A Project Report

On

“ONLINE BUS MANAGEMNET SYSTEM”

Submitted to the

Department of Computer Applications

In partial fulfilment of the Course

Integrated Master of Computer Applications

Under the guidance of

Dr.ANJANA S CHANDRAN

Project Done by

C A ANUROOP

(Reg no: 193241010113)

DEPARTMENT OF COMPUTER APPLICATIONS

SCMS SCHOOL OF TECHNOLOGY AND MANAGEMENT

December-2023

SCMS SCHOOL OF TECHNOLOGY AND MANAGEMENT

BONAFIDE CERTIFICATE

Certified that the Project Work entitled

"ONLINE BUS MANAGEMNT SYSTEM"

is a bonafide work done by

C A ANUROOP

In partial fulfillment of the requirement for the Award of

INTEGRATED MASTER OF COMPUTER APPLICATIONS

Degree From

Mahatma Gandhi University, Kottayam

(2020-2025)

Head of Department *Project Guide*

Submitted for the Viva-Voce Examination held on…………………………………….

External Examiner1 **External Examiner2**

(Name & Signature) (Name & Signature)

SCMS SCHOOL OF TECHNOLOGY AND MANAGEMENT

CERTIFICATE

This is to certify that the project entitled **“ONLINE BUS MANAGEMNT SYSTEM”** has been successfully carried out by **C A ANUROOP** (Reg no: 203242110111) in partial fulfilment of the Course **INTEGRATED MASTER OF COMPUTER APPLICATIONS.**

Date: **HEAD OF DEPARTMENT**

SCMS SCHOOL OF TECHNOLOGY AND MANAGEMENT

CERTIFICATE

This is to certify that the project entitled **“ONLINE BUS MANAGEMNT SYSTEM”** has been successfully carried out by **C A ANUROOP** (Reg no: 203242110111) in partial fulfilment of the course **INTEGRATED MASTER OF COMPUTER APPLICATIONS** under my guidance.

Date:

Dr. ANJANA S CHANDRAN
INTERNAL GUIDE

SCMS SCHOOL OF TECHNOLOGY AND MANAGEMENT

DECLARATION

I, **C A ANUROOP**, hereby declare that the project work entitled **“ONLINE BUS MANAGEMNET”** is an authenticated work carried out by me under the guidance of **Dr. ANJANA S CHANDRAN** for the partial fulfilment of the course **INTEGRATED MASTER OF COMPUTER APPLICATIONS**. This work has not been submitted for similar purpose anywhere else except to **SCMS SCHOOL OF TECHNOLOGY AND MANAGEMENT**, affiliated to **M.G. UNIVERSITY, KOTTAYAM** .

I understand that detection of any such copying is liable to be punished in any way the school deems fit.

Date:

Place: **C A ANUROOP**

ACKNOWLEDGEMENT

An endeavor over a long period can be successful with the advice, support, and blessings of many well-wishers. To acknowledge all of them is a blissful opportunity showered upon me by the Almighty. With great pleasure and privilege, I present here with full satisfaction, the Project report on "ONLINE BUS MANAGEMNT SYSTEM." I take this opportunity to express my gratitude and sincere thanks to all who helped me complete this project successfully.

I first thank God Almighty, who showered His immense blessings on my effort. I express my gratitude and sincere thanks to the Director, Dr. Indu Nair, for her kind consideration and valuable guidelines throughout our project work. Special thanks to Dr. G Sashi Kumar, our Principal, for his unwavering support and encouragement.

I sincerely express my gratitude to Dr. Anjana S Chandran, HOD, Department of Computer Applications, SCMS School of Technology and Management (SSTM). I extend my sincere gratitude to Dr. Anjana S Chadran, my project guide, for his valuable guidance, cooperation, and constant encouragement throughout the project work.

I also thank my friends and well-wishers who have provided their wholehearted support to me in this exercise. I believe that this endeavor has prepared me for taking up new challenging opportunities in the future.

C A ANUROOP

Table of Contents

1. EXECUTIVE SUMMARY 1

2. BACKGROUND 3

2.1. Existing System 3

2.2. Definition of Problem 3

2.3. Proposed System 3

3. PROJECT OVERVIEW 5

3.1. Objective of the project 5

3.2. Stakeholders 5

3.3. Scope of project 5

3.4. Feasibility Analysis 6

3.4.1 Feasibility study 6

3.4.2 Technical Feasibility 6

3.4.3 Operational Feasibility 6

3.4.4 Schedule Feasibility 7

3.4.5 Economic Feasibility 7

4 OVERALL PROJECT PLANNING 8

4.1. Development Environment 8

4.2. Constraints 8

4.3. Deliverables 9

4.4. Assumptions and Dependencies 9

4.5. Risks 10

4.6. Process Model 10

4.7. Test Strategy 10

4.7.1. System Testing 10

4.7.2. Types of Testing 11

4.8. Testing environment and tools 13

5. ITERATION PLANNING 14

5.1. Schedule.. 14

5.2. Risk 14

6. HIGH LEVEL SYSTEM ANALYSIS 15

6.1. User Characteristics 15

6.2. Summary of system features/Functional requirements 15

6.3. Non Functional Requirements/Supplementary Specifications.. 16

6.4. Glossary 17

6.5. Business Rules 17

6.6. Use Case 17

6.7. Use-case Diagram 18

7. DOMAIN MODEL 19

8. USE CASE MODEL.. 20

8.1. Use case Text 20

8.2. System Sequence Diagram 27

8.3. Operation Contracts 28

8.4. Reports 29

9. DESIGN MODEL... 30

9.1. Sequence Diagram 30

9.2. Class Diagram 31

9.3. UI Design 32

9.4. Theoretical Background 33

9.5. Architecture 34

9.6. Database Design 36

10. TESTING.. 40

10.1 Test cases 40

10.2 Test Report 50

10.3 Sample Code used for testing 50

11. TRANSITION 56

11.1 System Implementation 56

11.2 System Maintenance 57

11.2.1 Corrective maintenance 57

11.2.2 Adaptive maintenance 57

11.2.3 Preventive maintenance 58

12. ANNEXURE 59

12.1 References 59

12.2 Annexure I: User Interview Questionnaires 59

12.3 CONCLUSION 60

12.4 SAMPLE CODE 61

12.4.1 Screenshots 61

12.4.2 Sample Code... 64

1. EXECUTIVE SUMMARY

‘ A Online Bus Booking System ’ is designed to transform the customary way of reserving bus tickets. Passengers can easily browse, book, and manage their bus tickets online with this user-centric approach, doing away with the necessity for in-person trips to bus terminals. A secure and seamless user experience is ensured by the system's relational database, strong backend, and responsive and user-friendly interface.

