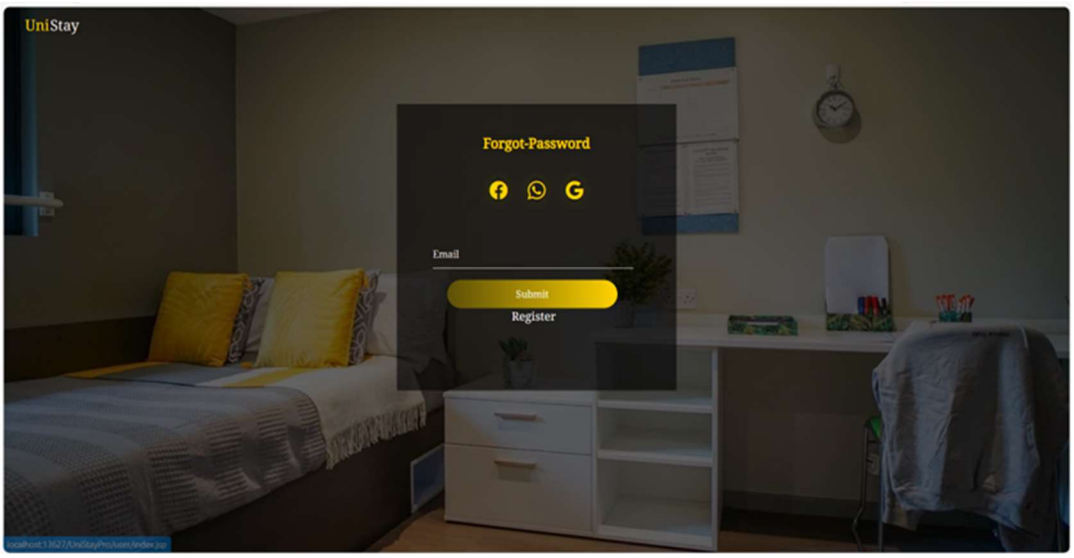

Figure 8.4 Forgot Password

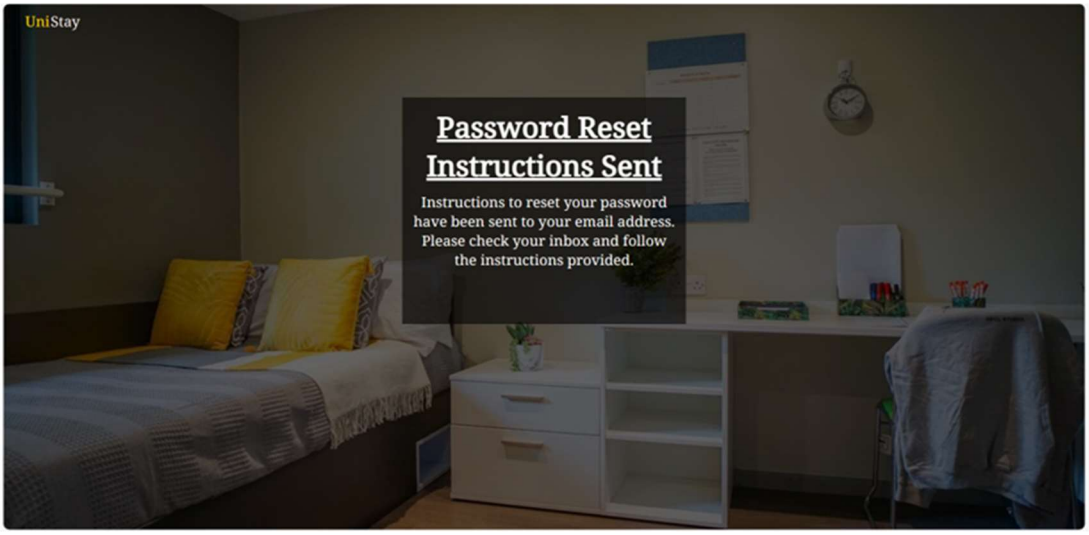

Figure 8.5 Password Reset Instructions sent

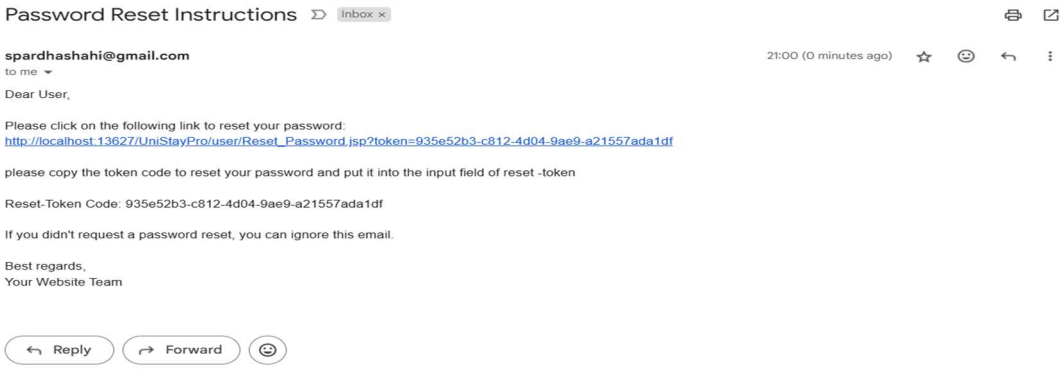

Figure 8.6 Password Reset Email with Reset Token

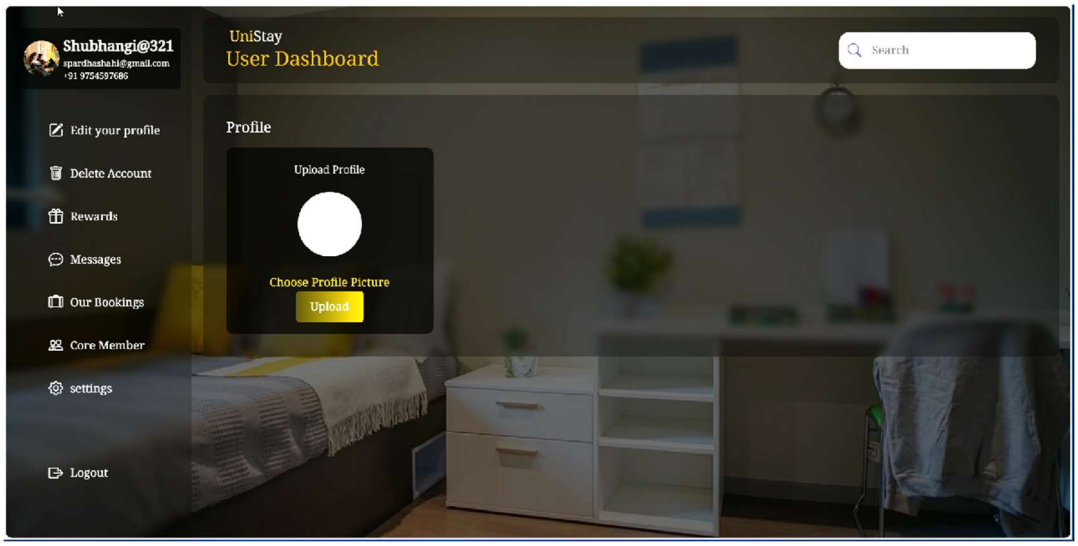

Figure 8.7 User Dashboard

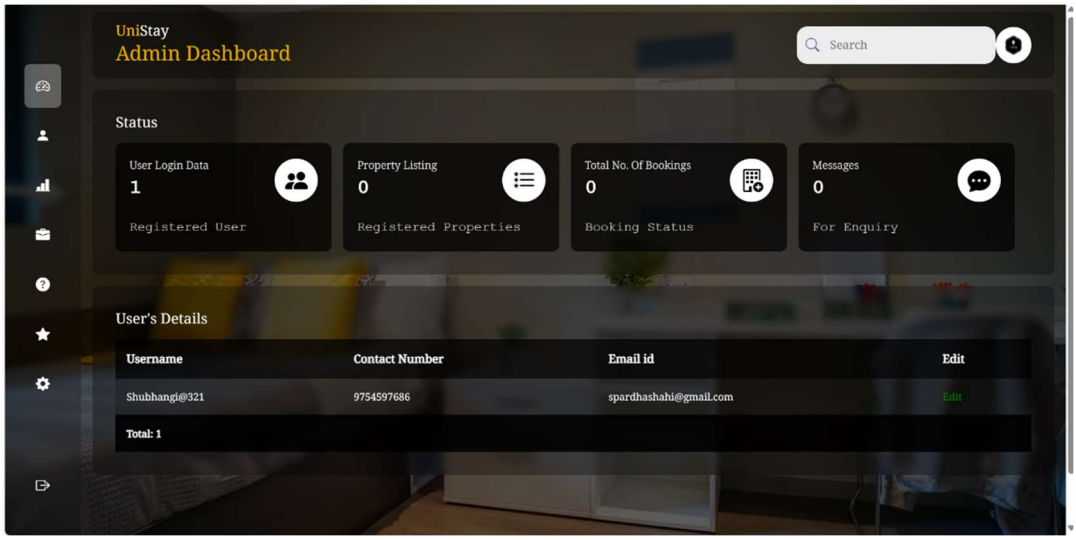

