PUSL2020 Software Development Tools and Practices

Project Report

Submission Deadline: April 08th 2024

Module Leader: Dr. Rasika Ranaweera | Ms. Pavithra Subhashini

	Student ID (Plymouth)	Name (as appeared on DLE)
1.	**10898536**	**Henaka Kumara**
2.	**10908162**	**Horagala Piyumani**
3.	**10898563**	**Weragampita Miyuranga**
4.	**10900374**	**Wickramaarachchi**
5.	**10898567**	**Dilanka Nagasinghe**
6.	**10749144**	**Kaluthanthiri Patabandi**

Table of Contents

Introduction

our newly created website, which aims to help new students locate appropriate housing close to campus. We created this platform with our new warden's assistance to make sure that incoming students have an easy transition.

Finding the ideal housing and relocating to a new campus can be difficult tasks, particularly for people who are not familiar with the area. We hope to streamline this process with our website by offering a thorough list of all the lodgings that are available, along with important details about each one.

Our platform has several tools that make finding lodging easier. Users can peruse listings according to criteria including amenities, price range, and location. Furthermore, our website offers helpful information about the community, including adjacent amenities, transportation alternatives, and safety concerns.

Under the direction of our new warden, we place equal emphasis on helping students feel supported and connected to one another as well as on locating appropriate accommodation. Through our website, prospective students can get in touch with present residents, pose queries, and get advice from seasoned people who are aware of the difficulties associated with adjusting to a new setting.

Objectives

1. Promote a Smooth Transition

- One of the main goals of the website is to promote a smooth transition for incoming students by offering an easy-to-use platform for locating appropriate housing options close to campus.

2. Comprehensive Directory

- Provide students with access to a variety of housing options that suit their preferences, lifestyle, and financial constraints by providing a comprehensive directory of lodgings that are available.

3. Advice from the Warden

- Make sure the new warden's knowledge and experience steers the accommodation search process and gives students insightful advice to help them choose their living arrangements.

4. Community Building

- Create a feeling of belonging for new students by helping them get in touch with existing residents and providing them with possibilities for support and interaction via the platform.

5. Safety and Security

- Give students safety and security priority by informing them of the safety precautions taken for each kind of housing and drawing their attention to any pertinent safety issues in the community.

6. Transparency and Information

- Make sure that all lodging options are transparent by giving accurate and current information about them, including specifics about amenities, rental agreements, and any extra costs.

7. Inclusivity and Accessibility

- Make sure that all students, including those with disabilities, can access the website, and that accommodations are available to suit a variety of requirements and preferences, encouraging inclusivity and equal opportunity for all students.

8. User input and improvement

- To make sure the website is still a useful tool for upcoming generations of students, it is important to regularly collect user feedback, identify areas for improvement, and apply changes.

Development Overview

Our team comprised six dedicated members who collaborated closely to develop the web application for NSBM Green University. Each member was assigned specific roles and responsibilities to ensure the smooth progress of the project. Below are the details of the team members along with their assigned roles and responsibilities:

1. Henaka Kumara (10898536):

Role: Project Manager and backend developer

Responsibilities:

- Coordination of team activities
- Task assignment and tracking
- Communication with stakeholders
- Overall project management
- Developing APIs for data retrieval and manipulation

2. Horagala Piyumani (10908162):

Role: Frontend Developer

Responsibilities:

- Designing and implementing user interfaces

- Ensuring responsiveness and user-friendliness of the application
- Collaborating with backend developers to integrate frontend and backend functionalities

3. Weragampita Miyuranga (10898563):

Role: Backend Developer

Responsibilities:

- Implementing server-side logic
- Developing APIs for data retrieval and manipulation
- Database design and management

4. Wickramaarachchi (10900374):

Role: Quality Assurance Engineer

Responsibilities:

- Designing test cases
- Conducting thorough testing of the application
- Identifying and documenting bugs and issues
- Ensuring the overall quality and reliability of the software product

5. Dilanka Nagasinghe (10898567):

Role: Database Administrator

Responsibilities:

- Database setup and configuration
- Ensuring data integrity and security
- Performance optimization and maintenance of the database

6. Kaluthanthiri Patabandi (10749144):

Role: Technical Support and Documentation Specialist

Responsibilities:

- Providing technical support to team members
- Documenting project progress and updates
- Creating user manuals and documentation for the web application

Each team member played a crucial role in the development process, contributing their expertise and skills to achieve the project's objectives. The coordinated effort and effective collaboration among team members were instrumental in successfully completing the development of the web application within the specified timeline.

Test Case Design

Our test case design process aimed to thoroughly evaluate the functionality and performance of the web application across various scenarios. Test cases were meticulously designed to cover all aspects of the system, including user interactions, data validation, and error handling. Each test case was categorized based on specific user roles and functionalities to ensure comprehensive coverage. Below are details of the test case design process.

1. Landlord Functionality

- **Test Case 1:** Landlord Registration
- **Description:** Verify that landlords can register for an account successfully.
- **Steps:**
    1. Navigate to the registration page.
    2. Fill in the required details.
    3. Submit the registration form.

- **Expected Outcome:** Landlord account is created, and the user is redirected to the login page.

- **Test Case 2:** Property Management
- **Description:** Ensure landlords can add, update, and delete their properties.
- **Steps:**
    1. Log in to the landlord account.
    2. Add a new property with relevant details.
    3. Update property information.
    4. Delete a property from the account.
- **Expected Outcome:** Property management actions are successfully executed without errors.

- **Test Case 3:** Request Handling
- **Description:** Test the functionality for landlords to accept or reject reservation requests from students.
- **Steps:**
    1. Receive a reservation request notification.
    2. Review the request details.
    3. Accept or reject the reservation.

- **Expected Outcome:** Landlords can accept or reject reservation requests, and notifications are sent to students accordingly.

2. Web Master Functionality (Includes Warden Responsibilities)

- **Test Case 4:** Advertisement Viewing and Approval
- **Description:** - Ensure the web master can view advertisements posted by landlords and approve or reject them.
- **Steps:**
    1. Log in to the web master account.

2. Navigate to the advertisement section.
3. View advertisements as a list and on a map.
4. Select an individual property advertisement.
5. Review property details.
6. Approve or reject the advertisement.

- **Expected Outcome:** Web master can view advertisements, approve or reject them, and notifications are sent to landlords accordingly.

3. Student Functionality

- **Test Case 5:** Advertisement Browsing and Reservation
- **Description:** Verify that students can browse the list of advertisements, explore property details, and reserve a property.
- **Steps:**
    1. Log in to the student account.
    2. Navigate to the advertisement section.
    3. Browse advertisements and select a property for more details.
    4. Fill in reservation details.
    5. Submit the reservation request.
- **Expected Outcome:** Students can view advertisements, access property details, and submit reservation requests successfully.

- **Test Case 6:** House Tour Booking
- **Description:** Test the functionality for students to book a house tour for a selected property.
- **Steps:**
    1. Log in to the student account.
    2. Navigate to the selected property details page.
    3. Find the option to book a house tour.
    4. Select a suitable date and time for the tour.
    5. Submit the house tour booking request.

- **Expected Outcome:** Students can successfully book a house tour for the selected property, and notifications are sent to the landlord.

4. Web Master/Admin Functionality (Additional)

- **Test Case 7:** Account Creation and Article Posting
- **Description:** Verify that the web master can create accounts for landlords, web master (warden role), and students, and post articles.
- **Steps:**
    1. Log in to the web master account.
    2. Navigate to the account creation section.
    3. Create accounts for landlords, web master (warden role), and students.
    4. Navigate to the article posting section.
    5. Create and publish an article.
- **Expected Outcome:** User accounts are created successfully with the specified roles, and articles are successfully posted and visible to users on the website.

Each test case was meticulously designed to validate specific functionalities and user interactions within the web application. By systematically executing these test cases, we aimed to ensure the reliability and accuracy of the software product while meeting the requirements outlined in the project scenario.