It is software application that include user registration with secure authentication, an efficient bus search and reservation system, and integration with secure payment gateways for seamless transactions. The platform also includes an admin panel, empowering bus operators to efficiently manage their services, updates schedules and monitor overall performance. The Online bus booking System is a big step in that right direction towards modernising bus ticket reservations increasing productivity, and raising general customer satisfaction in the public transportation industry.

Users of the System

- ADMINISTRATOR
- USERS

Main Functions

Admin Features:

- Manage all users.
- Manage ticket booked by users.
- Approve ticket to ensure authencity.
- Manage trip details
- Manage bus details
- Manage driver details
- Add bus stops and Delete bus stops.
- View tickets submitted by users.
- See an overview of total users, admins, ticket booked,bus details,driver details.

User Features:

- Register and create an account.
- Booking ticket for travel
- Pay for tickets.
- Manage their route.

2. BACKGROUND

2.1. Existing System

Existing system refers to the system that is being followed till now. The existing system requires more computational time, more manual calculations, and the complexity involved in Selection of features is high. The other disadvantages are lack of security of data, Deficiency of Data accuracy, Time consuming etc. To avoid all these limitations and make the working more accurately the system needs to be computerized. Here in the Electronic bus ticketing, a detailed study of existing system is carried along with all the steps in system analysis.

2.2. Definition of Problem

1. Storing the data manually is difficult and time-consuming.
2. Manually stored data may contain errors or invalid data.
3. May affect the business productivity.

2.3. Proposed System

'Online Bus Booking System' is poised to revolutionize property transactions. It addresses the shortcomings of existing systems by delivering an exceptionally user-friendly and responsive web application. Users can effortlessly navigate and search for properties based on search for properties based on their preferences and budget with a single click, ensuring an enhanced and streamlined experience.

The aim of proposed system is to develop a system of improved facilities. The proposed system can overcome all the limitations of the existing system. The system provides proper security and reduces the manual work. The existing system has several disadvantages and many more difficulties to work well. The proposed system tries to eliminate or reduce these difficulties up to some extent. The proposed system will help the user to reduce the workload and mental conflict. The proposed system helps the user to work user friendly and he can easily do his jobs without time lagging.

Advantages of Proposed System:

- **User-Friendly Interface:** Online bus booking system offers an intuitive and user-friendly interface, ensuring easy access and navigation for both admins and users.
- **Enhanced Efficiency and Flexibility:** The platform is designed to be highly efficient and flexible, allowing admins and users to log in and interact seamlessly, enhancing overall user experience.
- **Real-Time Availability and Booking:** Users can conveniently book bus tickets online from the comfort of their homes. 24/7 accessibility allows users to make reservations at any time, reducing the need to visit physical ticket counters.
- **Efficient Seat Management:** The system can efficiently manage seat allocations and reservations, reducing the changes of overbooking or conflicts.
- **Automated Payment Processing:** Integration with secure payment gateways facilitates hassle-free online transactions. Automated payment processing reduces the risk of errors and provides a seamless experience for users.
- **Security Measures:** Online bus booking system prioritizes security and confidentiality, with unique usernames and passwords ensuring that only authorized admins and registered users can access the platform.
- **Integration with Other Services:** Integration with other transportation services or travel-related services can provide users with comprehensive travel solutions.
- **Reduced Environmental Impact:** An efficient booking system can contribute to the optimization of routes, reducing unnecessary fuel consumption and minimizing the environmental impact of transportation.

3. PROJECT OVERVIEW

3.1. Objective of the project

The objective of an Online Bus Booking System is to create a user-friendly and efficient platform that needs of both travellers and bus operators. To provide user with a convenient and accessible platform for booking bus tickets from anywhere with an internet connection, reducing the need for physical visits to bus terminals. Simplify the booking procedure so that customers can look up bus routes, verify seat availability, and make reservations with ease. To implement a secure payment gateway to ensure the safety and confidently of financial transactions, promoting user trust and confidence. Generate electronic tickets automatically upon successful booking, eliminating the need for physical tickets and simplifying the boarding process.

3.2. Stakeholders

- Admin

The admin oversees running the booking system. They log in using a username and password, manage users, oversee ticket booking, approve transactions ,and view user feedback. Admins play a key role in ensuring the smooth operation of Online Bus Booking System.

- Users

Users, booking ticket, log in using their username and password. They can search the time of the buses, manage their travel locations. Users are required to pay for their tickets. Users are at the heart of online bus booking system, driving the booking system on the platform..

3.3. Scope of project

The scope of online bus booking system is defined after the ticket booking, emphasizing seamless payment transactions. The project primarily revolves around assisting users in buying, with the admin overseeing data updates

3.4. Feasibility Analysis

3.4.1 Feasibility study

Every project is feasible for given unlimited resources and infinitive time. Feasibility study is an evaluation of the proposed system regarding its workability, impact on the organization, ability to meet the user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development. Here the resources availability and requirements are said to feasible to create the proposed system.

3.4.2 Technical Feasibility

Technical feasibility assesses whether the current technical resources are sufficient for the new system. If they are not available, can they be upgraded to provide the level of technology necessary for the new system? It checks whether the proposed system can be implemented in the present system without supporting the existing hardware.