Figure 8.8 Admin Dashboard

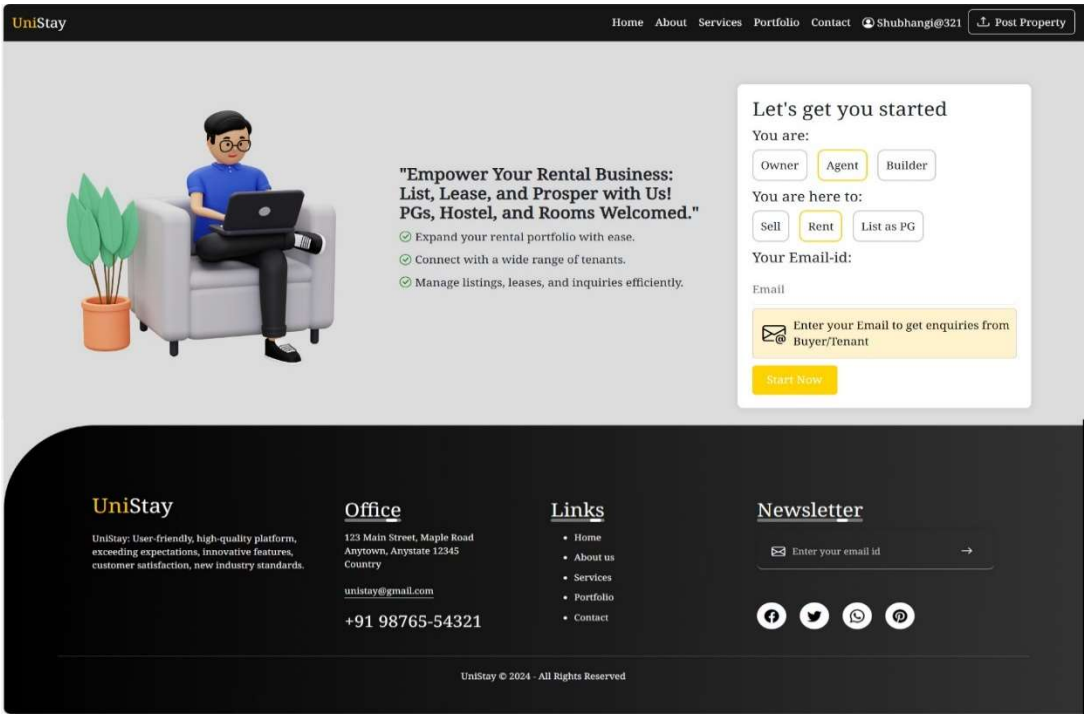

Figure 8.9 Post Property Form Part I

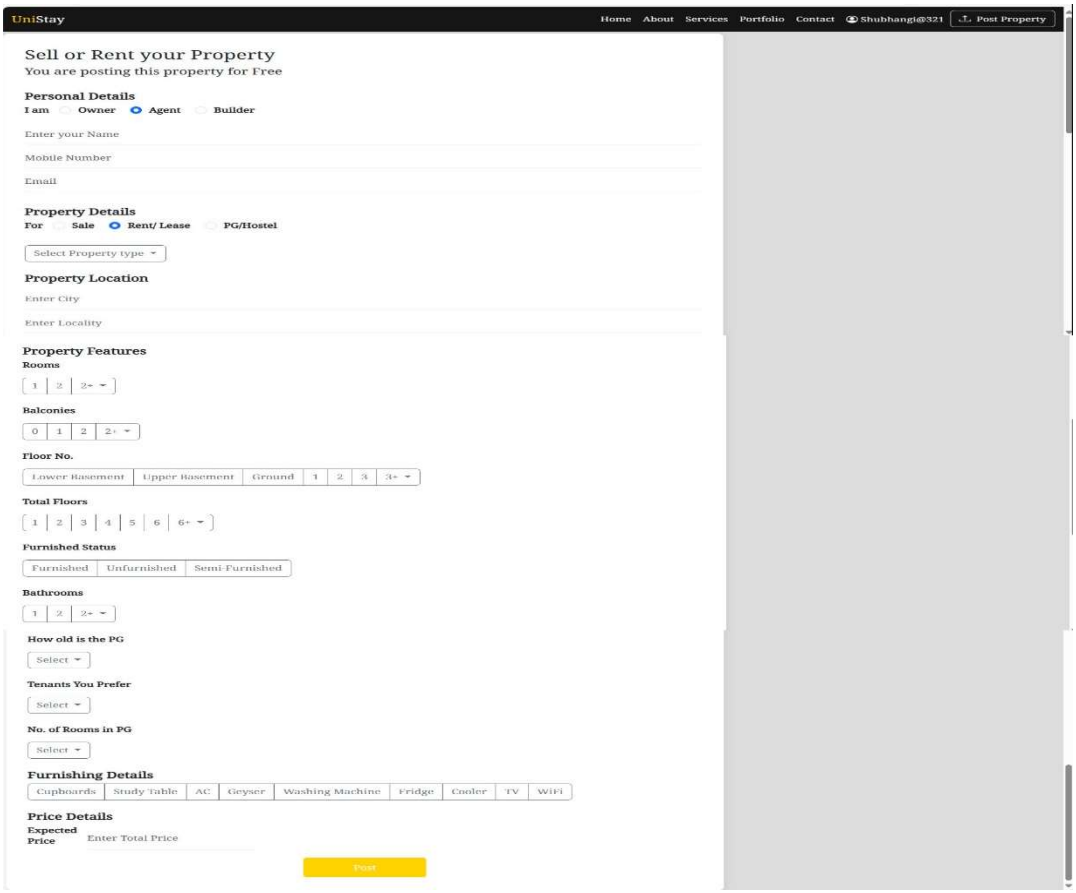

Figure 8.10 Post Property Form Part II

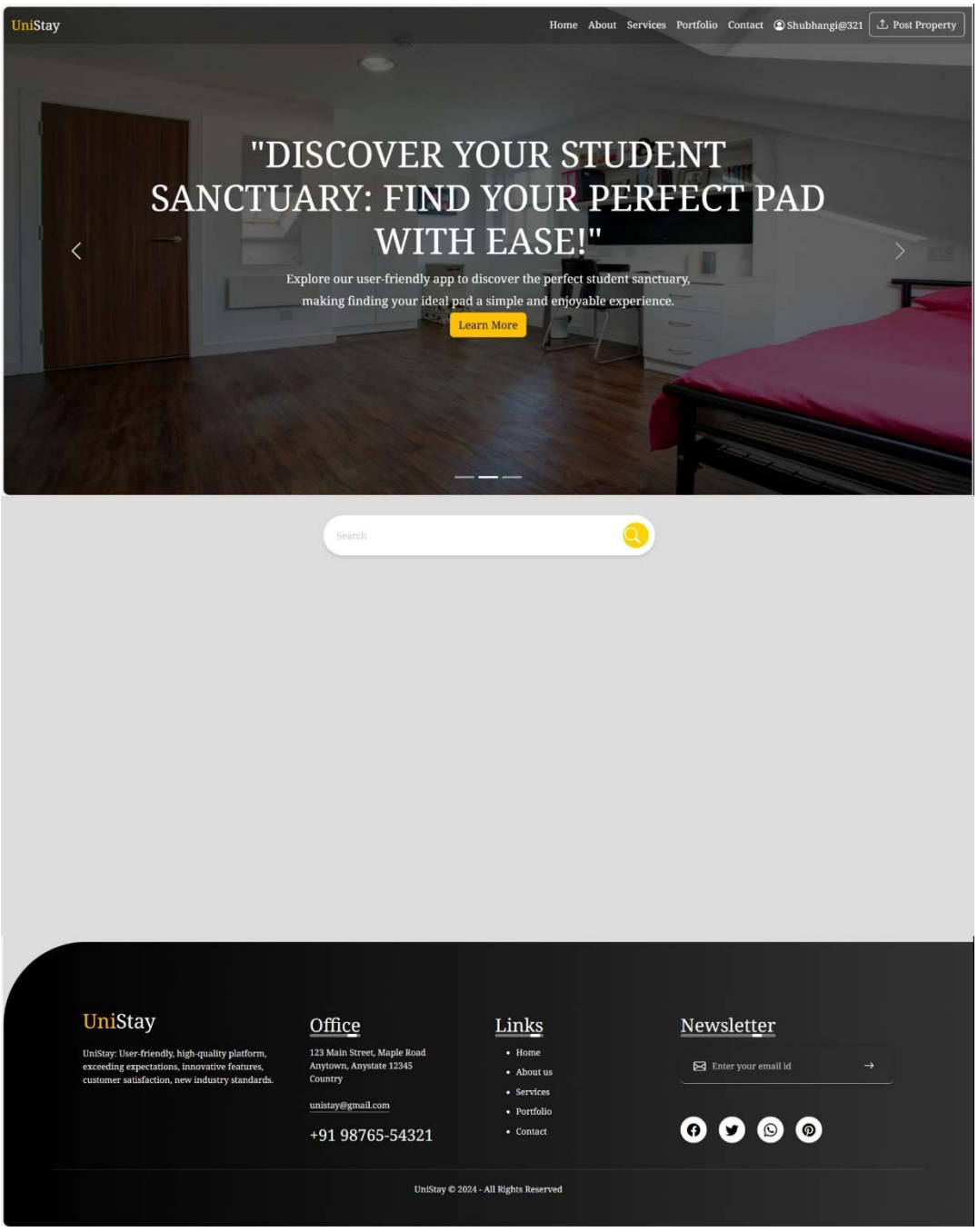

Figure 8.11 Page for Accommodation Listing and User Search all the Accommodation here