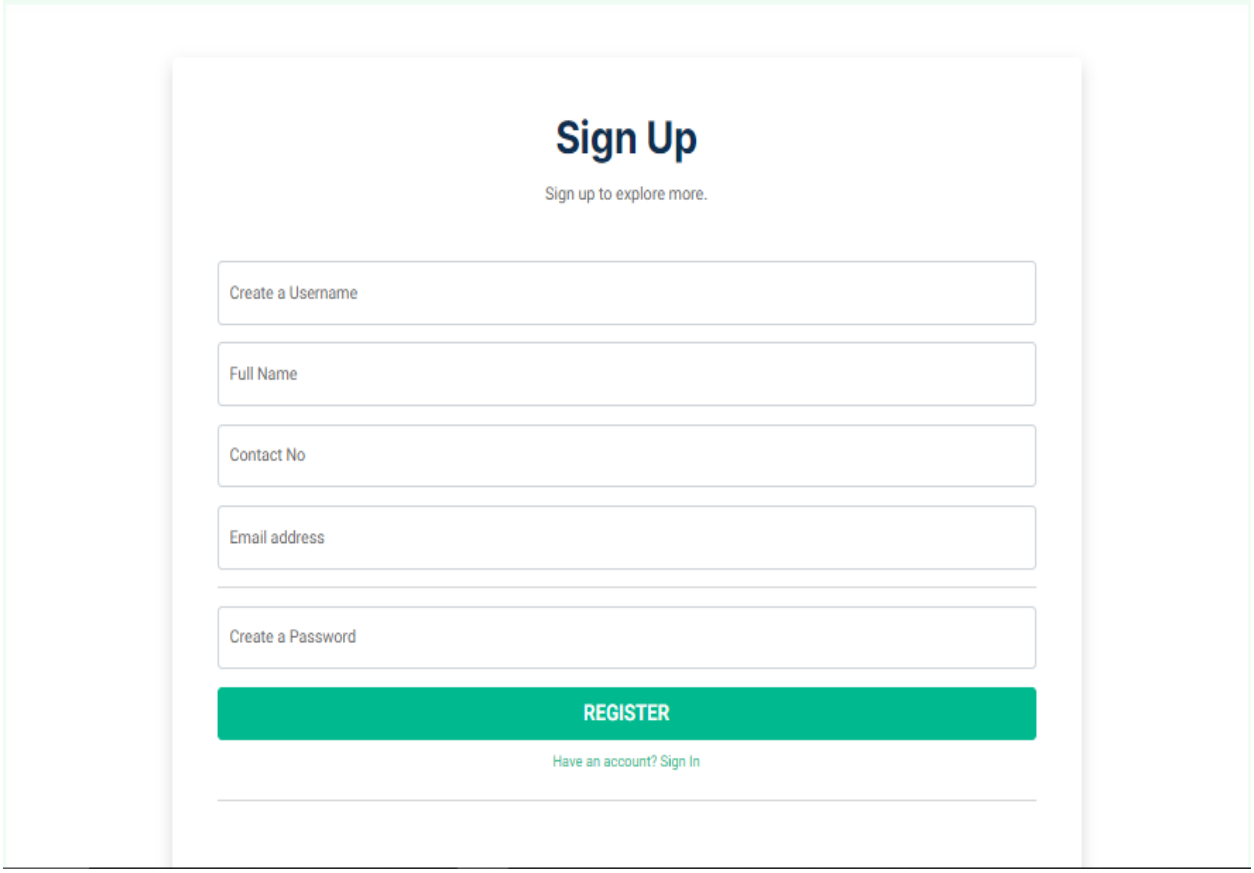
Evidence of Development

To provide comprehensive evidence of our development progress, we have captured screenshots of various pages and functionalities of the web application. Each screenshot is accompanied by a summary of the depicted feature and a list of developers involved in its implementation. This documentation serves to showcase the evolution of the software product and the contributions of individual team members. Below are the details of the evidence of development.

1. Front-end

i. Registration Pages

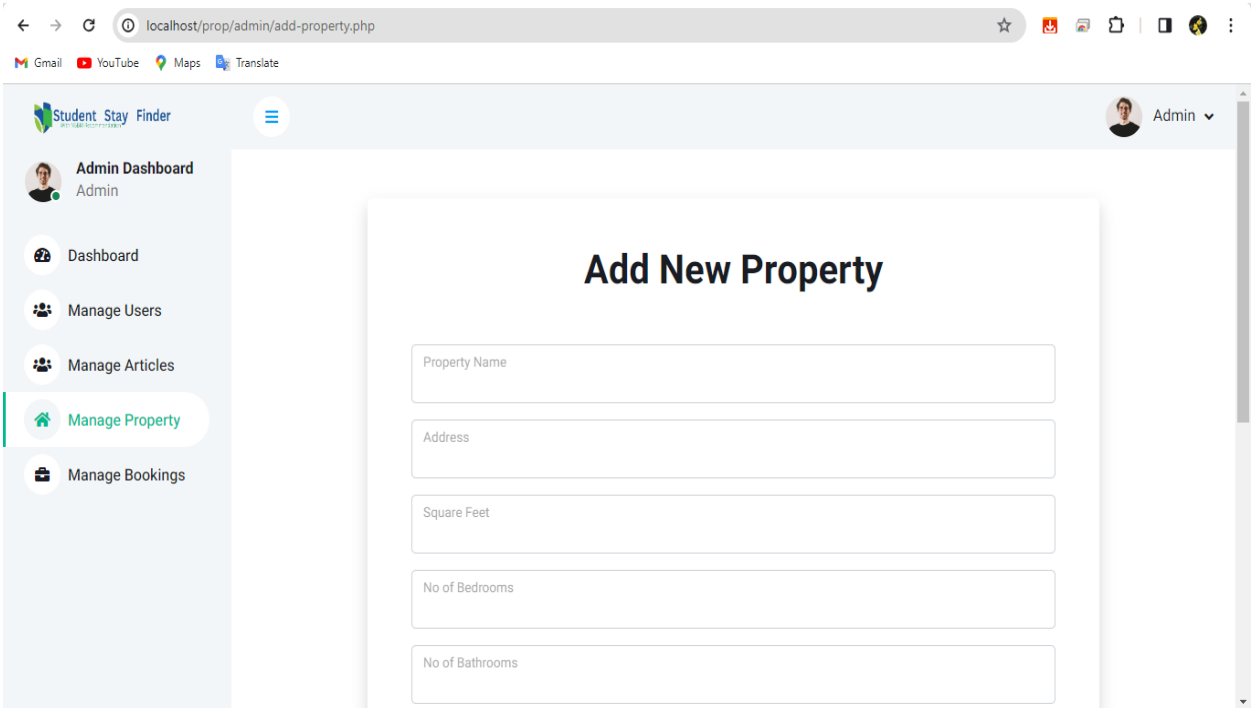
- **Summary:** This screenshots showcases the landlord and student registration pages where landlords and students can sign up for a new account by providing necessary details.
- **Developers:** Developed by Horagala Piyumani and Wickramaarachchi.

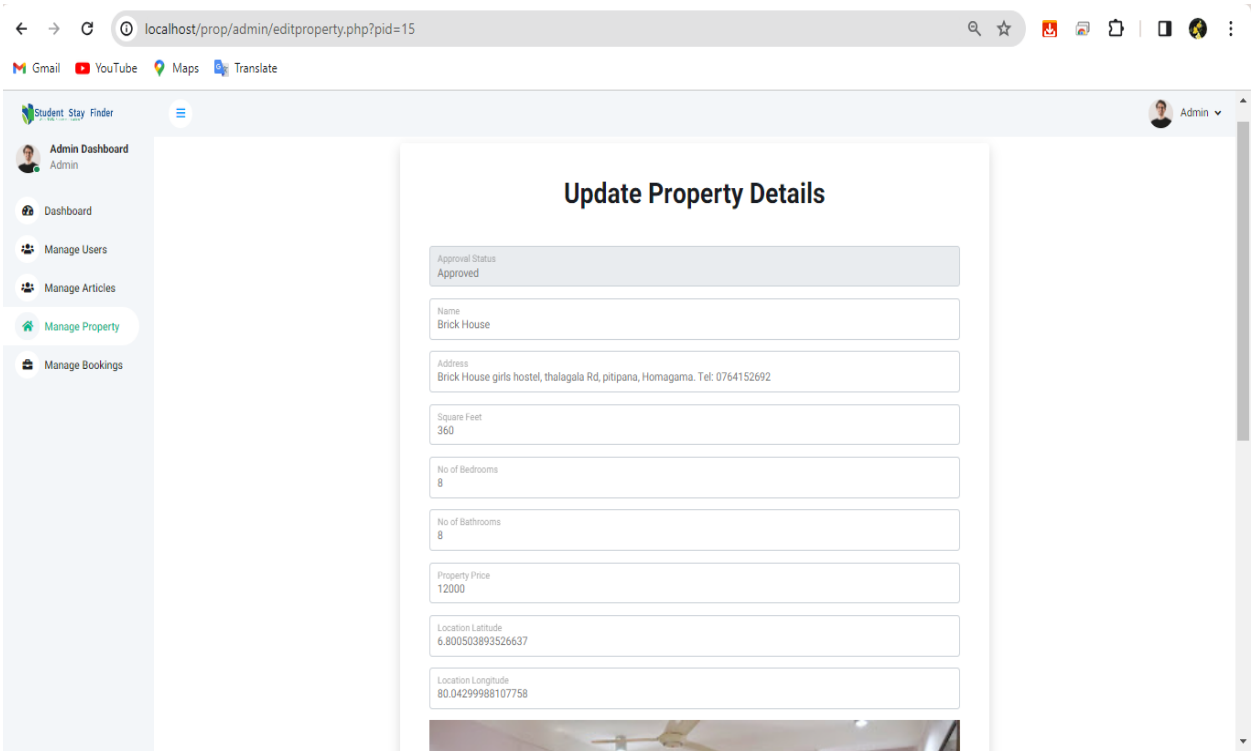


ii. Property Management Dashboard

- **Summary:** This screenshot displays the property management dashboard accessible to landlords upon login. Landlords can add, update, and delete their properties from this dashboard.