Currently,

- Technology exists to develop a system.
- The proposed system can hold data to be used.
- The proposed system can provide adequate response. Hence, we can say that the proposed system is technically feasible.

3.4.3 Operational Feasibility

Operational feasibility determines if the human resources are available to operate the system once it has been installed. The resources that are required to implement or install are already available with the Breakdown Assist. The persons of this Assist need no exposure to computer but have to be trained to use this particular software.
The project is optimally feasible.

3.4.4 Schedule Feasibility

An evaluation of the time needed for the development of this project. The time schedule required for the development of this project is very important, since more development time effects machine time, costs and delays in the development of the other systems. So the project should be complete within affixed schedule time as far as this is concerned.

Schedule feasibility study for the design is shown below

Problem identification	5

Requirement analysis	10
Overall design	20
Construction	22
Testing	15

3.4.5 Economic Feasibility

Economic feasibility determines whether the time and money are available to develop the system. It also includes the purchase of new equipment, hardware and software. Since software product must be cost effective in the development, on maintenance and in the use. It is affordable to allocate the required resources.

4. OVERALL PROJECT PLANNING

4.1. Development Environment

Hardware Specifications

- Intel i3 or above
- Memory: at least 4GB
- Display: Color monitor
- Keyboard: Windows Compatible
- Mouse: Windows Compatible

Software Specifications

Technology used:

i. Server side

- Front end : PHP, HTML, CSS, Bootstrap
- IDE : Visual Studio Code
- Back end : SQL server

- Operating System : Windows

4.2. Constraints

- The set of constraints that we come across this system is as follows
- User Interface is only in English i.e.no other language option is available.
- Admin can login with his assigned username and password i.e. no guest facility is available.

4.3. Deliverables

List of documents that shall be delivered are User Manual

- System maintenance documentation.
- Application archive with source code.
- Database backup and DDL script.
- Complete source code.

4.4. Assumptions and Dependencies

a) Assumptions

- All roles are created in the system already but further registration of users on given roles can be done.
- Roles and tasks are predefined and are made known to the administrator.
- The code should be free of compilation errors/syntax errors.
- The product must have an interface which is simple enough to understand.
- Roles and tasks are predefined and are made known to the administrator.
- End users should have basic knowledge of computer.

b) Dependencies

- All necessary hardware and software are available for implementing and use of the tool.
- All roles are created in the system already.
- The proposed system should be designed, developed and implemented based on the software requirements specifications document.

4.5. Risks

Some of the risks are follows

- Database crash will cause heavy data loss
- Wrong input will cause discrepancies in data
- Availability of the network.

4.6. Process Model

The process model for developing the project is agile model.

The phases are: -

- Requirement analysis
- System study
- Designing
- Coding
- Testing
- Maintenances

4.7. Test Strategy

4.7.1. System Testing

When a system is developed, it is hoped that it performs properly. In practice however some errors always occur. The main purpose of testing and information system is to find the errors and correct them. A successful test is one which finds an error. The main objectives of system testing are:

- To ensure during operation the system will perform as per specifications.
- To make sure that the system meets the requirements during operation.
- To verify that the controls incorporated in the system function as intended.
- To see that when correct inputs are fed to the system the outputs are correct.
- To make sure that during operation incorrect input and output will be deleted.

The scope of a system test should include both manual operations and computerized. Operation system testing is a comprehensive evaluation of the programs, manual procedures, computer operations and controls. System testing is the process of checking if the developed system is working according to the original objectives and requirements. All testing needs to be conducted in accordance with the test conditions specified earlier.

4.7.2 Types of testing

Unit Testing

Unit Testing will be done to test field validations, navigation, functionality of the programs and its block. These tests are applied on various functions within each program and other critical program blocks.

Module Testing

Module Testing will be each program done to test the interaction between the various programs within one module. It checks the functionality of with relation to other programs within the same module. It then tests the overall functionality of each module.

Integration Testing

The major concerns of integration testing are developing an incremental strategy that will limit the complexity of entire actions among components as they are added to the system.

Developing a component as they are added to the system, developing an implementation and integration schedules that will make the modules available when needed, and designing test cases that will demonstrate the viability of the evolving system. Though each program works individually they should work after linking them together. This is also referred to as interfacing. Data may be lost across interface and one module can have adverse effect on another. Subroutines after linking may not do the desired function expected by the main routine. Integration testing is a systematic technique for constructing program structure while at the same time conducting tests to uncover errors associated with the interface. In the testing, the programs are constructed and tested in small segments.

Validation Testing

This provides the final assurance that the software meets all the functional, behavioral and performance requirements. The software is completely assembled as a package. Validation succeeds when the software functions in a manner in which user wishes. Validation refers to the process of using software in live environment in order to find errors. During the course of validation, the system failure may occur and sometime the coding has to be hanged according to the requirement. Thus the feedback from the validation phase generally produces changes in the software. Once the application was made of all logical and interface errors, inputting dummy data ensure that the software developed satisfied all the requirements of the user. The dummy data is known as test cases.

Output Testing

After performing the validation testing, the next step is output testing of the proposed system since no system could be useful if it does not produce the required output in the specific format. Asking the users about the format of output they required, tests the output generated in two ways. One is on screen and another is printed format. The output format on the screen found to be correct as the format was designed in the system design phase according to the user needs. For the hard copy also, the output comes out as the specified requirement by the user. Hence output testing does not result in any correction in the system.

Acceptance Testing

Acceptance testing (also known as user acceptance testing) is a type of testing carried out in order to verify if the product is developed as per the standards and specified criteria and meets all the requirements specified by customer. This type of testing is generally carried out by a user/customer where the product is developed externally by another party. Acceptance testing falls under black box testing methodology where the user is not very much interested in internal working/coding of the system, but evaluates the overall functioning of the system and compares it with the requirements specified by them. User acceptance testing is considered to be one of the most important testing by user before the system is finally delivered or handled over to the end user. Acceptance testing is also known as validation testing, final testing, QA testing, factory acceptance testing and application testing etc. And in software engineering, acceptance testing may be carried out at two different levels; one at the system provider level and another at the end user level (hence called user acceptance testing, field acceptance testing or end-user testing). Acceptance test refers to the acceptance of data into the system for processing. The acceptance test contributes to the consistency and smooth working of the system. The system under consideration is tested for users at a time for developing and making changes whenever required.