CHAPTER IX

LIMITATIONS

9.1Limited Market Reach: The success of this platform mostly depends on drawing in a sizable user base, which includes both landlords and tenants. But at first, it can have trouble taking off and have a small user base.

9.2Geographical Constraints: The platform will initially just cover a small area at first, concentrating only on Indore location. This restriction would lessen its allure for people searching for rentals in places the platform does not service.

9.3Trust and Credibility: It might be difficult to gain users' confidence and credibility, especially in the beginning. Concerns about security, dependability, and validity may make users reluctant to deal on a relatively new site.

9.4User Experience: Ensuring a smooth and easily navigable user experience is essential for retaining and satisfying users. It might be necessary to continuously improve and iterate in order to create and optimise the platform's functionality and user experience in order to fulfil user expectations.

9.5Limited Industry Experience: Student project teams may lack practical experience working in the industry or understanding industry best practices. This could impact the project's ability to meet industry standards and requirements.

CHAPTER X

CONCLUSION

The UniStay project is dedicated to revolutionizing the student accommodation experience by offering a centralized, user-friendly platform. Our primary goal is to simplify the process of finding suitable accommodations for students while ensuring transparency, security, and compliance with regulations. Leveraging advanced search functionalities and secure booking systems, UniStay aims to enhance the overall experience for both students and accommodation providers alike.

Our platform caters to a diverse range of accommodation types, including rooms, paying guest accommodations, and hostels, providing students with a comprehensive selection to meet their specific needs and preferences. With UniStay, students can explore various options conveniently and confidently, knowing that each listing is thoroughly vetted for quality and reliability.

Furthermore, our admin features play a crucial role in maintaining the integrity and trustworthiness of the platform. Admin controls ensure transparency in listings, safeguarding users against fraudulent or misleading information. Additionally, stringent security measures are implemented to protect user data and transactions, fostering a safe and secure environment for all users.

At UniStay, we prioritize user experience, striving to deliver a seamless and intuitive platform that meets the evolving needs of our student community. Our commitment to scalability ensures that as our user base grows, the platform remains robust and responsive, capable of accommodating increasing demand and expanding services.

In summary, UniStay is more than just a housing platform—it is a reliable companion for students embarking on their academic journey, offering convenience, security, and peace of mind every step of the way.

References

Books:

- Kathy Sierra, Bert Bates (2003), Head First Java, O'Reilly Media.
- Herbert Schildt (1997), Java the Complete Reference Eleventh Edition
- Munro, J. (2013), Bootstrap 4 in Action, Manning Publications.
- Haverbeke, M. (2018), Eloquent JavaScript Third Edition, No Starch Press.
- Crockford, D. (2008). JavaScript: The Good Parts. O'Reilly Media.

Java:

- Oracle Java SE Documentation: https://docs.oracle.com/en/java/
- The Java Tutorials: https://docs.oracle.com/javase/tutorial/
- Java API Specification: https://docs.oracle.com/en/java/javase/19/docs/api/

Bootstrap:

- Bootstrap Documentation: https://getbootstrap.com/
- Get Bootstrap Blog: https://blog.getbootstrap.com/

JavaScript:

- Mozilla Developer Network (JavaScript): https://developer.mozilla.org/en-US/docs/Web/JavaScript
- The Modern JavaScript Tutorial: https://javascript.info/

Appendix I

Acronyms and Abbreviations:

1.	**PGs**	Paying Guests (Accommodations)
2.	**Admin**	Administrator
3.	**DBMS**	Database Management System
4.	**UI**	User Interface
5.	**UX**	User Experience
6.	**API**	Application Programming Interface
7.	**GPS**	Global Positioning System
8.	**SSL**	Secure Sockets Layer
9.	**CMS**	Content Management System
10.	**FAQ**	Frequently Asked Questions
11.	**CSV**	Comma-Separated Values
12.	**HTML**	Hypertext Markup Language
13.	**CSS**	Cascading Style Sheets
14.	**JS**	JavaScript
15.	**JSON**	JavaScript Object Notation
16.	**HTTP**	Hypertext Transfer Protocol
17.	**HTTPS**	Hypertext Transfer Protocol Secure
18.	**CRM**	Customer Relationship Management
19.	**ROI**	Return on Investment
20.	**API**	Application Programming Interface
21.	**SDK**	Software Development Kit
22.	**SQL**	Structured Query Language
23.	**SRS**	Software Requirements Specification
24.	**MVP**	Minimum Viable Product
25.	**QA**	Quality Assurance
26.	**UX/UI**	User Experience/User Interface
27.	**KPI**	Key Performance Indicator
28.	**GDPR**	General Data Protection Regulation
29.	**CSS**	Customer Support System
30.	**SMTP**	Simple Mail Transfer Protocol
31.	**LDAP**	Lightweight Directory Access Protocol
32.	**AWS**	Amazon Web Services
33.	**IoT**	Internet of Things
34.	**API**	Application Programming Interface
35.	**ERP**	Enterprise Resource Planning
36.	**VPN**	Virtual Private Network
37.	**CORS**	Cross-Origin Resource Sharing
38.	**DNS**	Domain Name System

Table 1. Acronyms and Abbreviations

Appendix II

➢ Home page Code

This code snippet demonstrates a Home Page of UniStay

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta  name="viewport"  content="width=device-width,  initial-
scale=1">
    <title>UniStay</title>
    <link                                          rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.11.3/font/bootstrap-icons.min.css">
      <link rel="stylesheet"  href="style.css" >
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootst
rap.min.css"          rel="stylesheet"          integrity="sha384-
T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN"
crossorigin="anonymous">
  <style>@import
url('https://fonts.googleapis.com/css2?family=Noto+Serif:wght@300;
400&display=swap');
   *{
        margin: 0;
        padding: 0;
    }
body{
  background:f1fbff;
  }
  .section-padding{
  padding:100px 0;
  }
  .carousel-item{
  height: 100vh;
  min-height: 300px;
  }
  .carousel-caption{
  bottom: 220px;
  z-index: 2;
  }
  .carousel-caption h5{
  font-size: 55px;
  text-transform:uppercase;
  margin-top: 25px;
  font-weight: 500;
  }
  .carousel-caption p{
  width: 60%;
  font-size: 18px;
  margin: auto;
  line-height: 1.9;

  }
  .carousel-inner::before{
```

```
  content: '';
  position: absolute;
  width: 100%;
  height: 100%;
  top: 0;
  left: 0;
  background: rgba(0,0,0,0.6);
  z-index: 1;
  }
  .w-100{
height: 100vh;
}
@media only screen and (min-width: 768px) and (max-width: 991px){
.carousel-caption{
bottom: 370px;
}
.carousel-caption p{
width: 100%;
}
}
@media only screen and (max-width: 767px){
.carousel-caption{
 bottom: 125px;
 }
 .carousel-caption h5{
 padding: 10px 15px;
 font-size: 17px;
 }
 .carousel-caption a{
 padding: 10px 15px;
 }
 .carousel-caption p{
 width: 100%;
 line-height: 1.6;
 font-size: 12px;
 }
}

section .button {
    margin: 16px 0;
 }

 section .button {
    margin: 16px 0;
 }

 section .button button {
    outline: none;
    padding: 8px 16px;
    border-radius: 4px;
    font-size: 25px;
    font-weight: 400;
    border: 2px solid transparent;
    cursor: pointer;
    transition: all 0.4s ease;
 }

 section .button button:hover {
    border-color: #ffc107;
    background-color: #fff;
 }
```