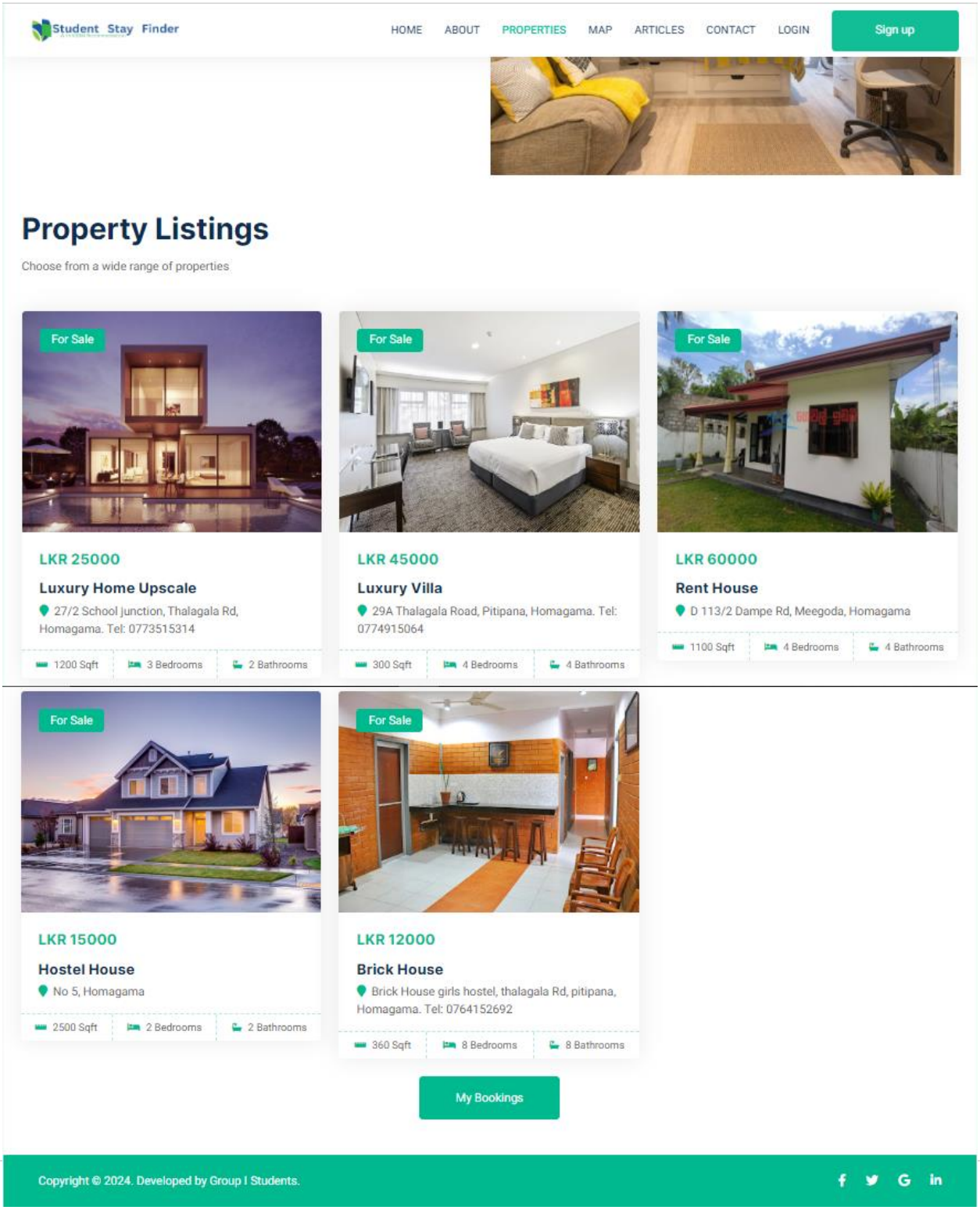
- **Developers:** Developed by Weragampita Miyuranga and Dilanka Nagasinghe.





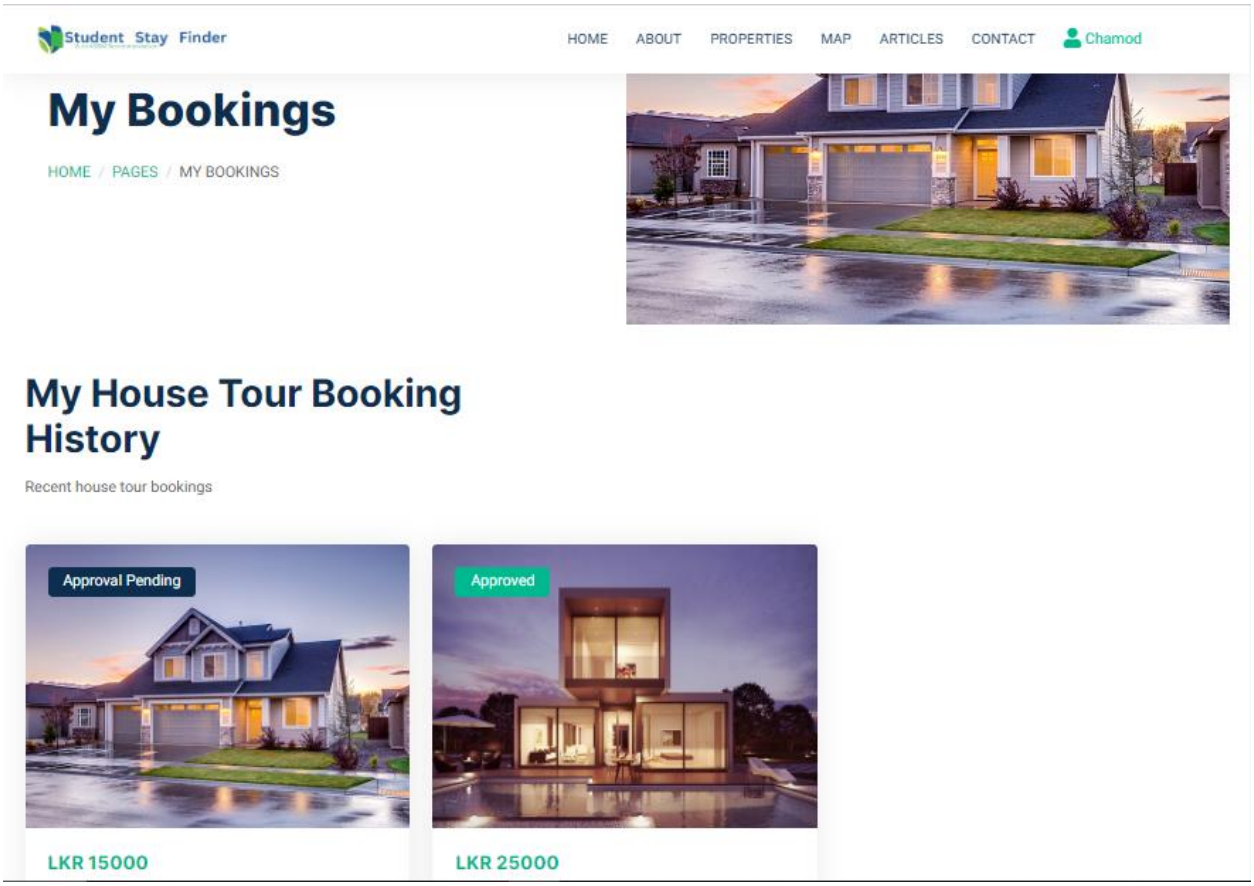
iii. Advertisement Viewing Interface

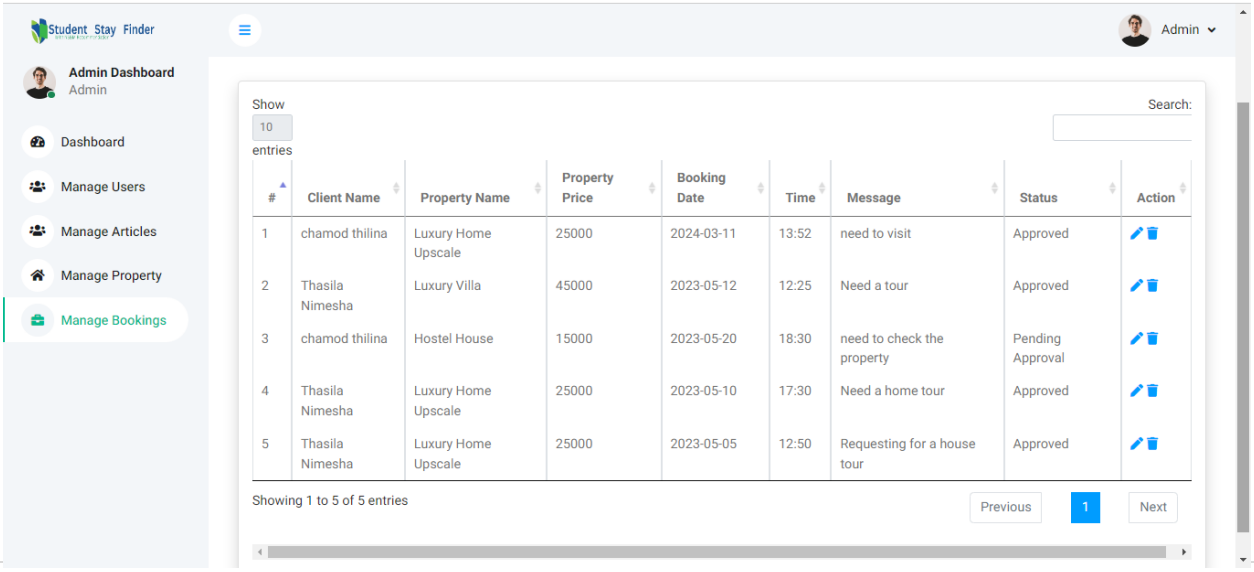
- **Summary:** This screenshot exhibits the interface where the web master (warden) can view advertisements posted by landlords. The web master can view advertisements as a list and on a map.
- **Developers:** Developed by Henaka Kumara and Kaluthanthiri Patabandi.



iv. Reservation Request Handling Page

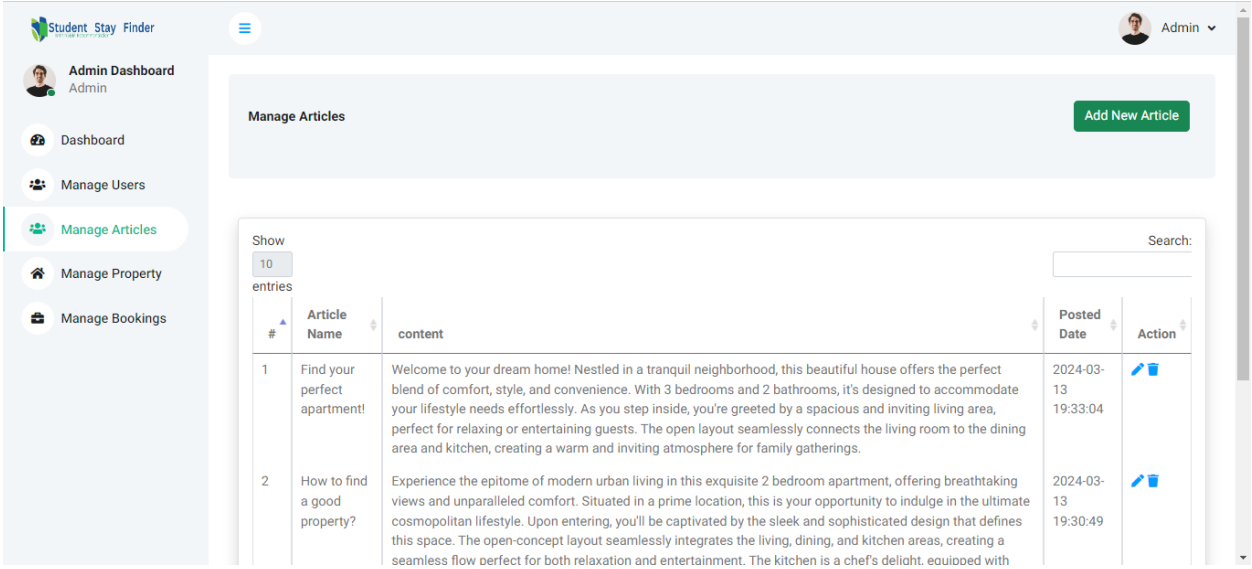
- **Summary:** This screenshot showcases the page where landlords can manage reservation requests from students. Landlords can accept or reject reservation requests and communicate with students.
- **Developers:** Developed by Weragampita Miyuranga and Wickramaarachchi.