4.8. Testing environment and tools

The hardware specification used for testing:

Operating system	Windows 11
Memory	8 GB
Hard Disk	512 GB

The software specification used for testing:

Front End	PHP, HTML, CSS, Bootstrap
Back End	WAMPP
Operating System	Windows 10

5. ITERATION PLANNING

5.1. Schedule

SERIAL NO.	TASK	DURATION
1	Problem identification	5 days
2	Requirement Specification	10 days
3	Database Design and Analysis	12 days
4	Design Analysis	9 days
5	Coding	24 days
6	Testing	10 days
	Total	70 days

5.2. Risk

- Wrong input
- Software installation issues.
- Database crash will cause heavy data loss

6. HIGH LEVEL SYSTEM ANALYSIS

6.1. User Characteristics

All users of the system are expected to have basic knowledge of using a computer and basic knowledge in English language.

Users of the system:

- Admin
- Users

6.2. Summary of system features/Functional requirements

Manage Users

Admin can view, and delete user accounts.

Manage Properties

Admin can add, view, and delete user accounts

Ticket Approval

Admin has the authority to approve properties for authenticity.

Manage Tickets

Admin can add, update, view, and delete.

Manage Drivers

Admin can generate various reports based on specified criteria.

Manage Bus Details

Admin can add, view, update and delete the bus details.

Account Registration

Users can register and create their accounts.

Subscription Payment

Users can pay for subscriptions to post properties.

Ticket Management

Users can manage the ticket they have booked.

Send Messages

Users can send messages, suggestions, or feedback to the admin.

Enquiries

Users can send inquiries to property owners and view received inquiries.

Profile Management

Users can view, edit, and manage their profiles. The users can also change password and phone number visibility settings.

6.3. Non Functional Requirements/Supplementary Specifications

The non-functional requirements which define the system performance are:

Accuracy:

The level of accuracy in the proposed system will be high. All operations would be done correctly and it ensures that whatever information that comes from the center is accurate.

Reliability:

The reliability of the proposed system will be high. The reason for the increased reliability of the system is that system uses correct formulas to calculate the results.

Immediate Response:

The system is highly responsive because it uses well accurate formulas to calculate required results provided the user should enter the valid input data.

Easy to Operate:

The system should be easy to operate and should be such that it can be developed within a short period of time and fit in the limited budget of the user.

The other non-functional requirements are:

- Security
- Maintainability
- Extensibility
- Reusability
- Resource utilizations

6.4. Glossary

Admin	Administrator
Users	Property Seekers and Sellers

6.5. Business Rules

The information should be correct and valid.

6.6. Use Case

<u>Registration for new user</u>

The User can register new account and login.

Services

Users can search for buses, view bus details, book tickets and manage bus schedules.

Feedback

Users can view, update, and delete their listings, as well as check the approval status of their posted properties.

My Tickets

Customers can book tickets for a selected bus at selected time.

Contact

Customers can contact the admins for enquiries

Manage Bus

Admin can manage the bus details, routes, drivers and add, delete, view, update the details

Manage Route

Admin can add, delete, view, update the routes.