```
 .services .card-body i{
 font-size: 50px;
 }

 .portfolio .card{
 box-shadow: 15px 15px 40px rgba(0,0,0,0.15);
 }

 @media only screen and (max-width: 1280px) and (min-height: 551px){
    .carousel-caption{
    top: 50px;
}
}
  </style>
  </head>
  <body>
    <!-- Navbar/Banner -->
   <%@include file="navbar.jsp" %>
   <div id="carouselExampleCaptions" class="carousel slide">
  <div class="carousel-indicators">
    <button type="button" data-bs-target="#carouselExampleCaptions"
data-bs-slide-to="0"   class="active"   aria-current="true"   aria-
label="Slide 1"></button>
    <button type="button" data-bs-target="#carouselExampleCaptions"
data-bs-slide-to="1" aria-label="Slide 2"></button>
    <button type="button" data-bs-target="#carouselExampleCaptions"
data-bs-slide-to="2" aria-label="Slide 3"></button>
  </div>
  <div class="carousel-inner">
    <div class="carousel-item active">
      <img src="image/New2.webp" class="d-block w-100" alt="...">
      <div class="carousel-caption">
        <h5>"Navigate Student Living: Your Accommodation Awaits,
Just a Click Away!"</h5>
        <p>Easily find student accommodation online with our
accessible web app.</p>
          <button   type="button"   class="btn   btn-warning">Learn
More</button>
      </div>
    </div>
    <div class="carousel-item">
      <img src="image/second.jpg" class="d-block w-100" alt="...">
      <div class="carousel-caption">
        <h5>"Discover Your Student Sanctuary: Find Your Perfect Pad
with Ease!"</h5>
        <p>Explore our user-friendly app to discover the perfect
student sanctuary,
        making finding your ideal pad a simple and enjoyable
experience.</p>
          <button   type="button"   class="btn   btn-warning">Learn
More</button>
      </div>
    </div>
    <div class="carousel-item">
      <img src="image/first.jpg" class="d-block w-100" alt="...">
      <div class="carousel-caption">
        <h5>"Unlock Your Student Experience: Explore, Compare, and
Secure Your Ideal Digs Today!"</h5>
        <p>Our app empowers students to explore, compare, and secure
their ideal accommodation swiftly,
```

```
          enhancing their overall student experience.</p>
          <button    type="button"    class="btn    btn-warning">Learn
More</button>
        </div>
      </div>
    </div>
    <button   class="carousel-control-prev"   type="button"   data-bs-
target="#carouselExampleCaptions" data-bs-slide="prev">
      <span          class="carousel-control-prev-icon"          aria-
hidden="true"></span>
      <span class="visually-hidden">Previous</span>
    </button>
    <button   class="carousel-control-next"   type="button"   data-bs-
target="#carouselExampleCaptions" data-bs-slide="next">
      <span          class="carousel-control-next-icon"          aria-
hidden="true"></span>
      <span class="visually-hidden">Next</span>
    </button>
</div>

  <!-- about section -->

<section id="about" class="about-section-padding">
<div class="container">
<div class="row">
<div class="col-lg-4 col-md-12 col-12">
<div class="about-img">
<img       src="image/unistay-high-resolution-logo-transparent.png"
alt="" class="img-fluid">
</div>
</div>
<div class="col-lg-8 col-md-12 col-12 ps-lg-5 mt-md-5">
<div class="about-text">
<h2>We Provide Best Quality <br>Services Ever</h2>
<p class="text-justify">
Our application,UniStay, proudly boasts a commitment to delivering
unparalleled quality services.
With a relentless pursuit of excellence, we have crafted a platform
that stands out for its
user-friendly   interface,   innovative   features,   and   unwavering
dedication to customer satisfaction.
From seamless user experiences to cutting-edge solutions, UniStay
exemplifies our organization's
dedication to providing the very best in the industry. As a testament
to our commitment,
we continuously strive to exceed expectations and set new benchmarks
for quality and service excellence.
</p>
 <button type="button" class="btn btn-warning">Learn More</button>
</div>
</div>
</div>
</div>
</section>

  <!-- Services Section -->
<section id="services" class="services section-padding">
<div class="container">
<div class="row">
<div class="col-md-12">
<div class="section-header text-center pb-5">
```

```
<h2>Our Services</h2>
<p>UniStay is a platform that simplifies<br> the search for student
accommodations.
</p>
</div>
</div>
</div>

<div class="row">
<div class="col-12 col-md-12 col-lg-4">
<div class="card text-white text-center bg-dark pb-2">
<div class="card-body">
<i class="bi bi-subtract"></i>
<h3 class="card-title">Best Quality</h3>
<p  class="lead">Lorem  ipsum  egestas  gravida  id  fringilla  diam
volutpat,
 tincidunt  euismod  himenaeos  potenti  dictum  sollicitudin  commodo,
egestas  sodales  per
  iaculis  cubilia  hac.</p>
 <button class="btn btn-warning text-dark">Read More</button>
</div>
</div>
</div>

<div class="col-12 col-md-12 col-lg-4">
<div class="card text-white text-center bg-dark pb-2">
<div class="card-body">
<i class="bi bi-slack"></i>
<h3 class="card-title">Sustainability</h3>
<p  class="lead">Lorem  ipsum  egestas  gravida  id  fringilla  diam
volutpat,
 tincidunt  euismod  himenaeos  potenti  dictum  sollicitudin  commodo,
egestas  sodales  per
  iaculis  cubilia  hac.</p>
 <button class="btn btn-warning text-dark">Read More</button>
</div>
</div>
</div>

<div class="col-12 col-md-12 col-lg-4">
<div class="card text-white text-center bg-dark pb-2">
<div class="card-body">
<i class="bi bi-playstation"></i>
<h3 class="card-title">Integrity</h3>
<p  class="lead">Lorem  ipsum  egestas  gravida  id  fringilla  diam
volutpat,
 tincidunt  euismod  himenaeos  potenti  dictum  sollicitudin  commodo,
egestas  sodales  per
  iaculis  cubilia  hac.</p>
 <button class="btn btn-warning text-dark">Read More</button>
</div>
</div>
</div>
</div>
</div>
</section>

<!-- Portfolio Section -->
<section id="portfolio" class="portfolio section-padding">
<div class="container">
<div class="row">
```

```
<div class="col-md-12">
<div class="section-header text-center pb-5">
<h2>Our Portfolio</h2>
<p>UniStay <br> Where Every Stay is a Home.</p>
</div>
</div>
</div>
<div class="row">
<div class="col-12 col-md-12 col-lg-4">
<div class="card text-center bg-white pb-2">
<div class="card-body text-dark">
<div class="img-area mb-4">
<img src="image/port1.jpg" alt="" class="img-fluid">
</div>
<h3 class="card-title">Secure and Efficient Booking System</h3>
<p class="lead">UniStay offers a secure and user-friendly booking
system
for students, ensuring a streamlined and trustworthy process for
reserving accommodations.</p>
<button class="btn btn-warning text-dark">Learn More</button>
</div>
</div>
</div>

<div class="col-12 col-md-12 col-lg-4">
<div class="card text-center bg-white pb-2">
<div class="card-body text-dark">
<div class="img-area mb-4">
<img src="image/port2.jpg" alt="" class="img-fluid">
</div>
<h3 class="card-title">Real-Time Availability Updates</h3>
<p class="lead">Keep pace with the dynamic nature of student
accommodations through UniStay's real-time availability updates,
 ensuring students access the latest and most accurate information
about the availability of rooms and spaces.</p>
<button class="btn btn-warning text-dark">Learn More</button>
</div>
</div>
</div>

<div class="col-12 col-md-12 col-lg-4">
<div class="card text-center bg-white pb-2">
<div class="card-body text-dark">
<div class="img-area mb-4">
<img src="image/port3.jpg" alt="" class="img-fluid">
</div>
<h3 class="card-title">Comprehensive Accommodation Listings</h3>
<p class="lead">Comprehensive database for student accommodations,
offering detailed info on rent,
amenities, and location for informed decision-making.</p>
<button class="btn btn-warning text-dark">Learn More</button>
</div>
</div>
</div>

</div>
</div>
</section>

<!-- Contact Section -->
<section id="contact" class="contact section-padding">
```

```
<div class="container">
<div class="row">
<div class="col-md-12">
<div class="section-header text-center pb-5">
<h2>Contact Us</h2>
<p>An Easy <br> Accommodation Finder For Students </p>
</div>
</div>
</div>

<div class="row m-0">
<div class="col-md-12 p-0 pt-4 pb-4">
<form action="#" class="bg-light p-4.m-auto"><div class="row">
<div class="col-md-12">
<div class="mb-3">
<input type="text" class="form-control" required placeholder="Your
Full Name">
</div>
</div>
<div class="col-md-12">
<div class="mb-3">
<input type="email" class="form-control" required placeholder="Your
Email Here">
</div>
</div>
<div class="col-md-12">
<div class="mb-3">
<textarea rows="3" required class="form-control" placeholder="Your
Query Here"></textarea>
</div>
</div>
<button class="btn btn-warning btn-lg btn-block mt-3 text-dark">Send
Now</button>
</div>
</form>
</div>
</div>
</div>
</section>