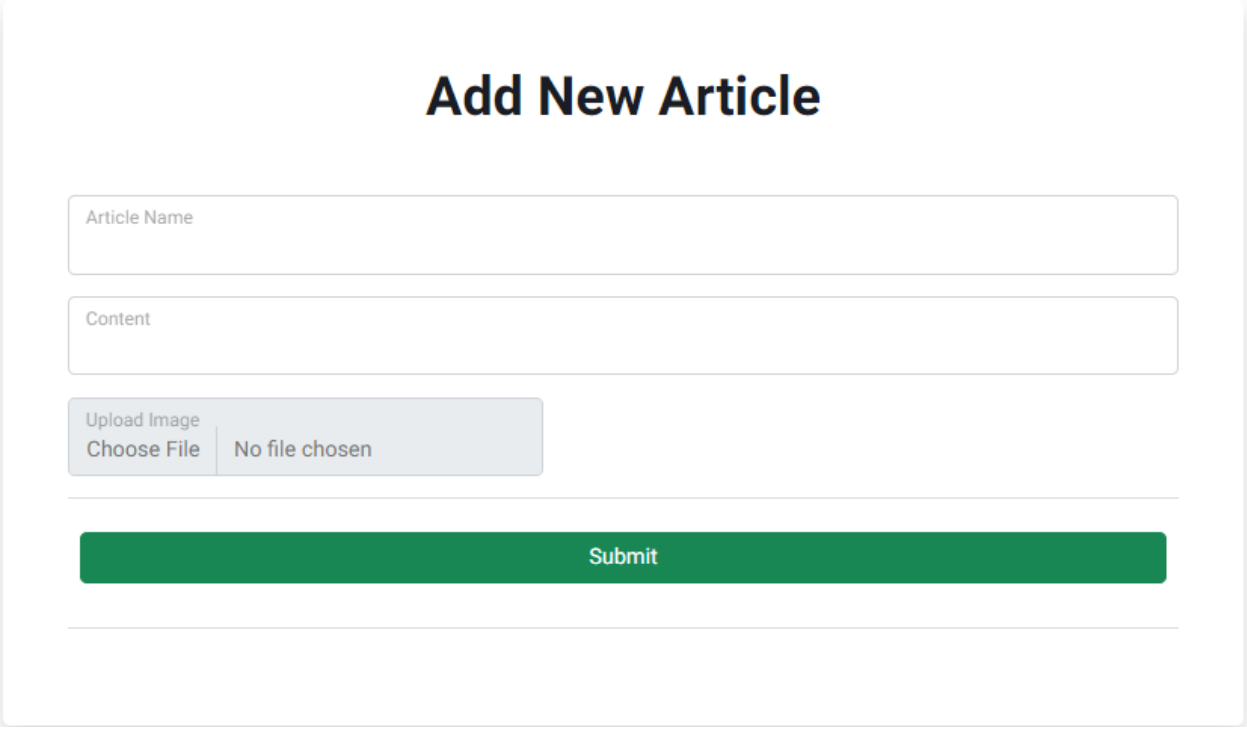


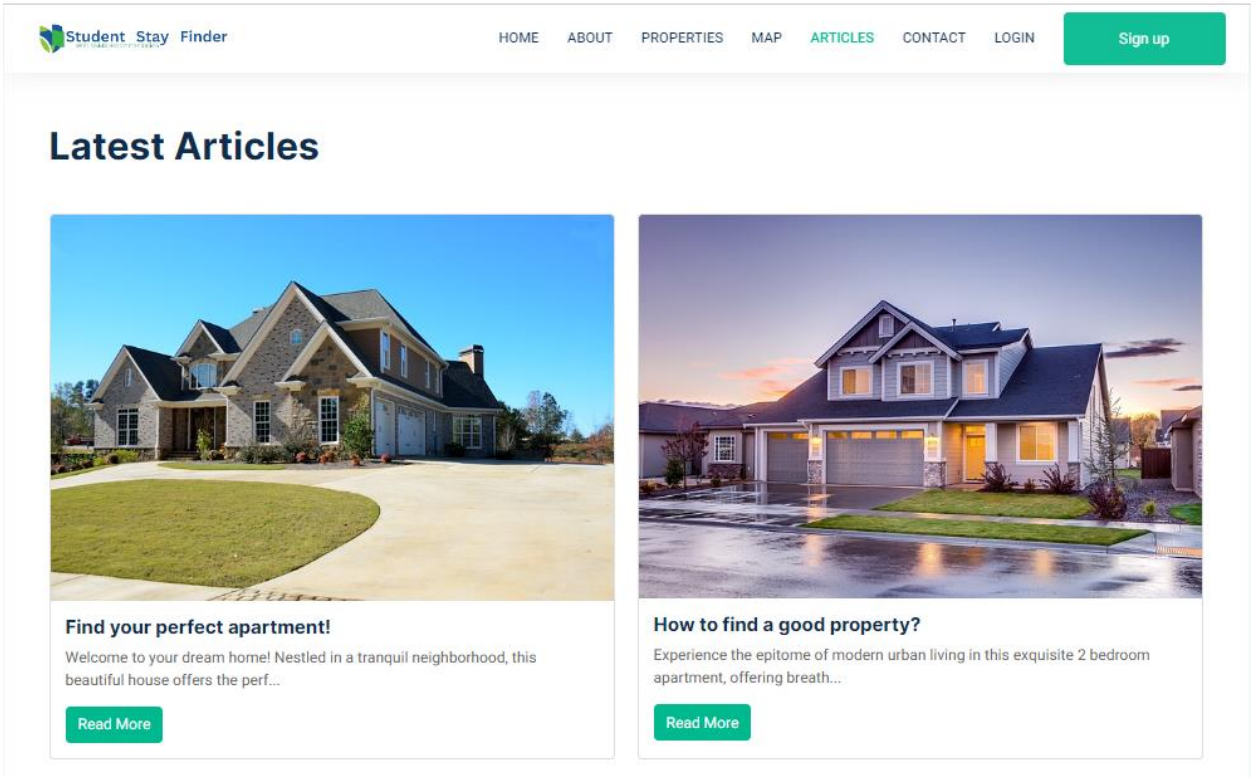


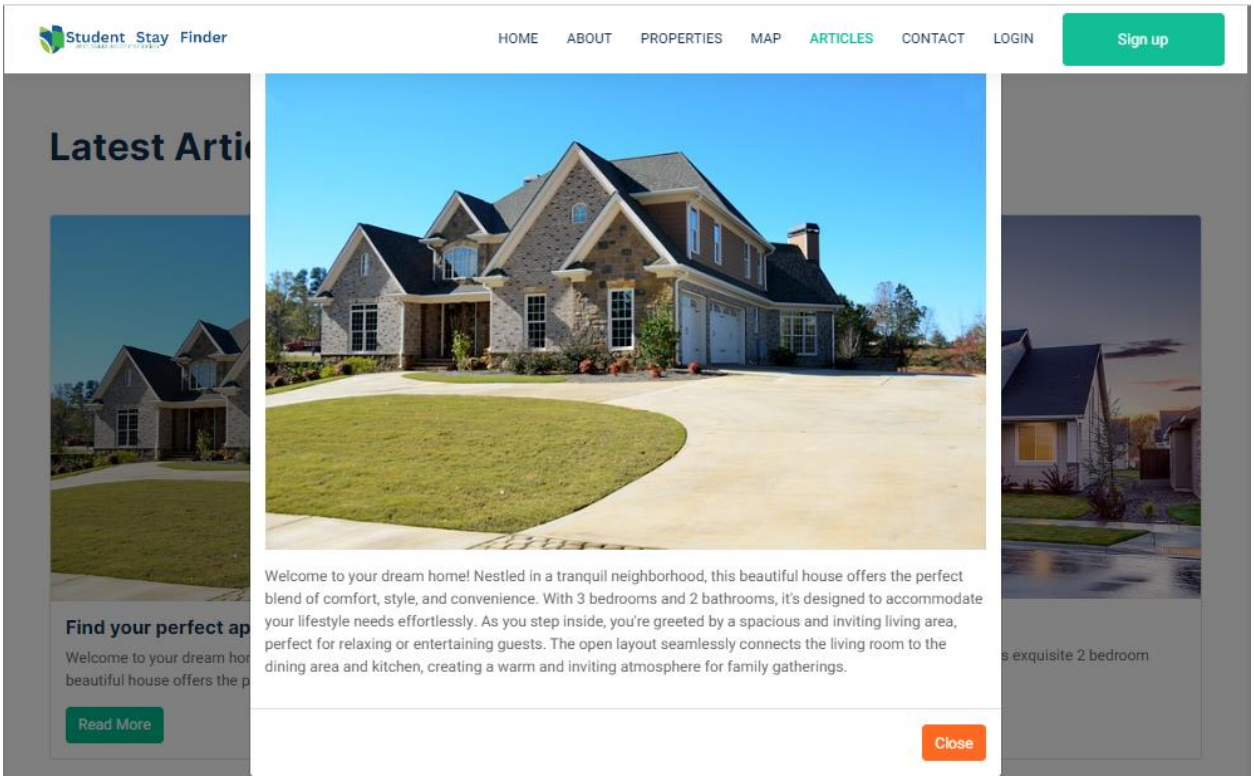
v. Article Posting Section

- **Summary:** This screenshot displays the article posting section accessible to the web master (warden) for sharing information and updates with users.
- **Developers:** Developed by Henaka Kumara and Kaluthanthiri Patabandi.