View Bookings

Admin can view, add, delete, update and confirm the bookings

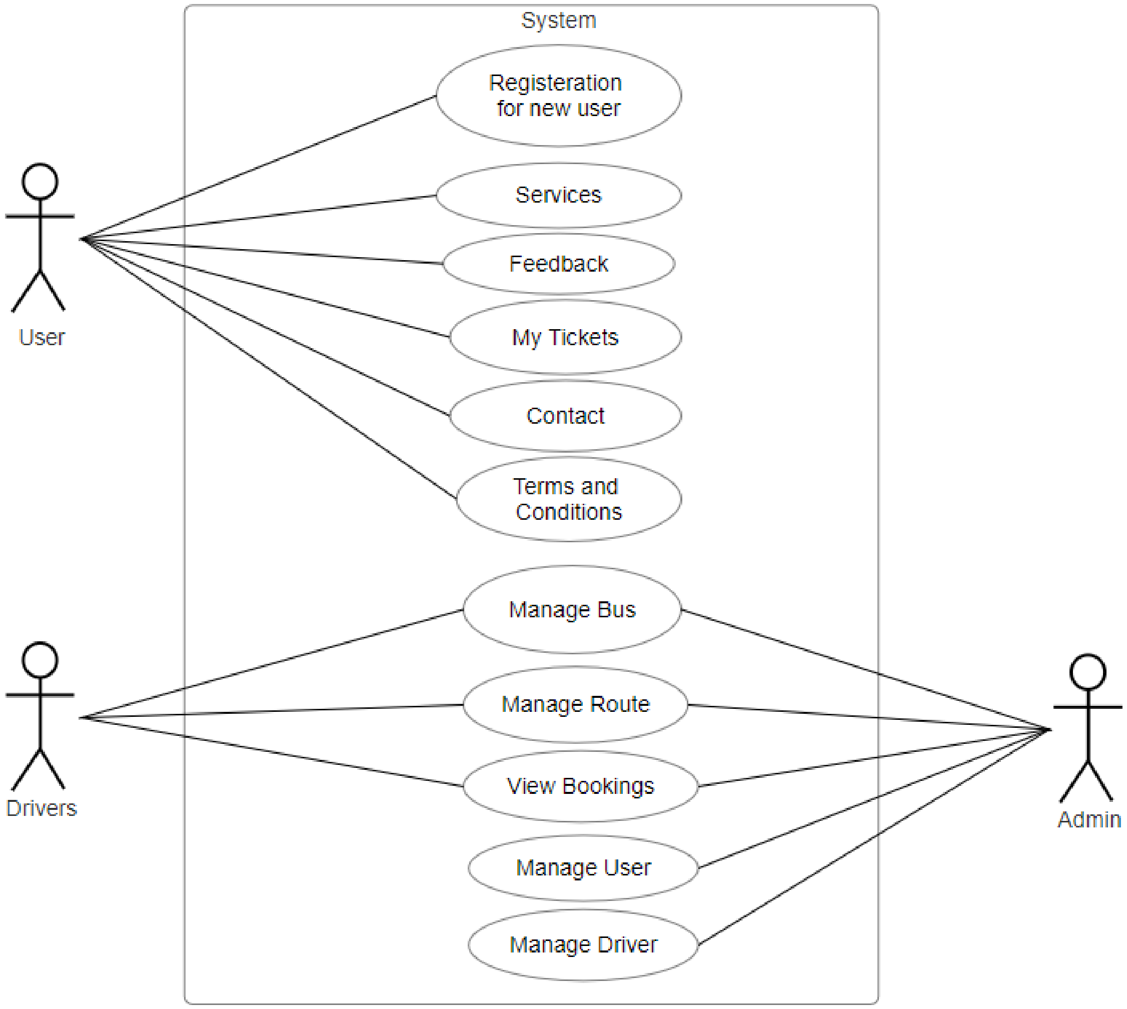
Manage User

Admins can view, search, and delete user.

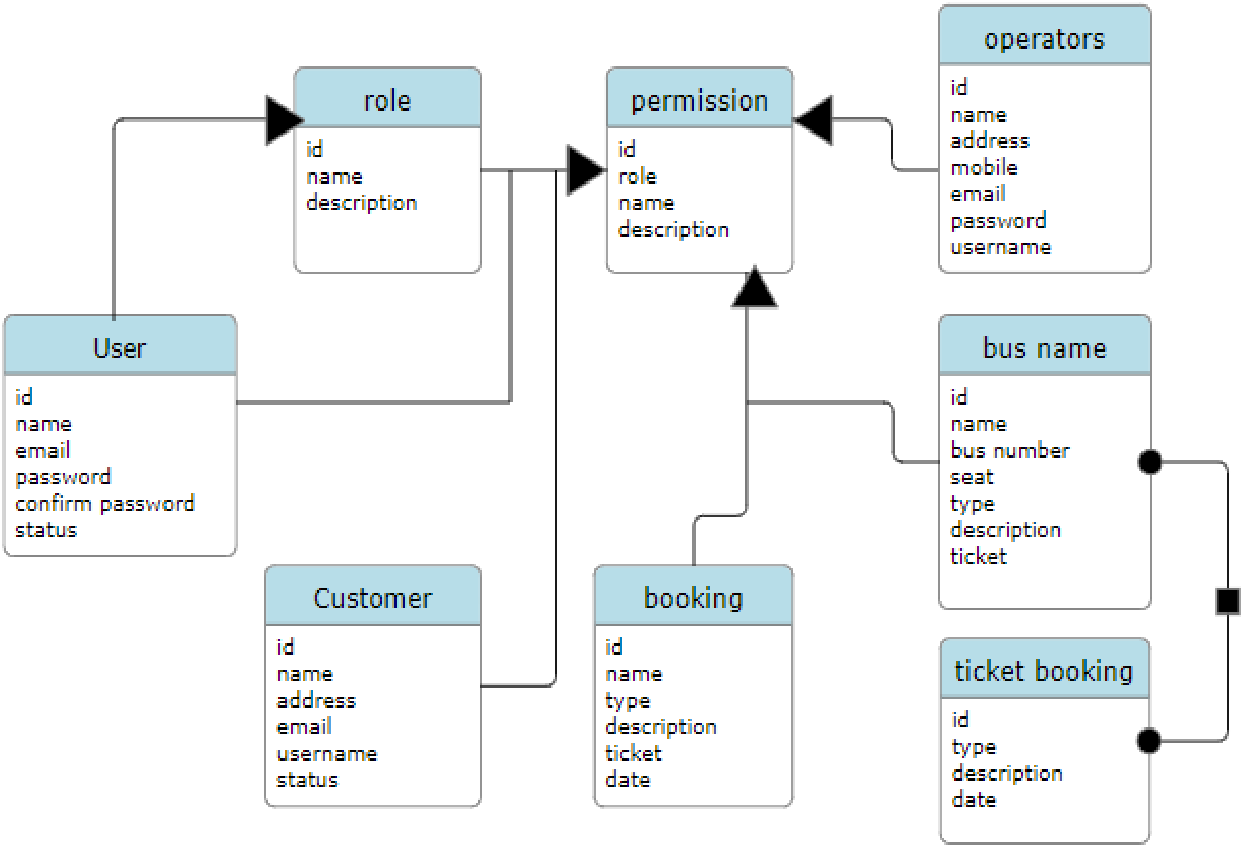
Manage Drivers

Admins can view, add, delete, update the drivers.

6.7. Use-Case Diagram



7. DOMAIN MODEL



8. USE CASE MODEL

8.1 Use case text

Scope: Real Estate Management

Primary Actor: Admin

Stakeholders and Interests:

- **Users**: Users can be the one who posts properties or a person who wishes to acquire a property posted by another user. Users can log in to the system, search and book ticket. They can manage their tickets, and view enquiry requests for the properties they posted. They will also have to pay a subscription fee of 1000rs for posting an unlimited number of properties for 5 months if they have already posted a property in the month. They can also send an enquiry request for a property posted by another user. They can also send their messages to the admin.
- **Administrator:** Admin is responsible for operating the whole system. The admin gets logged in by a valid username and password. Admin can add and manage other Admins, Manage and Approve listings, View Messages from users as well as generate reports.

Preconditions:

No user can use the system without logging into the system.

Success Guarantee (Post conditions):

Really easy to use the system.