<!-- footer -->
<footer class="bg-dark p-2 text-center">
<div class="container">
<p class="text-white">
 Copyright: &copy; All rights reserved by:
 <a href="#" style="text-decoration: none">
 <strong class="text-warning">UniStay</strong>
  </a>
 </p>
</div>
</footer>

    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstra
p.bundle.min.js"                                    integrity="sha384-
C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL"
crossorigin="anonymous"></script>
  </body>
</html>
```

➢ Java Email Sender Code:

This code snippet demonstrates a Java Email Sender Code

```
package project;

import javax.mail.*;
import javax.mail.internet.*;
import java.util.Properties;

public class EmailSender {
    public static void sendResetPasswordEmail(String recipientEmail, String resetToken) {
        final String senderEmail = "spardhashahi@gmail.com";
        final String username = "spardhashahi@gmail.com";
        final String password = "iico islw wqbe bwtd";

        // Setup mail server properties
        Properties props = new Properties();
        props.put("mail.smtp.auth", "true");
        props.put("mail.smtp.starttls.enable", "true");
        props.put("mail.smtp.host", "smtp.gmail.com");
        props.put("mail.smtp.port", "465"); // Use port 465 for SSL/TLS
        props.put("mail.smtp.ssl.enable", "true");

        // Create session with authenticator
        Session session = Session.getInstance(props, new Authenticator() {
            @Override
            protected PasswordAuthentication getPasswordAuthentication() {
                return new PasswordAuthentication(username, password);
            }
        });

        try {
            // Create a default MimeMessage object
            Message message = new MimeMessage(session);

            // Set From: header field
            message.setFrom(new InternetAddress(senderEmail));

            // Set To: header field
            message.setRecipients(Message.RecipientType.TO, InternetAddress.parse(recipientEmail));

            // Set Subject: header field
            message.setSubject("Password Reset Instructions");

            // Set email body
            String emailBody = "Dear User,\n\n";
            emailBody += "Please click on the following link to reset your password:\n";
            emailBody += "http://localhost:13627/UniStayPro/user/Reset_Password.jsp?token=" +
resetToken + "\n\n";
            emailBody += "please copy the token code to reset your password and put it into the input
field of reset -token \n\n";
            emailBody += "Reset-Token Code: " + resetToken + "\n\n";
            emailBody += "If you didn't request a password reset, you can ignore this email.\n\n";
            emailBody += "Best regards,\nYour Website Team";
            message.setText(emailBody);

            // Send message
```

```
            Transport.send(message);