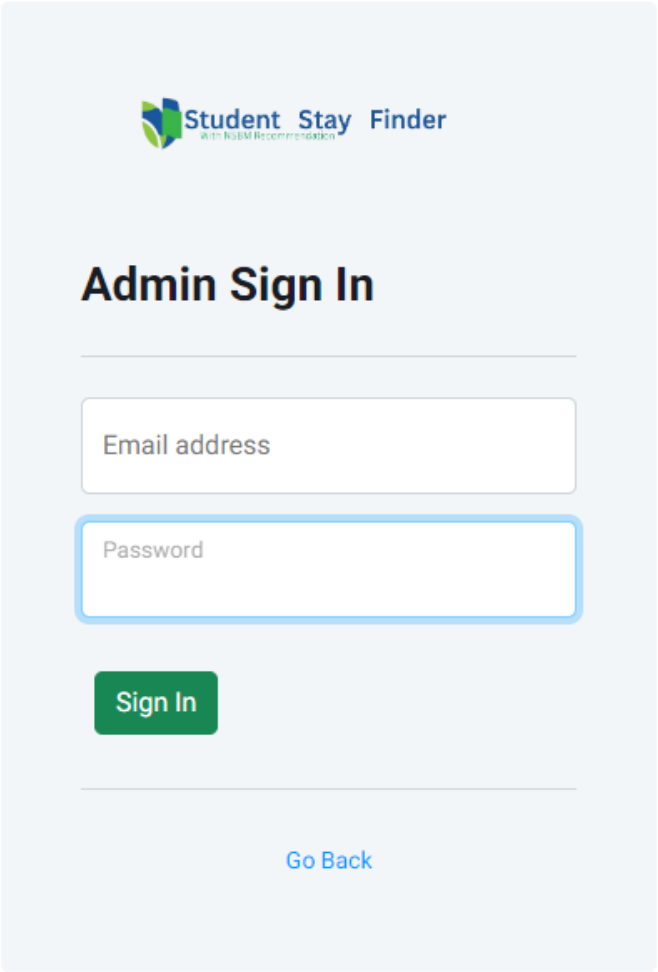


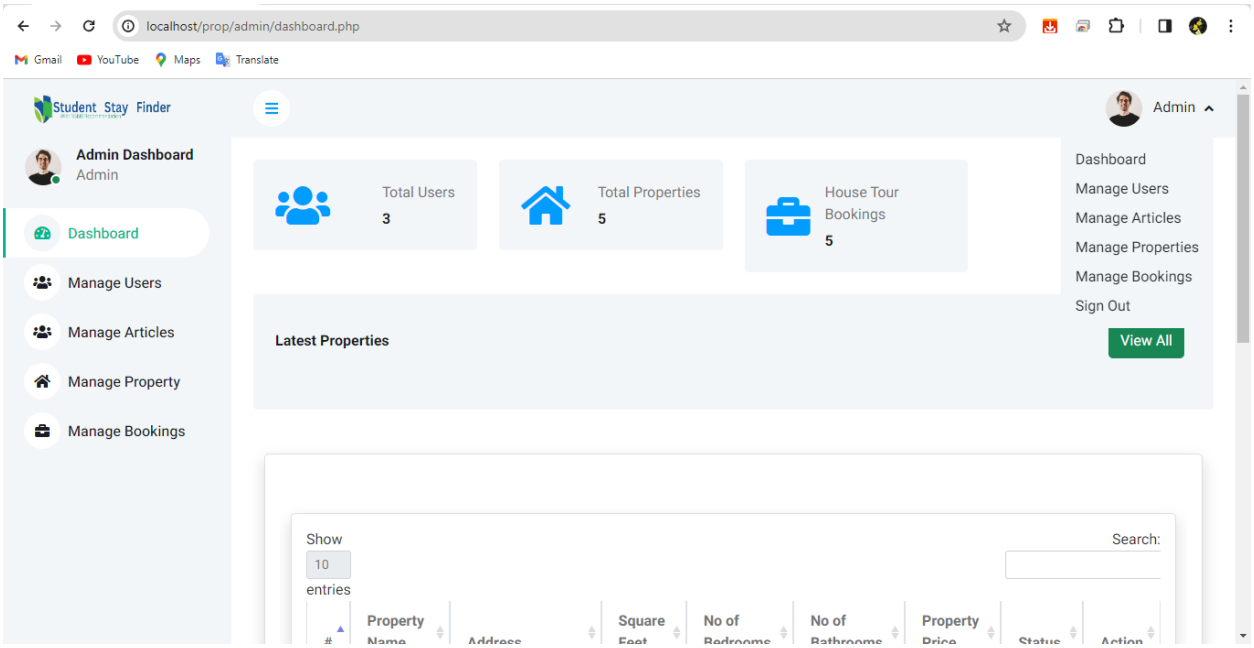

Student Stay Finder
HOME
ABOUT
PROPERTIES
MAP
ARTICLES
CONTACT
LOGIN
Sign up
Latest Articles
Find your perfect apartment!
Welcome to your dream home! Nestled in a tranquil neighborhood, this beautiful house offers the perf...
Read More
How to find a good property?
Experience the epitome of modern urban living in this exquisite 2 bedroom apartment, offering breath...
Read More


Student Stay Finder
HOME
ABOUT
PROPERTIES
MAP
ARTICLES
CONTACT
LOGIN
Sign up
Welcome to your dream home! Nestled in a tranquil neighborhood, this beautiful house offers the perfect blend of comfort, style, and convenience. With 3 bedrooms and 2 bathrooms, it's designed to accommodate your lifestyle needs effortlessly. As you step inside, you're greeted by a spacious and inviting living area, perfect for relaxing or entertaining guests. The open layout seamlessly connects the living room to the dining area and kitchen, creating a warm and inviting atmosphere for family gatherings.
Close

vi. Admin Panel

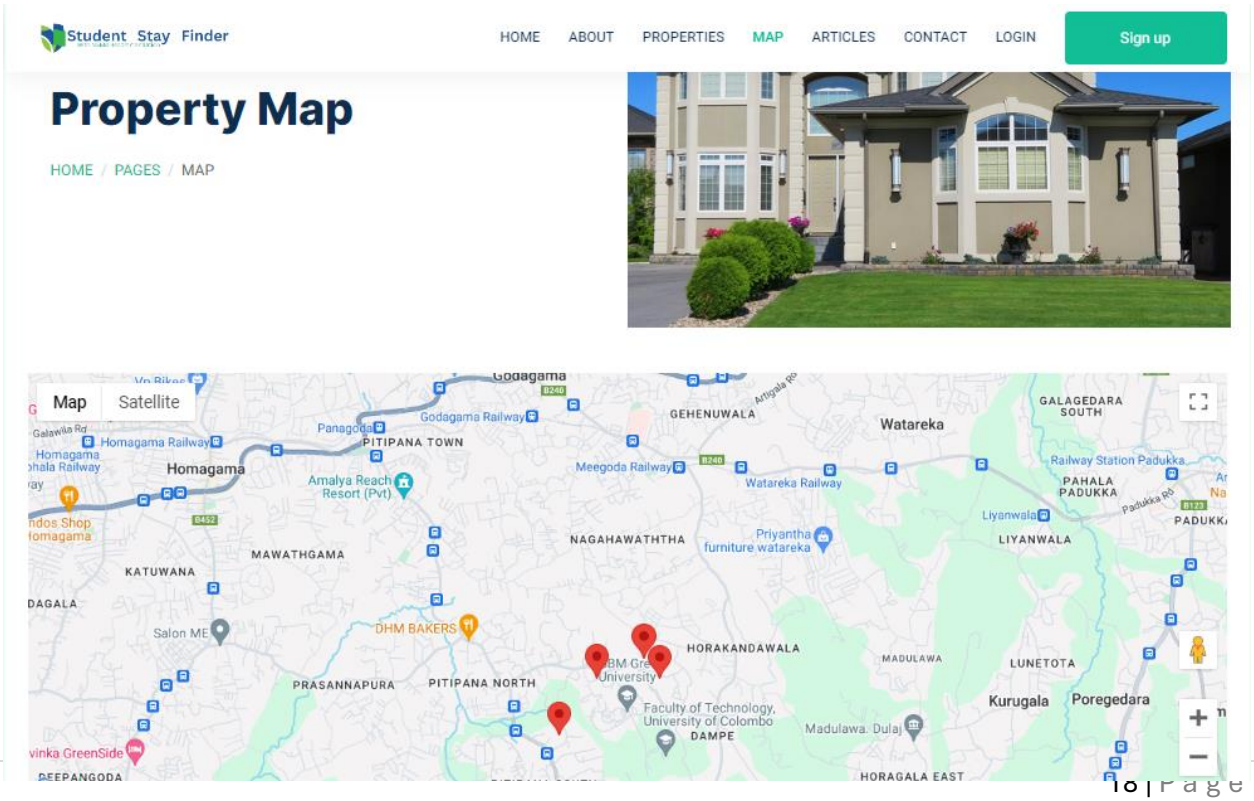
- **Summary:** This screenshot showcases the admin panel, providing the web master (warden) with access to various administrative functionalities, such as user management and system settings.
- **Developers:** Developed by Henaka Kumara, Kaluthanthiri Patabandi and Horagala Piyumani.