Main Success Scenario:

Use Case: Admin - Manage & Approve Listings

1. System redirects to admin's listing page.
2. The admin views all the listings and their details.
3. Admin clicks on 'View Property' button for viewing property.
    a. The system redirects to a new page where the admin can view the property along with the description, images, and videos.

b. The admin can also delete the property from the same page

i. System validates the property information and if there's no issue, the data is deleted from the database.

4. Admin clicks on the 'Delete Listing' button.

a. System prompts for confirmation and validates the property information and if there's no issue, the data is deleted from the database.

5. Admin clicks on the 'Approve Listing' button.

a. System prompts for confirmation and validates the property information and if there's no issue, the data is updated in the database making the property status as approved.

6. Admin types in the search query and clicks on the search button.

a. System validates the entered information and if there exists such a property, the data is retrieved from the database.

Use Case: Admin - Manage Admins

1. System redirects to admin's page where they can manage administrators.
2. The admin views all the admins and their details.
3. Admin clicks on the 'Update Profile' button to update their profile.
    a. The system redirects to a new page where the admin can enter their new account details and update their account.
        i. The system validates the account information and if there's no issue, the data is updated in the database.
4. Admin clicks on the 'Register New' button to add a new admin to access the system.
    a. The system redirects to a new page where the admin can enter the details of the new admin account.
        i. The system validates the account information and if there's no issue, the data is added to the database.
5. Admin clicks on the 'Delete Admin' button.
    a. System prompts for confirmation and validates the admin information and if there's no issue, the data is deleted from the database.

6. Admin types in the search query and clicks on the search button.

a. System validates the entered information and if there exists such a admin, the data is retrieved from the database.

Use Case: Admin - Manage Users

1. System redirects to admin's page where they can manage users.
2. The admin views all the users and their details.
3. Admin clicks on the 'Delete User' button.
    a. System prompts for confirmation and validates the information and if there's no issue, the data is deleted from the database.
4. Admin types in the search query and clicks on the search button.
    a. System validates the entered information and if there exists such a user, the data is retrieved from the database.

Use Case: Admin – Manage Messages

1. System redirects to admin's page where they can manage messages.
2. The admin views all the messages from the users and their details.
3. Admin clicks on the 'Delete Message' button.
    a. System prompts for confirmation and validates the information and if there's no issue, the data is deleted from the database.
4. Admin types in the search query and clicks on the search button.
    a. System validates the entered information and if there exists such a message, the data is retrieved from the database.

Use Case: User- Post Property

1. User clicks the 'Post Property' button.
2. If the user has already posted a property in the same month, the system redirects to the payment page to pay the subscription fee.
    a. The system prompts the user to enter their payment details including card number and cvv.
        i. System validates the entered information and if there's no issue, the payment details are entered into the database.

3. The system redirects to the post property page.
    a. System prompts the user for entering details of the property, upload the images and videos of the property.

b. Users enters the property details and clicks ‘Post Property’ button.

i. System validates the entered information and if there’s no issue, the details are inserted in the database and shows a success message.

Use Case: User- Manage Posted Property

1. System redirects to user’s ‘My Listings page’.
2. The user views all the listings posted by them and their details.
3. User clicks on the ‘Update Property’ button to update their property.
   a. The system redirects to a new page where the user can update their property and change their descriptions, images and videos.
      i. The system validates the property information and if there’s no issue, the data is updated in the database.
4. User clicks on the ‘Delete Property’ button.
   a. System prompts for confirmation and validates the property information and if there’s no issue, the data is deleted from the database.
5. User clicks on ‘View Enquiry’ button for viewing property requests.
   a. The system redirects to a new page where the user can view all the enquiry requests from users for that property.
   b. The user can click on ‘Delete Request’ button for deleting the request.
      i. System prompts for confirmation and validates the property information and if there’s no issue, the data is deleted from the database.
   c. The user can click on ‘View Property button’ to view the property for which the request was received.
      i. The system redirects to a new page where the user can view the property along with the description, images, and videos.
6. User clicks on ‘View Property’ button for viewing property.
   a. The system redirects to a new page where the user can view the property along with the description, images, and videos.
   b. The user can also view enquiries from the same page by clicking ‘View Enquiries’ button.
      i. Goto step 5

Use Case: User – Send Property Enquiry

1. The user clicks on the ‘All listings’ page and the system redirects to a new page where they can view all the approved properties and its details.
2. User clicks on the ‘Send Enquiry’ button.
    a. System prompts for confirmation and validates the information and if there’s no issue, the request is sent to the user who posted the property.

Use Case: User – Manage All Enquiry Requests

1. The user clicks on the ‘My Requests’ option and the system redirects to a new page.
2. The user views all the requests from other users and their details.
3. The user can click on ‘Delete Request’ button for deleting the request.
    a) The system prompts for confirmation and validates the property information and if there’s no issue, the data is deleted from the database.
4. The user can click on ‘View Property button’ to view the property for which the request was received.
    a) The system redirects to a new page where the user can view the property along with the description, images, and videos.

Use Case: User – Send Messages

1. The user clicks on the ‘Contact Us’ page and the system redirects to a new page where they can view a contact us form.
2. The user fills in all the fields with their personal details along with the message they wish to send to the admin.
3. User clicks on the ‘Send Message’ button.
    a. System prompts for confirmation and validates the information and if there’s no issue, the request is sent to the user who posted the property.

Use Case: Admin- Generate Reports

1. System redirects to admin’s generate reports page.
2. The admin click on the button appropriate for generating the desired report.
3. Based on the selected button, the system redirects to the appropriate reports page.
4. Admin can generate various reports based on different criteria.

Extensions:

Admin - Manage Admins

3. a. The new password and password retyped for confirmation isn't same.

1. System tells the Admin that the passwords entered doesn't match.

2. The use case continues at step 3.

4. a. The new admin information entered doesn't contain all the necessary details.

1. System tells the Admin that some information is missing.

2. The use case continues at step 4.

4. b. The entered admin information is in incorrect format.