            System.out.println("Password reset email sent successfully to " + recipientEmail);
        } catch (MessagingException e) {
            System.err.println("Error sending password reset email to " + recipientEmail);
            e.printStackTrace();
        }
    }
}
```

➢ User Dashboard

This code snippet demonstrates a User Dashboard

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<%@ page import="java.sql.*, java.awt.image.BufferedImage,
javax.imageio.ImageIO, java.io.ByteArrayInputStream,
java.util.Base64" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>MyProfile</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootst
rap.min.css" rel="stylesheet" integrity="sha384-
QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH"
crossorigin="anonymous">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.11.3/font/bootstrap-icons.min.css">
    <link rel="stylesheet" href="style.css">
    <style>
        @import
url('https://fonts.googleapis.com/css2?family=Noto+Serif:wght@300;
400&display=swap');

        * {
            margin: 0;
            padding: 0;
            border: none;
            outline: none;
            box-sizing: border-box;
        }

        body{
            width: 100%;
            height: 100vh;
            background-image: linear-gradient(rgba(0, 0, 0, 0.6),
rgba(0, 0, 0, 0.6)), url(image/New2.webp);
            background-position: center;
            background-size: cover;
            display: flex;
        }
        .side-nav {
            width: 110px;
            height: 100%;
            position: fixed;
            top: 0;
```

```
            left: 0;
            padding: 30px 15px;
            background: rgba(0, 0, 0, 0.2);
            backdrop-filter: blur(5px);
            color: white;
            display: flex;
            justify-content: space-between;
            flex-direction: column;
            z-index: 1; /* Ensure the side nav is above the main
content */
        }

        .nav{
            font-size: 20px;
            align-items: center;
            margin-left: 5px;
        }

        .user {
            display: flex;
            align-items: center;
            justify-content: space-between;
            width: 60px;
            font-size: 12px;
            padding: 10px;
            border-radius: 8px;
            margin-left: auto;
            margin-right: auto;
            overflow: hidden;
        }

        .user .user-name {
            display: none;
        }

        .user-name {
            margin-left: 7px;
        }

        .user-name h5 {
            font-size: 20px;
            font-weight: 600;
            white-space: nowrap;
        }

        .user-img {
            width: 40px;
            border-radius: 50%;
            margin: auto;
        }

        #imageContainer1 {
            width: 50px;
            height: 50px;
            border-radius: 50%;
            background-size: cover;
        }

        ul {
            list-style: none;
            padding: 0 15px;
```

```
}

ul li {
    margin: 30px;
    display: flex;
    align-items: center;
    justify-content: center;
    cursor: pointer;
    border-radius: 5px;
    transition: all 0.5s ease-in-out;
}

ul li i {
    font-size: 20px;
    margin-right: 0px;
}

ul li:hover{
    text-decoration:underline;
}

ul li a {
    white-space: nowrap;
    text-decoration: none;
    color: white;
    display: none;
}

.side-nav:hover {
    width: 260px;
}

.side-nav:hover .user .user-name {
    display: block;
}

.side-nav:hover .user {
    width: 100%;
    background: rgba(0,0,0,0.5);
    backdrop-filter: blur(5px);
}

.side-nav:hover .user-img{
    margin: 0;
}

.side-nav:hover ul li a{
    display: block;
}

.side-nav:hover ul li i{
    margin-right: 10px;
}

.side-nav:hover ul li{
    justify-content: flex-start;
}

.main--content{
    margin-left: 110px; /* Adjusted margin */
    color: white;
```

```
            position: relative;
            width: calc(100% - 110px); /* Calculate width based on
side nav */
            padding: 1rem;
            transition: margin-left 0.5s; /* Added transition for
smooth shifting */
            z-index: 0; /* Ensure the main content is behind the
side nav */
        }

        .header--wrapper{
            cursor: pointer;
            display: flex;
            justify-content: space-between;
            align-items: center;
            flex-wrap: wrap;
            background: rgba(0,0,0,0.3);
            backdrop-filter: blur(5px);
            border-radius: 10px;
            padding: 10px 2rem;
            margin-bottom: 1rem;
        }

        .user--info{
            display: flex;
            align-items: center;
            gap: 1rem;
        }

        .search-box{
            background: rgba(237,237,237);
            border-radius: 15px;
            color: rgba(113, 99, 186, 255);
            display: flex;
            align-items: center;
            gap: 5px;
            padding: 4px 12px;
        }

        .search-box input{
            background: transparent;
            padding: 10px;
        }

        .search-box i{
            font-size: 1.2rem;
            cursor: pointer;
            transition: all 0.5s ease-out;
        }

        .search-box i:hover{
            transform: scale(1.2);
        }

    @media only screen and (max-width: 1280px) and (min-height:
551px){
      .user {
            display: flex;
            align-items: center;
            justify-content: space-between;
            width: 60px;
```

```
    font-size: 12px;
    padding: 10px;
    border-radius: 8px;
    margin-left: auto;
    margin-right: auto;
    overflow: hidden;
}

.user .user-name {
    display: none;
}

.user-name {
    margin-left: 7px;
}

.user-name h5 {
    font-size: 15px;
    font-weight: 600;
    white-space: nowrap;
}

.user-img {
    width: 30px;
    border-radius: 50%;
    margin: auto;
}

#imageContainer1 {
    width: 35px;
    height: 35px;
    border-radius: 50%;
    background-size: cover;
}

ul {
    list-style: none;
    padding: 0 5px;
}

ul li {
    margin: 20px;
    display: flex;
    align-items: center;
    justify-content: center;
    cursor: pointer;
    border-radius: 5px;
    transition: all 0.5s ease-in-out;
}

ul li i {
    font-size: 13px;
    margin-right: 0px;
}

ul li:hover{
    text-decoration:underline;
}

ul li a {
    white-space: nowrap;
```

```
            text-decoration: none;
            color: white;
            display: none;
            font-size: 13px;
        }

        .side-nav:hover {
            width: 260px;
        }

        .side-nav:hover .user .user-name {
            display: block;
        }

        .side-nav:hover .user {
            width: 100%;
            background: rgba(0,0,0,0.5);
            backdrop-filter: blur(5px);
        }

        .side-nav:hover .user-img{
            margin: 0;
        }

        .side-nav:hover ul li a{
            display: block;
        }

        .side-nav:hover ul li i{
            margin-right: 10px;
        }

        .side-nav:hover ul li{
            justify-content: flex-start;
        }

        .main--content{
            margin-left: 110px; /* Adjusted margin */
            color: white;
            position: relative;
            width: calc(100% - 110px); /* Calculate width based on
side nav */
            padding: 1rem;
            transition: margin-left 0.5s; /* Added transition for
smooth shifting */
            z-index: 0; /* Ensure the main content is behind the
side nav */
        }

        .header--wrapper{
            cursor: pointer;
            display: flex;
            justify-content: space-between;
            align-items: center;
            flex-wrap: wrap;
            background: rgba(0,0,0,0.3);
            backdrop-filter: blur(5px);
            border-radius: 10px;
            padding: 10px 2rem;
            margin-bottom: 1rem;
        }
```

```
        .user--info{
            display: flex;
            align-items: center;
            gap: 1rem;
        }

        .search-box{
            background: rgba(237,237,237);
            border-radius: 15px;
            color: rgba(113, 99, 186, 255);
            display: flex;
            align-items: center;
            gap: 5px;
            padding: 4px 12px;
        }

        .search-box input{
            background: transparent;
            padding: 10px;
        }

        .search-box i{
            font-size: 1.2rem;
            cursor: pointer;
            transition: all 0.5s ease-out;
        }

        .search-box i:hover{
            transform: scale(1.2);
        }
    }

 /* card container */
 .card--container{
 background:rgba(0,0,0,0.3);
 backdrop-filter: blur(5px);
 padding: 2rem;
 border-radius: 10px;
 }

 .card--wrapper{
 display: flex;
 flex-wrap: wrap;
 gap: 1rem;
 }

 .main--title{
 padding-bottom: 10px;
 font-size: 20px;
 }

 .header--title{
color: #EBAB00;
}

 .upload--card{
 background-color: rgba(0,0,0,0.7);
 border-radius: 10px;
 padding: 1.2rem;
 width: 290px;
```

```
 height: 260px;
 display: flex;
 flex-direction: column;
 justify-content: space-between;
 transition: all 0.5s ease-in-out;
 }

  .upload--card:hover{
  transform: translateY(-5px);
  }

 form {
  padding: 20px;
  border-radius: 8px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  color: white
}
 #imageContainer {
  background-color: #ddd; /* Set background color */
  display: flex;
}

 .file-link {
  color: #ffd300; /* Link color */
  cursor: pointer; /* Show pointer cursor on hover */
}

.file-link:hover{
 text-decoration: underline; /* Underline the text */
}

.card--header{
diaplay: flex;
justify-content: space-between;
align-items: center;
margin-bottom: 20px;
}

.upload{
display: flex;
flex-direction: column;
}

.title{
font-size: 15px;
font-weight: 200;
text-align: center;
}

    </style>
</head>
<body>

<!-- header -->
<div class="side-nav">
    <div class="user">
        <div class="user-img">
            <%
                Connection conn = null;
                PreparedStatement pstmt = null;
                ResultSet res = null;
```

```
                try {
                    // Connect to the database
                    Class.forName("com.mysql.cj.jdbc.Driver");
                    conn                                                =
DriverManager.getConnection("jdbc:mysql://localhost:3306/unistay",
"root", "root");

                    // Get the email from the session
                    String       userEmail       =       (String)
session.getAttribute("userEmail");

                    // Prepare and execute the query to retrieve the
Blob data based on email
                    String  sql  =  "SELECT  image  FROM  image  WHERE
email = ?";
                    pstmt = conn.prepareStatement(sql);
                    pstmt.setString(1,  userEmail);  //  Corrected
from 'email' to 'userEmail'
                    res = pstmt.executeQuery();

                    // Check if there are results
                    if (res.next()) {
                        // Get the Blob data from the result set
                        byte[] blobData = res.getBytes("image");

                        // Convert the Blob data to Base64
                        String            base64Image            =
Base64.getEncoder().encodeToString(blobData);

                        // Set the background image of the container
div
                        out.println("<div    id=\"imageContainer1\"
style=\"background-image:      url('"      +
base64Image + "')\"></div>");
                    } else {
                        out.println("upload profile");
                    }
                } catch (Exception e) {
                    e.printStackTrace();
                } finally {
                    // Close resources
                    try {
                        if (res != null) res.close();
                        if (pstmt != null) pstmt.close();
                        if (conn != null) conn.close();
                    } catch (SQLException e) {
                        e.printStackTrace();
                    }
                }
            %>
        </div>
        <div class="user-name">
            <%
                try {
                    Class.forName("com.mysql.cj.jdbc.Driver");
                    Connection                con                =
DriverManager.getConnection("jdbc:mysql://localhost:3306/unistay",
"root", "root");

                    Statement stmt = con.createStatement();
```

```
                    ResultSet rs = stmt.executeQuery("SELECT * FROM
user");

                    // Get the email from the session
                    String       userEmail       =       (String)
session.getAttribute("userEmail");

                    while (rs.next()) {
                        String username = rs.getString("username");
                        String email = rs.getString("email");
                        String            mobileNumber            =
rs.getString("mobilenumber");

                        // Check if the current user's email matches
the one in the session
                        if      (userEmail      !=      null      &&
userEmail.equals(email)) {
                            // Set session attribute for current
user's username
                            session.setAttribute("username",
username);
                            session.setAttribute("mobileNumber",
mobileNumber);
            %>
                            <h5             class="modal-title"><%=
username%></h5>
                            <p class="modal-title"><%= email%></p>
                            <p      class="modal-title">+91      <%=
mobileNumber %></p>
            <%
                            break;  // Exit the loop once the
username is found
                        }
                    }

                    rs.close();
                    stmt.close();
                    con.close();
                } catch (Exception e) {
                    e.printStackTrace();
                }
            %>
        </div>
    </div>
    <ul class="bar">
<li><i               class="bi               bi-pencil-square"></i>
<ahref="EditUserProfile.jsp">Edit your profile</a></li>
<li><i class="bi bi-trash"></i> <a href="DeleteAccount.jsp">Delete
Account</a></li>
<li><i class="bi bi-gift"></i> <a href="#">Rewards</a></li>
<li><i class="bi bi-chat-dots"></i> <a href="#">Messages</a></li>
<li><i    class="bi    bi-suitcase-lg"></i>    <a    href="#">Our
Bookings</a></li>
<li><i class="bi bi-people"></i> <a href="#">Core Member</a></li>
<li><i class="bi bi-gear"></i> <a href="#">settings</a></li>
    </ul>

    <ul>
        <li><i      class="bi      bi-box-arrow-right"></i>      <a
href="logout_action.jsp">Logout</a></li>
    </ul>
```

```
</div>

<!-- Upload Profile pic -->
<div class="main--content">
    <div class="header--wrapper">
        <div class="header--title">
            <a class="nav navbar-brand text-white" href="Home.jsp"
><span class="text-warning">Uni</span>Stay</a>
            <h3>User Dashboard</h3>
        </div>
        <div class="user-info">
            <div class="search-box">
                <i class="bi bi-search"></i>
                <input type="text" placeholder="Search">
            </div>
        </div>
    </div>

    <div class="card--container">
    <h3 class="main--title">Profile</h3>
    <div class="card--wrapper">
    <div class="upload--card">
    <div class="card--header">
    <div class="upload">
    <span class="title"> Upload Profile</span>
     <div class="text-center">
          <form    id="uploadForm"    action="ProFileUploadServlet"
method="post"    enctype="multipart/form-data"    onsubmit="return
validateForm()">
   <div  id="imageContainer"  style="width:  90px;  height:  90px;
border-radius: 50%; overflow: hidden; margin: 0 auto;">
            <img  id="previewImage"  style="max-width:  100%;  max-
height: 100%; display: none;">
          </div>

          <br>
<label  for="profilePic"  class="file-link"  id="fileLabel">Choose
Profile Picture</label>
<input      type="file"      id="profilePic"      name="profilePic"
style="display: none;" onchange="displayFile()">
    <br>
    <input     type="submit"     id="submitBtn"     value="Upload"
style="background:linear-gradient(to  right,  #5d4f08,  #ffd300);
color: white; padding: 10px 20px; border: none; border-radius: 5px;
cursor: pointer;">

</form>
        </div>
    </div>

    </div>
    </div>
    </div>
    </div>
</div>


<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstra
p.bundle.min.js"                                integrity="sha384-
```

```
YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz"
crossorigin="anonymous"></script>
<script>
    const sideNav = document.querySelector('.side-nav');
    const mainContent = document.querySelector('.main--content');

    sideNav.addEventListener('mouseenter', () => {
        mainContent.style.marginLeft = '260px'; // Adjust as per
your side nav width
    });

    sideNav.addEventListener('mouseleave', () => {
        mainContent.style.marginLeft = '110px'; // Adjust as per
your side nav width
    });
</script>

<script>
    function validateForm() {
        // Disable submit button to prevent multiple submissions
        document.getElementById("submitBtn").disabled = true;
        return true; // You can perform additional validation here
if needed
    }

    function displayFile() {
        var input = document.getElementById('profilePic');
        var previewImage = document.getElementById('previewImage');

        if (input.files && input.files[0]) {
            var reader = new FileReader();

            reader.onload = function(e) {
                previewImage.src = e.target.result;
                previewImage.style.display = 'block';
            }

            reader.readAsDataURL(input.files[0]);
        } else {
            previewImage.style.display = 'none';
        }
    }
</script>
</body>
</html>
```