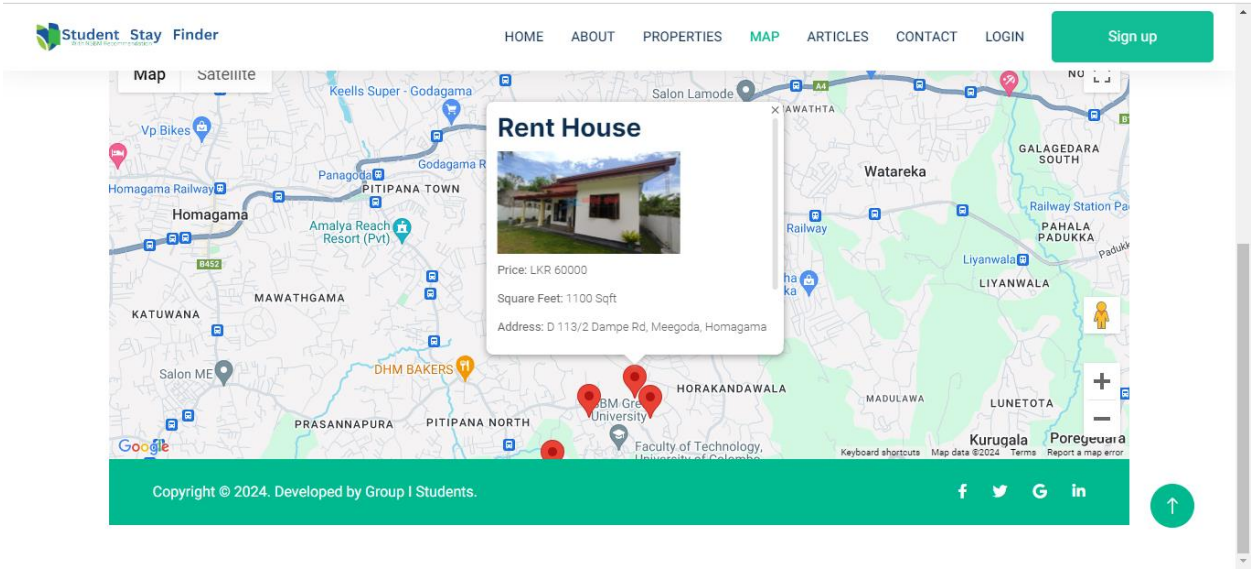


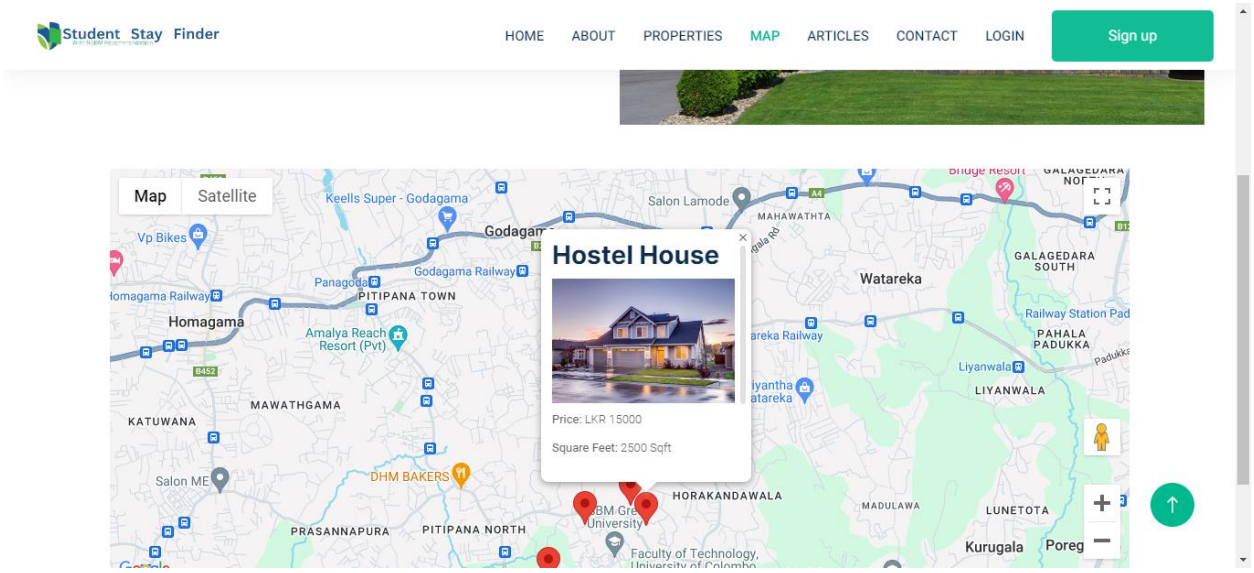


vii. Map Locator Page

- **Summary:** This screenshot displays the map locator page where users can view the locations of available accommodations on a map. Users can interact with markers to access property details.
- **Developers:** Developed by All Members.



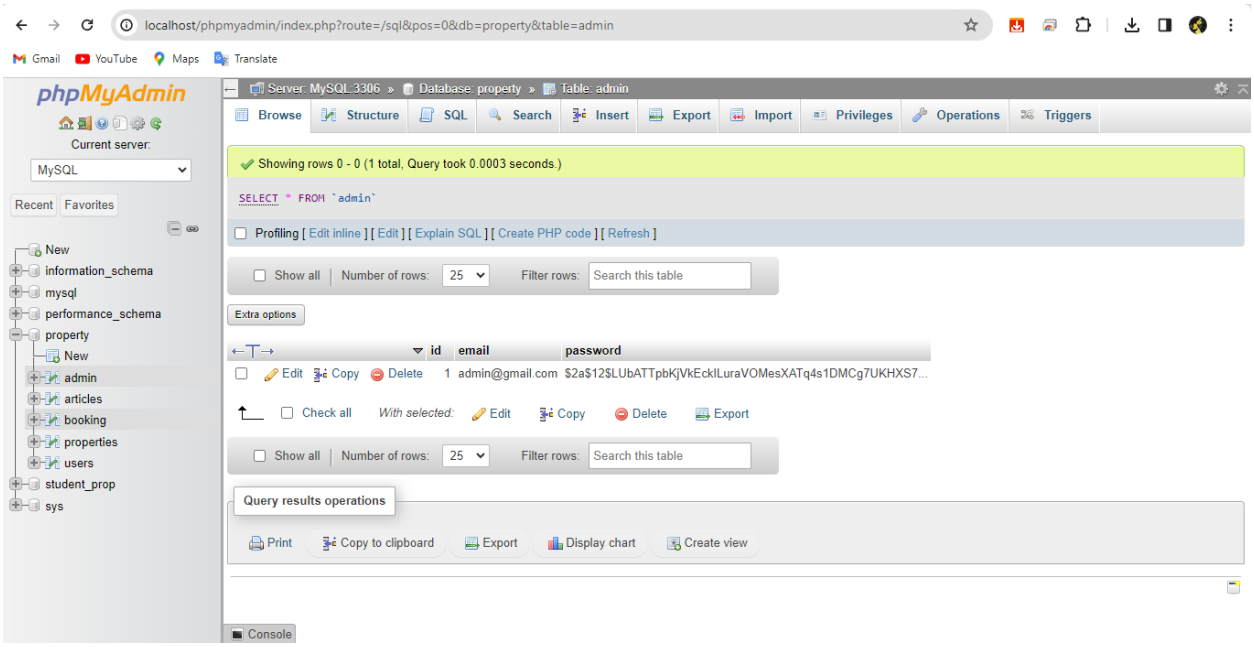

Student Stay Finder
HOME
ABOUT
PROPERTIES
MAP
ARTICLES
CONTACT
LOGIN
Sign up
Map
Satellite
Rent House
Price: LKR 60000
Square Feet: 1100 Sqft
Address: D 113/2 Dampe Rd, Meegoda, Homagama
Copyright © 2024. Developed by Group I Students.

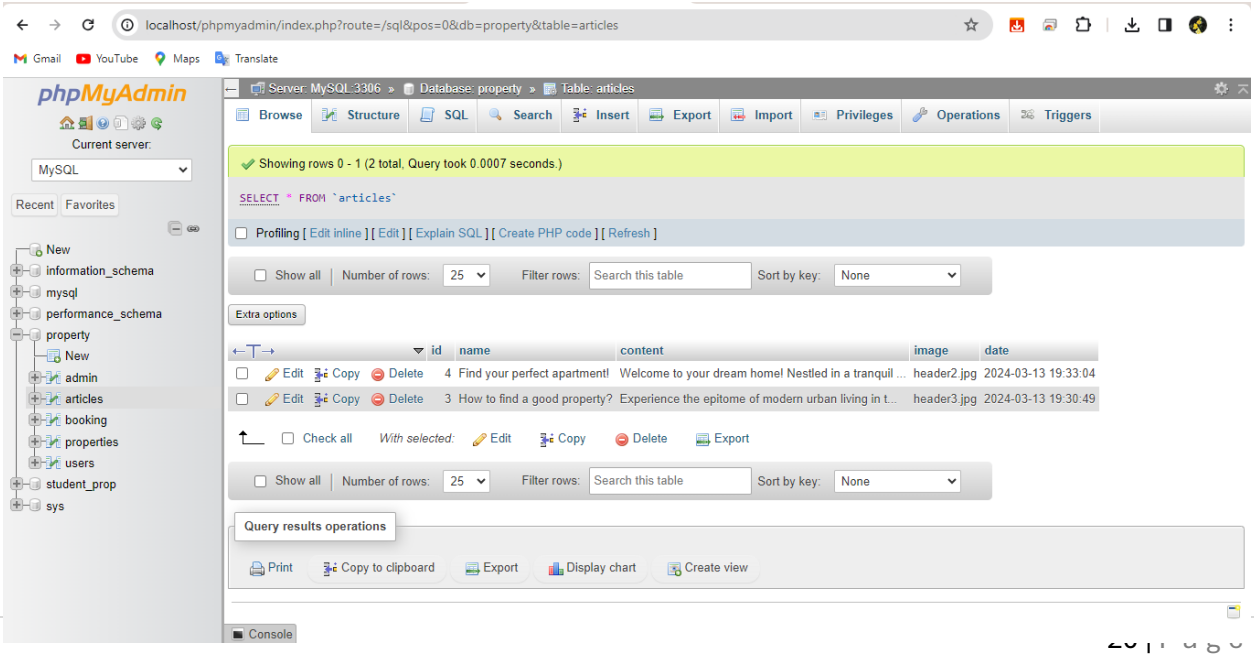

Student Stay Finder
HOME
ABOUT
PROPERTIES
MAP
ARTICLES
CONTACT
LOGIN
Sign up
Map
Satellite
Hostel House
Price: LKR 15000
Square Feet: 2500 Sqft