➢ Admin Dashboard

This code snippet demonstrates a Admin Dashboard

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
    <%@   page   import="java.sql.*,   java.awt.image.BufferedImage,
javax.imageio.ImageIO,                java.io.ByteArrayInputStream,
java.util.Base64" %>
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
```

```
    <meta  name="viewport"  content="width=device-width,  initial-
scale=1">
    <title>Admin | DashBoard</title>
  <link                                          rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.11.3/font/bootstrap-icons.min.css">
      <link rel="stylesheet"  href="style.css" >
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootst
rap.min.css"          rel="stylesheet"          integrity="sha384-
T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN"
crossorigin="anonymous">
  <style>
  @import
url('https://fonts.googleapis.com/css2?family=Noto+Serif:wght@300;
400&display=swap');
  *{
  margin: 0;
  padding: 0;
  border: none;
  outline: none;
  box-sizing: border-box;
  }

body{
     width: 100%;
     height: 100vh;
     background-image: linear-gradient(rgba(0, 0, 0, 0.6), rgba(0,
0, 0, 0.6)), url(image/New2.webp);
     background-position: center;
     background-size: cover;
     display: flex;
}

.sidebar{
position: stick;
top: 0;
left: 0;
bottom: 0;
width: 110px;
height: 110vh;
padding: 0 1.7rem;
color: #fff;
overflow: hidden;
transition: all 0.5s linear;
 background: rgba(0, 0, 0, 0.2);
 backdrop-filter: blur(5px);
}

.sidebar:hover{
width: 260px;
transition: 0.5s ease-in-out;
}

.logo{
height: 80px;
padding: 16px;
}

.menu{
height: 85%;
```

```
position: realtive;
list-style: none;
padding: 0;
}

.menu li{
padding: 1rem;
margin: 8px 0;
border-radius: 8px;
transition: all 0.5s ease-in-out;
}

.menu li:hover, .active{
background: #e0e0e058;
}

.menu a{
color: #fff;
font-size: 14px;
text-decoration: none;
display: flex;
align-items: center;
gap: 1.2rem;
}

.menu a span{
overflow: hidden;
}

.menu a i{
font-size: 1.2rem;
}

.menu .logout{
margin-top: 80px;
}

.main--content{
position: relative;
width: 100%;
padding: 1rem;
color: #fff;
}

.header--wrapper img{
width: 50px;
height: 50px;
cursor: pointer;
border-radius: 50%;
}

.header--wrapper{
display: flex;
justify-content: space-between;
align-items: center;
flex-wrap: wrap;
 background: rgba(0,0,0,0.3);
 backdrop-filter: blur(5px);
 border-radius: 10px;
 padding: 10px 2rem;
 margin-bottom: 1rem;
```

```
}

.header--title{
color: #EBAB00;
}

.header--title a{
font-size: 20px;
}

.user-info{
display: flex;
align-items: center;
gap: 1;
}

.search-box{
background: rgb(237, 237, 237);
border-radius: 15px;
color: rgba(113, 99, 186, 255);
display: flex;
align-items: center;
gap: 5px;
padding: 4px 12px;
}

.search-box input{
background: transparent;
padding: 10px;
}

.search-box i{
font-size: 1.2rem;
cursor: pointer;
transition: all 0.5s ease-out;
}

.search-box i:hover{
transform: scale(1.2);
}

 @media only screen and (max-width: 1280px) and (min-height: 551px){
    .sidebar{
position: stick;
top: 0;
left: 0;
bottom: 0;
width: 110px;
height: 150vh;
padding: 0 1.7rem;
color: #fff;
overflow: hidden;
transition: all 0.5s linear;
 background: rgba(0, 0, 0, 0.2);
 backdrop-filter: blur(5px);
}

.sidebar:hover{
width: 260px;
transition: 0.5s ease-in-out;
}
```

```css
.logo{
height: 80px;
padding: 16px;
}

.menu{
height: 85%;
position: realtive;
list-style: none;
padding: 0;
}

.menu li{
padding: 1rem;
margin: 8px 0;
border-radius: 8px;
transition: all 0.5s ease-in-out;
}

.menu li:hover, .active{
background: #e0e0e058;
}

.menu a{
color: #fff;
font-size: 14px;
text-decoration: none;
display: flex;
align-items: center;
gap: 1.2rem;
}

.menu a span{
overflow: hidden;
}

.menu a i{
font-size: 1.2rem;
}

.menu .logout{
margin-top: 80px;
}

.main--content{
position: relative;
width: 100%;
padding: 1rem;
color: #fff;
}

.header--wrapper img{
width: 50px;
height: 50px;
cursor: pointer;
border-radius: 50%;
}

.header--wrapper{
display: flex;
```

```
justify-content: space-between;
align-items: center;
flex-wrap: wrap;
 background: rgba(0,0,0,0.3);
 backdrop-filter: blur(5px);
 border-radius: 10px;
 padding: 10px 2rem;
 margin-bottom: 1rem;
}

.header--title{
color: #EBAB00;
}

.header--title a{
font-size: 20px;
}

.user-info{
display: flex;
align-items: center;
gap: 1;
}

.search-box{
background: rgb(237, 237, 237);
border-radius: 15px;
color: rgba(113, 99, 186, 255);
display: flex;
align-items: center;
gap: 5px;
padding: 4px 12px;
}

.search-box input{
background: transparent;
padding: 10px;
}

.search-box i{
font-size: 1.2rem;
cursor: pointer;
transition: all 0.5s ease-out;
}

.search-box i:hover{
transform: scale(1.2);
}
}

/* card container */
 .card--container{
 background:rgba(0,0,0,0.3);
 backdrop-filter: blur(5px);
 padding: 2rem;
 border-radius: 10px;
 }

 .card--wrapper{
 display: flex;
 flex-wrap: wrap;
```

```
 gap: 1rem;
 }

.main--title{
 padding-bottom: 10px;
 font-size: 20px;
 }

 .user--card{
 background-color: rgba(0,0,0,0.7);
 border-radius: 10px;
 padding: 1.2rem;
 width: 300px;
 height: 150px;
 display: flex;
 flex-direction: column;
 justify-content: space-between;
 transition: all 0.5s ease-in-out;
 }

 .user--card:hover{
 transform: translateY(-5px);
 }

.card--header{
display: flex;
justify-content: space-between;
align-items: center;
margin-bottom: 20px;
}

.user{
display: flex;
flex-direction: column;
}

.title{
font-size: 15px;
font-weight: 200;
}

.user-value{
font-size: 28px;
font-family: "Courier New", Courier, monospace;
font-weight: 600;
}

.icon{
color: #000;
padding: 0.4rem;
height: 60px;
width: 60px;
text-align: center;
border-radius: 50%;
font-size: 2rem;
background: #fff;
}

.card-details{
font-size: 18px;
color: #dcdcdc;
```

```
font-family: "Courier New", Courier, monospace;
}

/* user detail table */
.tabular--wrapper{
 background: rgba(0,0,0,0.3);
 backdrop-filter: blur(5px);
 margin-top: 1rem;
 border-radius: 10px;
 padding: 2rem;
}

.table-container{
width: 100%;
}

table{
width: 100%;
border-collapse: collapse;
}

thead{
background-color: rgba(0,0,0,0.7);
color: #fff;
}

th{
padding: 15px;
text-align: left;
}

tbody{
background-color: rgba(0,0,0,0.2);
}

td {
padding: 15px;
font-size: 14px;
}

tr:nth-child(even){
background: rgba(0,0,0,0.5);
}

tfoot{
background-color: rgba(0,0,0,0.9);
font-weight: bold;
color: #fff;
}

tfoot td{
padding: 15px;
}

.table-container button{
color: green;
background: none;
cursor: pointer;
}
  </style>
  </head>
```