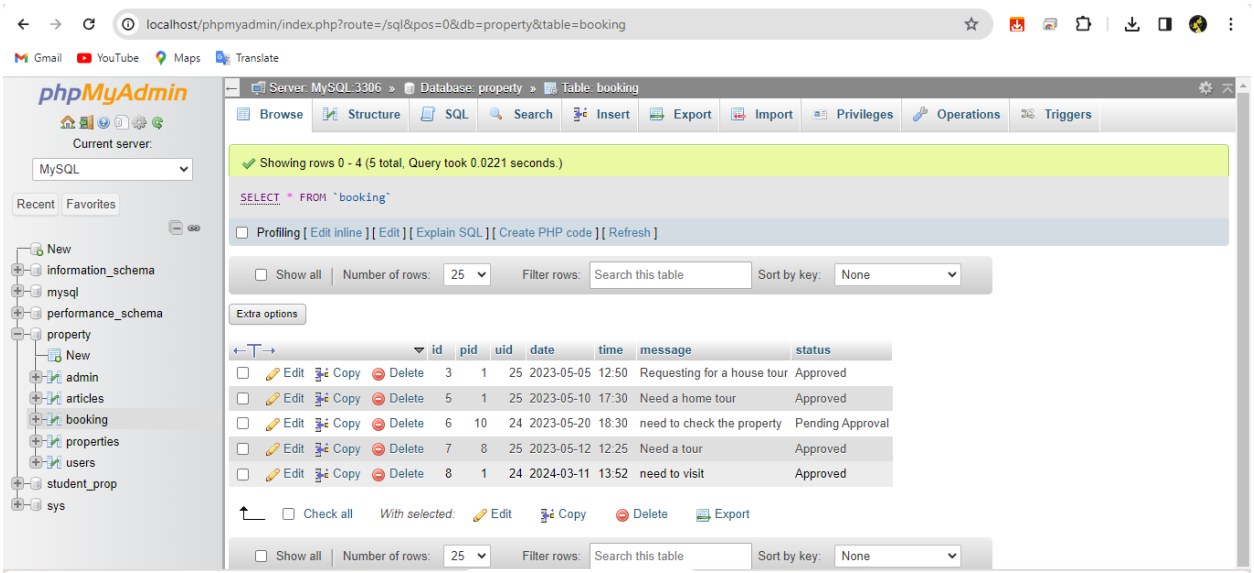
2. Back-end

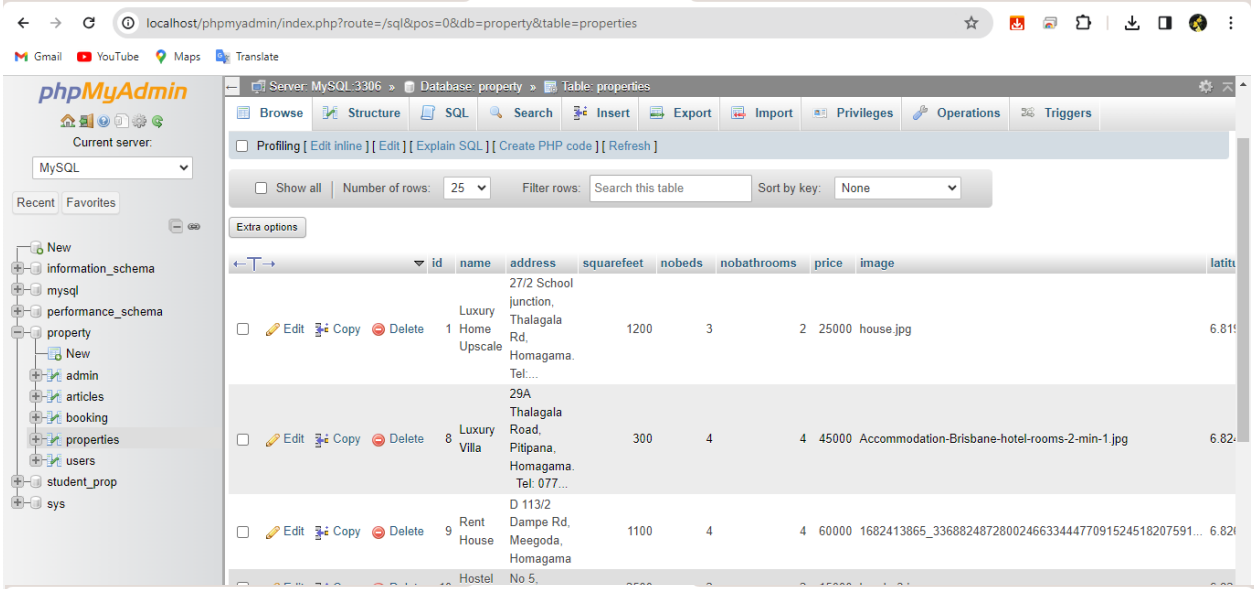
- **Click the Link to refer the backend** :- https://liveplymouthac-my.sharepoint.com/:v:/r/personal/10898536_students_plymouth_ac_uk/Documents/SDTP/2024-04-06%2013-40-08.mkv?csf=1&web=1&nav=eyJyZWZlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAiOiJPbmVEcml2ZUZvckJ1c2luZXNzIiwicmVmZXJyYWxBcHBQbGF0Zm9ybSI6IldlYiIsInJlZmVycmFsTW9kZSI6InZpZXciLCJyZWZlcnJhbFZpZXciOiJNeUZpbGVzTGlua0NvcHkifX0&e=d6zDGA

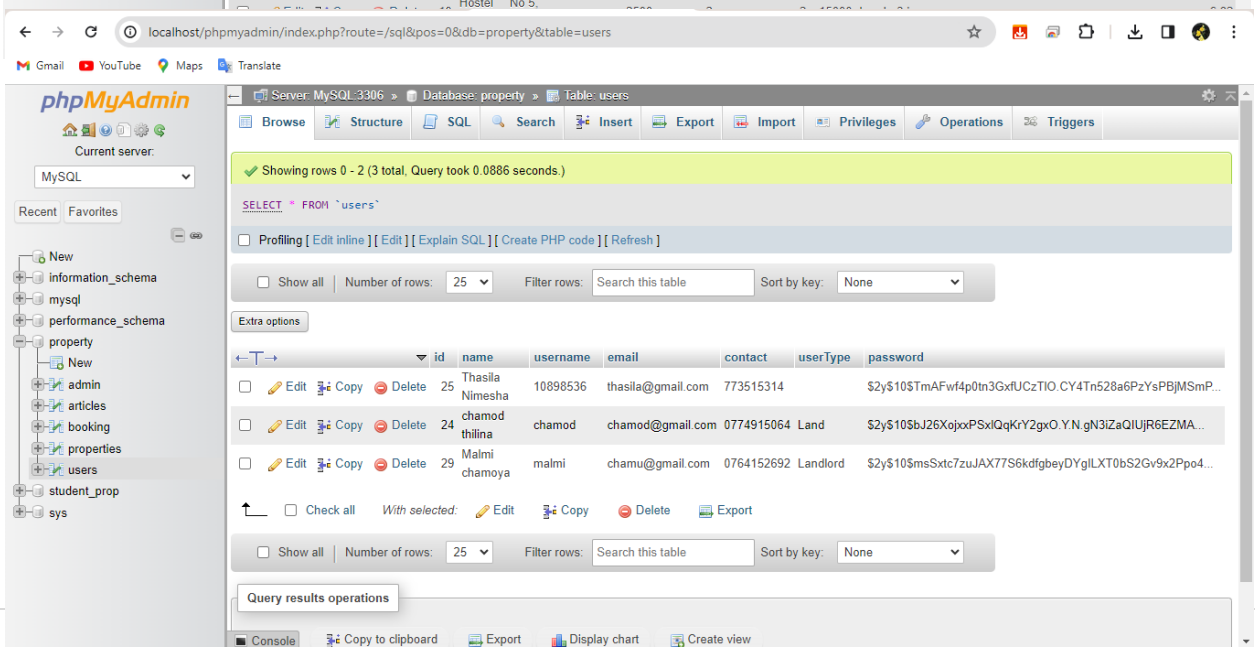
3. Data base






localhost/phpmyadmin/index.php?route=/sql&pos=0&db=property&table=booking
phpMyAdmin
Server: MySQL:3306 » Database: property » Table: booking
Browse Structure SQL Search Insert Export Import Privileges Operations Triggers
Showing rows 0 - 4 (5 total, Query took 0.0221 seconds.)
SELECT * FROM `booking`
id pid uid date time message status
3 1 25 2023-05-05 12:50 Requesting for a house tour Approved
5 1 25 2023-05-10 17:30 Need a home tour Approved
6 10 24 2023-05-20 18:30 need to check the property Pending Approval
7 8 25 2023-05-12 12:25 Need a tour Approved
8 1 24 2024-03-11 13:52 need to visit Approved


localhost/phpmyadmin/index.php?route=/sql&pos=0&db=property&table=properties
phpMyAdmin
Server: MySQL:3306 » Database: property » Table: properties
Browse Structure SQL Search Insert Export Import Privileges Operations Triggers
id name address squarefeet nobeds nobathrooms price image
1 Luxury Home Upscale 27/2 School junction, Thalagala Rd, Homagama. Tel:... 1200 3 2 25000 house.jpg
8 Luxury Villa 29A Thalagala Road, Pitipana, Homagama. Tel: 077... 300 4 4 45000 Accommodation-Brisbane-hotel-rooms-2-min-1.jpg
9 Rent House D 113/2 Dampe Rd, Meegoda, Homagama 1100 4 4 60000 1682413865_336882487280024663344477091524518207591...