```
<body>
<div class="sidebar">
<div class="logo"></div>
<ul class="menu">
<li class="active">
<a href="#" >
<i class="bi bi-speedometer2"></i>
<span>Dashboard</span>
</a>
</li>

<li>
<a href="#">
<i class="bi bi-person-fill"></i>
<span>Profile</span>
</a>
</li>

<li>
<a href="#">
<i class="bi bi-bar-chart-line-fill"></i>
<span>Statistics</span>
</a>
</li>

<li>
<a href="#">
<i class="bi bi-briefcase-fill"></i>
<span>Careers</span>
</a>
</li>

<li>
<a href="#">
<i class="bi bi-question-circle-fill"></i>
<span>FAQ</span>
</a>
</li>

<li>
<a href="#">
<i class="bi bi-star-fill"></i>
<span>Testimonials </span>
</a>
</li>

<li>
<a href="#">
<i class="bi bi-gear-fill"></i>
<span>Settings</span>
</a>
</li>

<li class="logout">
<a href="#">
<i class="bi bi-box-arrow-right"></i>
<span>Logout</span>
</a>
</li>
</ul>
</div>
```

```
  <!-- main content -->
  <div class="main--content">
  <div class="header--wrapper">
        <div class="header--title">
            <a class="nav navbar-brand text-white" href="index.jsp"
><span class="text-warning">Uni</span>Stay</a>
            <h3>Admin Dashboard</h3>
        </div>
        <div class="user-info">
            <div class="search-box">
                <i class="bi bi-search"></i>
                <input type="text" placeholder="Search">
            </div>
             <img      src="image/unistay-high-resolution-logo.png"
alt="...">
        </div>
    </div>

    <!-- card-container -->
  <div class="card--container">
  <h3 class="main--title">Status</h3>
  <div class="card--wrapper">
  <div class="user--card">
  <div class="card--header">
  <div class="user">
  <span class="title">User Login Data</span>
  <span class="user-value">
   <%
        String  url  =  "jdbc:mysql://localhost:3306/unistay";  //
Replace "your_database_name" with your actual database name
        String username1 = "root"; // Replace "your_username" with
your actual database username
        String password = "root"; // Replace "your_password" with
your actual database password

        try {
            Class.forName("com.mysql.jdbc.Driver");  //  Load  the
MySQL JDBC driver
            Connection                connection                =
DriverManager.getConnection(url, username1, password);
            Statement statement = connection.createStatement();
            ResultSet  resultSet  =  statement.executeQuery("SELECT
COUNT(*) FROM user"); // Replace "user" with your actual table name

            if (resultSet.next()) {
                int count = resultSet.getInt(1);
    %>
               <%= count %>
    <%
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    %>
  </span>
  </div>
  <i class="bi bi-people-fill icon"></i>
  </div>
  <span class="card-details">Registered User</span>
  </div>
```

```
<!-- Property listing no. -->
<div class="user--card">
<div class="card--header">
<div class="user">
<span class="title">Property Listing</span>
<span class="user-value">
 0
</span>
</div>
<i class="bi bi-list-stars icon"></i>
</div>
<span class="card-details">Registered Properties</span>
</div>


<!-- Booking Status -->
<div class="user--card">
<div class="card--header">
<div class="user">
<span class="title">Total No. Of Bookings</span>
<span class="user-value">
 0
</span>
</div>
<i class="bi bi-building-add icon"></i>
</div>
<span class="card-details">Booking Status</span>
</div>

<!-- Booking Status -->
<div class="user--card">
<div class="card--header">
<div class="user">
<span class="title">Messages</span>
<span class="user-value">
 0
</span>
</div>
<i class="bi bi-chat-dots-fill icon"></i>
</div>
<span class="card-details">For Enquiry </span>
</div>
</div>
</div>

<!-- table -->
<div class="tabular--wrapper">
<h3 class="main--title">User's Details</h3>
<div class="table-container">
  <table>
    <thead>
      <tr>
        <th>Username</th>
        <th>Contact Number</th>
        <th>Email id</th>
        <th>Edit</th>
      </tr>
    </thead>
    <tbody>
      <%
```

```
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection                          con                          =
DriverManager.getConnection("jdbc:mysql://localhost:3306/unistay",
"root", "root");

            Statement stmt = con.createStatement();
            ResultSet rs = stmt.executeQuery("SELECT * FROM user");

            while (rs.next()) {
                String username = rs.getString("username");
                String mobileNumber = rs.getString("mobilenumber");
                String email = rs.getString("email");
        %>
                <tr>
                    <td><%= username %></td>
                    <td><%= mobileNumber %></td>
                    <td><%= email %></td>
                    <td><button>Edit</button></td>
                </tr>
        <%
            }
            rs.close();
            stmt.close();
            con.close();
        } catch(Exception e) {
            e.printStackTrace();
        }
        %>
      </tbody>
      <tfoot>
      <tr>
      <td  colspan="4">Total:  <%
        String  url1  =  "jdbc:mysql://localhost:3306/unistay";  //
Replace "your_database_name" with your actual database name
        String username2 = "root"; // Replace "your_username" with
your actual database username
        String password1 = "root"; // Replace "your_password" with
your actual database password

        try {
            Class.forName("com.mysql.jdbc.Driver");  //  Load  the
MySQL JDBC driver
            Connection                  connection                  =
DriverManager.getConnection(url1, username2, password1);
            Statement statement = connection.createStatement();
            ResultSet  resultSet  =  statement.executeQuery("SELECT
COUNT(*) FROM user"); // Replace "user" with your actual table name

            if (resultSet.next()) {
                int count = resultSet.getInt(1);
    %>
               <%= count %>
    <%
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    %></td>
      </tr>
      </tfoot>
```

```
    </table>
  </div>
</div>

  </div>


    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstra
p.bundle.min.js"                                    integrity="sha384-
YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz"
crossorigin="anonymous"></script>
  </body>
</html>
```

➢ Profile Picture Upload

This code snippet demonstrates a Profile Picture Upload

```
package project;

import java.io.IOException;

import java.io.InputStream;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.SQLException;

import jakarta.servlet.annotation.MultipartConfig;

import jakarta.servlet.ServletException;

import jakarta.servlet.http.HttpServlet;

import jakarta.servlet.http.HttpServletRequest;

import jakarta.servlet.http.HttpServletResponse;

import jakarta.servlet.http.Part;


@MultipartConfig(

   fileSizeThreshold = 1024 * 1024 * 2,  // 2MB

   maxFileSize = 1024 * 1024 * 50,       // 50MB

   maxRequestSize = 1024 * 1024 * 100    // 100MB

)

public class ProFileUploadServlet extends HttpServlet {

   private static final long serialVersionUID = 1L;
```

```
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // Database connection details
        String url = "jdbc:mysql://localhost:3306/unistay";
        String user = "root";
        String password = "root";

        // Get the uploaded file
        Part filePart = request.getPart("profilePic");
        InputStream inputStream = filePart.getInputStream();

        // Get the email from the session
        String userEmail = (String) request.getSession().getAttribute("userEmail"); // Updated from "username" to "email"

        Connection conn = null;
        String message = null;

        try {
            // Connect to the database
            Class.forName("com.mysql.cj.jdbc.Driver");
            conn = DriverManager.getConnection(url, user, password);

            // Delete the existing image for this email
            deleteExistingImage(conn, userEmail); // Updated from "email" to "userEmail"

            // Create SQL query to insert the image into the database
            String sql = "INSERT INTO image (image, email) VALUES (?, ?)";
            PreparedStatement statement = conn.prepareStatement(sql);
            statement.setBlob(1, inputStream);
            statement.setString(2, userEmail); // Updated from "email" to "userEmail"

            // Execute the query
            int row = statement.executeUpdate();
            if (row > 0) {
```

```
                message = "Image uploaded and saved into database";
            }
        } catch (SQLException | ClassNotFoundException ex) {
            message = "ERROR: " + ex.getMessage();
            ex.printStackTrace();
        } finally {
            if (inputStream != null) {
                try {
                    inputStream.close();
                } catch (IOException ex) {
                    ex.printStackTrace();
                }
            }
            if (conn != null) {
                try {
                    conn.close();
                } catch (SQLException ex) {
                    ex.printStackTrace();
                }
            }
            request.setAttribute("Message", message);
            getServletContext().getRequestDispatcher("/user/MyProfile.jsp").forward(request,
response);
        }
    }
    // Method to delete the existing image for the given email
    private void deleteExistingImage(Connection conn, String userEmail) throws SQLException { //
Updated from "email" to "userEmail"
        String sql = "DELETE FROM image WHERE email = ?";
        PreparedStatement statement = conn.prepareStatement(sql);
        statement.setString(1, userEmail); // Updated from "email" to "userEmail"
        statement.executeUpdate();
    }
}
```

GitHub Repository Link:

https://github.com/ShubhangiShahi02/UniStay.git