localhost/phpmyadmin/index.php?route=/sql&pos=0&db=property&table=users
phpMyAdmin
Server: MySQL:3306 » Database: property » Table: users
Browse Structure SQL Search Insert Export Import Privileges Operations Triggers
Showing rows 0 - 2 (3 total, Query took 0.0886 seconds.)
SELECT * FROM `users`
id name username email contact userType password
25 Thasila Nimesha 10898536 thasila@gmail.com 773515314 $2y$10$TmAFwf4p0tn3GxfUCzTlO.CY4Tn528a6PzYsPBjMSmP...
24 chamod thilina chamod chamod@gmail.com 0774915064 Land $2y$10$bJ26XojxxPSxlQqKrY2gxO.Y.N.gN3iZaQIUjR6EZMA...
29 Malmi chamoya malmi chamu@gmail.com 0764152692 Landlord $2y$10$msSxtc7zuJAX77S6kdfgbeyDYgILXT0bS2Gv9x2Ppo4...
Query results operations
Console Copy to clipboard Export Display chart Create view

Mock Objects Structure and Role

In our development process, we employed mock objects to facilitate testing and simulate interactions with external components or dependencies. Mock objects were utilized primarily during unit testing to isolate the code under test and ensure that it behaves as expected in various scenarios. Below, we outline the structure and role of the mock objects used in our web application

Database Mock

- **Structure:** The database mock is a simulated database interface that mimics the behavior of a real database without actually interacting with the database management system.
- **Role:** During unit testing, the database mock is used to emulate database operations such as data retrieval, insertion, and updates. It allows us to test database-dependent functionalities without relying on a live database, ensuring test independence and repeatability.

Authentication Service Mock

- **Structure:** The authentication service mock is a mock implementation of the authentication service used in the web application.
- **Role:** During unit testing, the authentication service mock simulates user authentication and authorization processes. It allows us to test user authentication-related functionalities, such as login and session management, without relying on a real authentication service.

External API Mock

- **Structure:** The external API mock is a mock implementation of external APIs or services used by the web application.
- **Role:** During unit testing, the external API mock simulates interactions with external APIs or services, such as geolocation services or notification services. It allows us to test functionalities that depend on external services in isolation, without making actual requests to external APIs.

Email Service Mock

- **Structure:** The email service mock is a mock implementation of the email sending service used in the web application.
- **Role:** During unit testing, the email service mock simulates email sending operations. It allows us to test email-related functionalities, such as sending verification emails or notifications, without actually sending emails to real recipients.

File Upload Service Mock

- **Structure:** The file upload service mock is a mock implementation of the file upload service used in the web application.
- **Role:** During unit testing, the file upload service mock simulates file upload operations. It allows us to test functionalities that involve file uploads, such as profile picture uploads or property image uploads, without interacting with a file storage system.

Functional Test Plans

1. Overview

- **Objective:** The objective of functional testing is to validate the functionality of the web application against the defined requirements outlined in the project scenario.
- **Scope:** Functional testing will cover all user roles (landlords, students, and admin) and functionalities described in the project requirements.
- **Approach:** Functional testing will be conducted using both manual and automated testing techniques to ensure comprehensive coverage.

2. Test Environment

- **Environment:** The web application is hosted locally on a WAMP server running on localhost.
- **Hardware:** Standard desktop or laptop computers with internet connectivity for accessing the web application.
- **Software:** Compatible web browsers (Chrome, Firefox, Safari) for testing the web application's compatibility and functionality across different platforms.
- **Tools:** Testing frameworks and tools such as Selenium WebDriver for automated testing, Postman for API testing, and JUnit for unit testing.

3. Test Scenarios

a. Landlord Functionality

- Verify that landlords can register for an account successfully.
- Test property management functionalities, including adding, updating, and deleting properties.
- Validate the handling of reservation requests from students.

b. Student Functionality

- Test the ability of students to browse advertisements, view property details, and reserve accommodations.
- Verify the functionality of booking house tours for selected properties.

c. **Admin Functionality**
    - Test account creation for landlords, students, and admin roles.
    - Validate the posting of articles and administrative tasks such as user management.

4. Test Execution

a. **Manual Testing**
    - Test scenarios will be executed manually by testers to ensure the correctness and usability of the application.
    - Testers will navigate through the web application, perform actions as different user roles, and verify the expected outcomes.

b. **Automated Testing**
    - Automated test scripts will be developed using testing frameworks such as Selenium WebDriver to automate repetitive test scenarios.
    - Automated tests will be executed to validate critical functionalities and ensure regression testing coverage.

5. Test Reporting

- Test results, including test case execution status, defects identified, and any deviations from expected behavior, will be documented in a test report.
- Test reports will be shared with the development team for review and resolution of identified issues.

6. Test Maintenance

- Test cases and scripts will be regularly updated to accommodate changes in the application's functionality or requirements.
- Regression testing will be conducted to ensure that new changes do not introduce defects into existing functionalities.

Critical Analysis of Test Strategy

1. **Coverage and Completeness**
   - **Strengths:** Our test strategy covered a wide range of functionalities, including user roles (landlords, students, admin), property management, reservation handling, and administrative tasks.
   - **Weaknesses:** Despite the extensive coverage, there may be gaps in test scenarios, especially in edge cases or rare user interactions. Further refinement of test scenarios and continuous feedback from stakeholders could enhance coverage.

2. **Automation vs. Manual Testing**
   - **Strengths:** We utilized a combination of automated and manual testing techniques to balance efficiency and thoroughness. Automated tests helped streamline repetitive tasks and ensure consistent validation, while manual testing allowed for exploratory testing and user-centric validation.
   - **Weaknesses:** Limited resources and time constraints may have hindered the extent of automation. Increasing automation coverage and incorporating continuous integration practices could improve efficiency and reduce regression testing efforts.

3. **Environment and Data Management**
   - **Strengths:** We established a test environment that closely mirrored the production environment, ensuring realistic testing conditions. Data management practices, such as using mock objects and fixtures, helped isolate dependencies and maintain test stability.
   - **Weaknesses:** Dependency on a local WAMP server for hosting the application may have introduced inconsistencies in test results compared to a production environment. Exploring cloud-based testing environments or containerization could mitigate environment-related issues.

4. **Test Maintenance and Scalability**
    - **Strengths:** We recognized the importance of test maintenance and scalability by implementing strategies for updating test cases and scripts in response to changes in the application. Regular regression testing helped ensure the stability of existing functionalities.
    - **Weaknesses:** Test maintenance efforts may become increasingly challenging as the application grows in complexity. Establishing robust version control practices and automated test pipelines could streamline test maintenance and ensure scalability.

5. **Feedback Loop and Continuous Improvement**
    - **Strengths:** Our test strategy emphasized collaboration and feedback loops between development, testing, and stakeholders. Regular review meetings and retrospective sessions facilitated continuous improvement and alignment with project goals.
    - **Weaknesses:** While we valued feedback and collaboration, there may have been instances where communication gaps hindered the effectiveness of our testing efforts. Improving communication channels and fostering a culture of transparency could enhance the feedback loop and promote continuous improvement.

Conclusion

In conclusion, our development and testing efforts have culminated in the creation of a robust and functional web application for the NSBM Green University accommodation system. Throughout the project lifecycle, we have adhered to industry best practices, collaborated effectively as a team, and remained committed to delivering a high-quality software product that meets the needs of our stakeholders.

Our development process involved careful consideration of the project requirements, selection of appropriate technologies, and meticulous implementation of key features. By leveraging PHP and MySQL within the WAMP server environment, we were able to create a scalable and efficient solution that facilitates seamless communication between landlords, students, and administrators.

The testing phase played a crucial role in validating the functionality, reliability, and performance of our web application. Through a combination of unit tests, integration tests, and functional tests, we systematically verified the behavior of individual components and the system as a whole. Our testing strategy focused on achieving comprehensive coverage, balancing automation and manual testing, and maintaining a realistic test environment.

While our efforts have resulted in a functional and reliable software product, there are opportunities for further improvement and refinement. Continuous feedback, collaboration, and iteration will be essential in addressing any remaining issues, enhancing the user experience, and ensuring the long-term success of the application.

Looking ahead, we are confident that our web application will serve as a valuable resource for NSBM Green University students, landlords, and administrators, facilitating the search for suitable accommodations and improving overall campus life. We remain committed to supporting and maintaining the application, incorporating feedback from users, and adapting to evolving requirements and technologies.

In conclusion, we are proud of the accomplishments achieved throughout this project and look forward to the positive impact our web application will have on the NSBM Green University community.

References

1. NSBM Green University. (2024). PUSL2020 Course Material. NSBM Green University.

2. Web Development Tutorial. (2024). Building Dynamic Websites with PHP and MySQL. Retrieved from https://www.youtube.com/watch?v=AwsozVSBnu0

3. Software Testing Guide. (2024). Comprehensive Guide to Software Testing. Retrieved from https://www.linkedin.com/pulse/understanding-software-testing-comprehensive-guide-hasan-akdogan-h68xf/

4. PHP Documentation. (2024). PHP Manual. Retrieved from https://www.php.net/manual/en/index.php

5. MySQL Documentation. (2024). MySQL Documentation. Retrieved from https://dev.mysql.com/doc/

-END-