1. System tells the admin that some information is incorrect.

2. The use case continues at step 4.

User- Post Property

2. a. The entered payment information doesn't contain all the necessary details.

1. System tells the User that some information is missing.

2. The use case continues at step 2.

2. b. The entered payment information is incorrect.

1. System tells the User that some information is incorrect.

2. The use case continues at step 2.

3. a. The entered property information doesn't contain all the necessary details.

1. System tells the User that some information is missing.

2. The use case continues at step 3.

3. b. The entered property information is in incorrect format.

1. System tells the User that some information is incorrect.

2. The use case continues at step 3.

User- Manage Posted Properties

3. The entered property information is in incorrect format.

1. System tells the user that some information is incorrect.

2. The use case continues at step 3.

User- Send Messages

3. a. The entered message information doesn't contain all the necessary details.

1. System tells the User that some information is missing.

2. The use case continues at step 2.

3. b. The entered message information is in incorrect format.

1. System tells the User that some information is incorrect.

2. The use case continues at step 2.

Special Requirements:

1. Text must be visible from 1 meter.
2. We want robust recovery when the system fails.
3. Language internationalization on the text displayed.

Frequency of Occurrence:

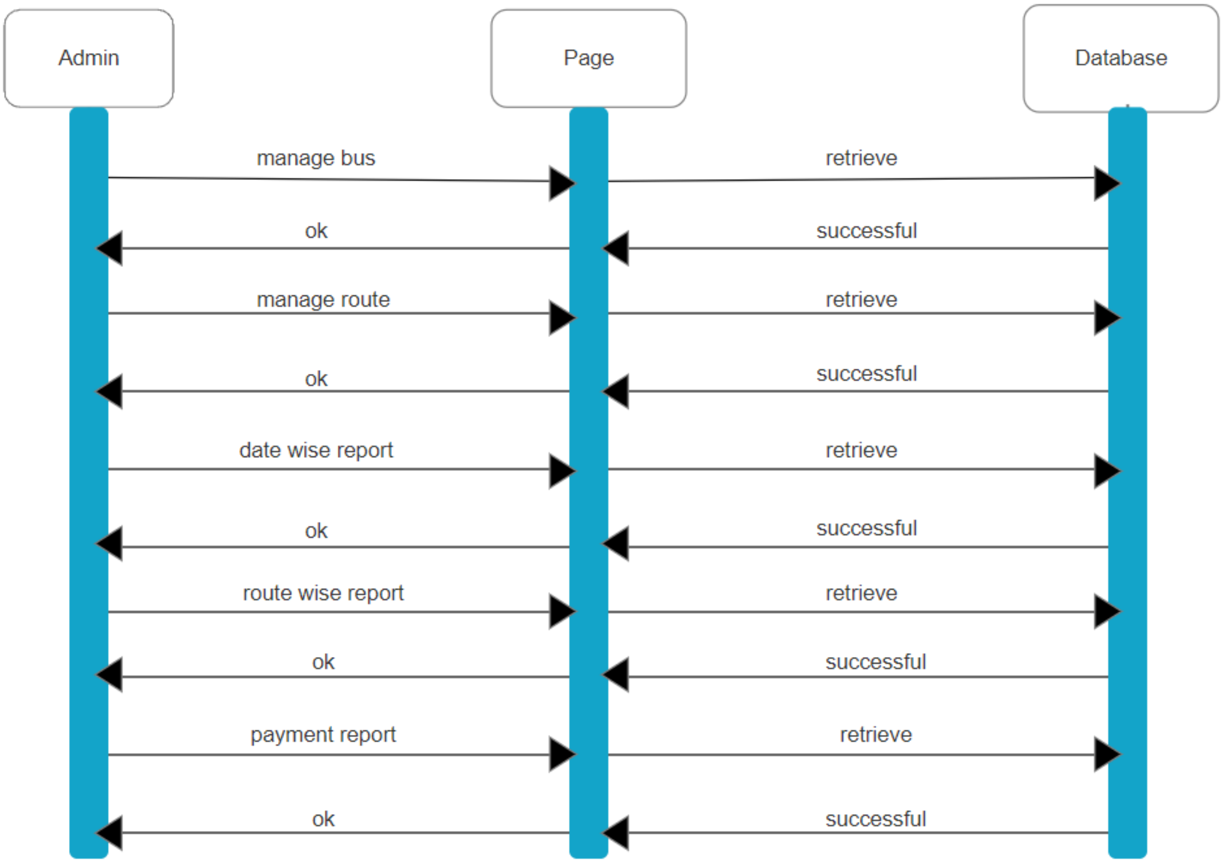
Could be nearly continuous.

Open Issues:

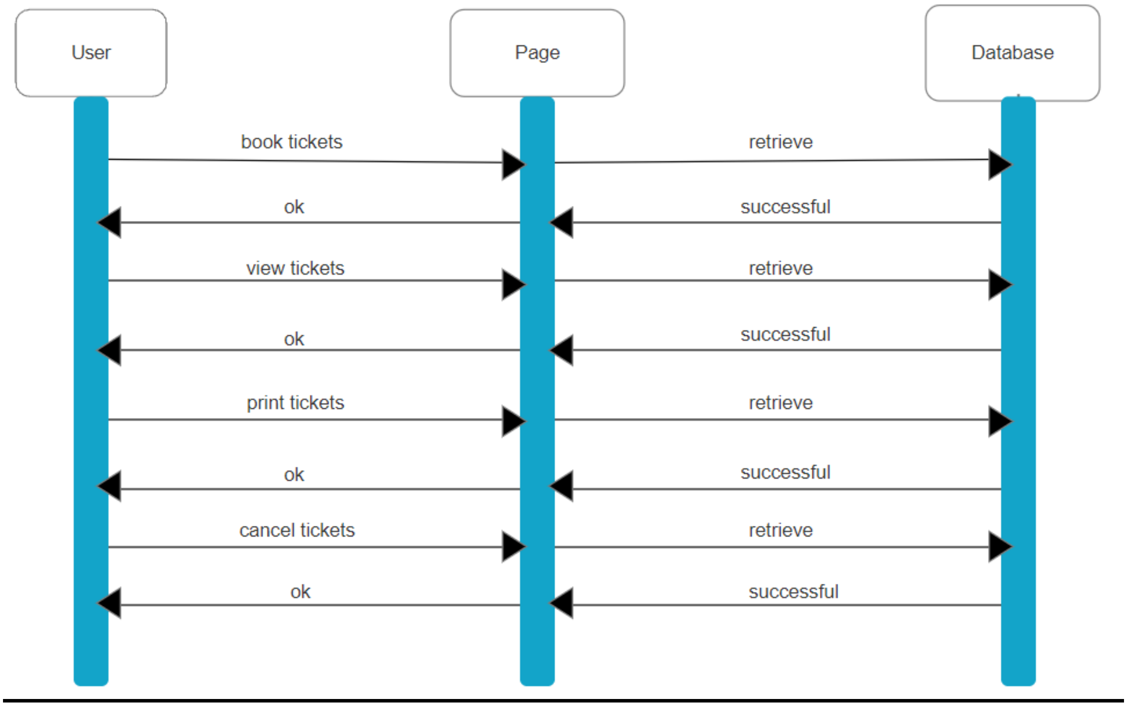
1. Explore the recovery issues.

8.2 System Sequence Diagram

Admin Side


Admin
Page
Database
manage bus
retrieve
ok
successful
manage route
retrieve
ok
successful
date wise report
retrieve
ok
successful
route wise report
retrieve
ok
successful
payment report
retrieve
ok
successful

User Side



8.3 Operation Contracts

Operation: Registered User (username: string, password: string)

Cross Reference: Use case: User login

Preconditions: Proper communication between pages

Conditions :

- A new login instance was created was created by the admin/user.
- Accepted username and password and stored it in respective attributes.
- After validation, the user gets logged in to the system.

Operation: Admin (username: string, password: string)

Cross Reference: Use case: Admin login

Preconditions: Proper communication between pages

Conditions :

- An admin login instance was defined.
- Accepted username and password and stored it in respective attributes.
- After validation, the admin gets logged in to the system.

8.4 Reports

User Payment Report: Admin can generate reports based on user payment done by the users between the selected date range.

Bus Details Report: Admin can generate reports based on the bus details ,if it is sleeper or semi-sleeper and based on the count of the seats

Bus Price Report: Admin can generate reports based on properties whose pricing is between the entered budget range.

Place Report: Admin can generate reports based on place that is already entered.

Bus Time Report: Admin can view data reports based on properties that are posted between the selected date range.

Monthly Payment Report: Admin can generate reports based on payment done by the users between the selected month and year.

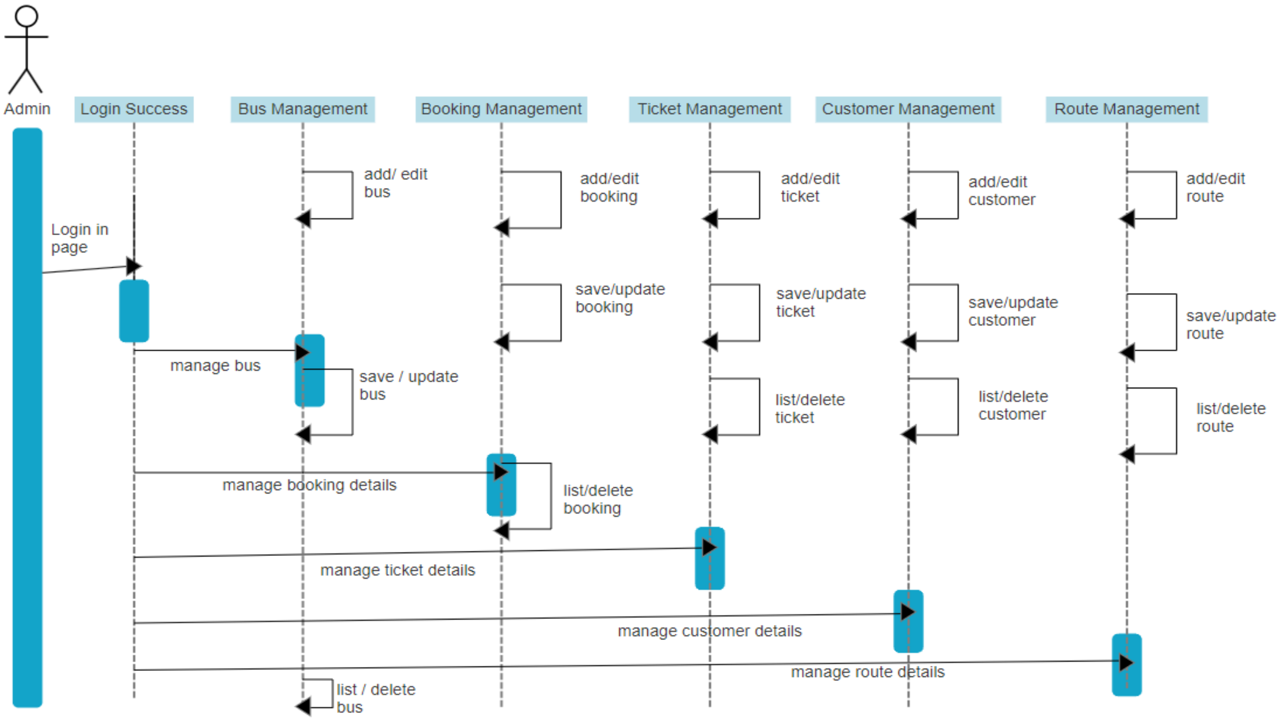
Yearly Payment Report: Admin can generate reports based on payment done by the users between the selected year.

Bus Type Visualization Report: Admin can generate a bar chart report based on bus type – sleeper, semi-sleeper.

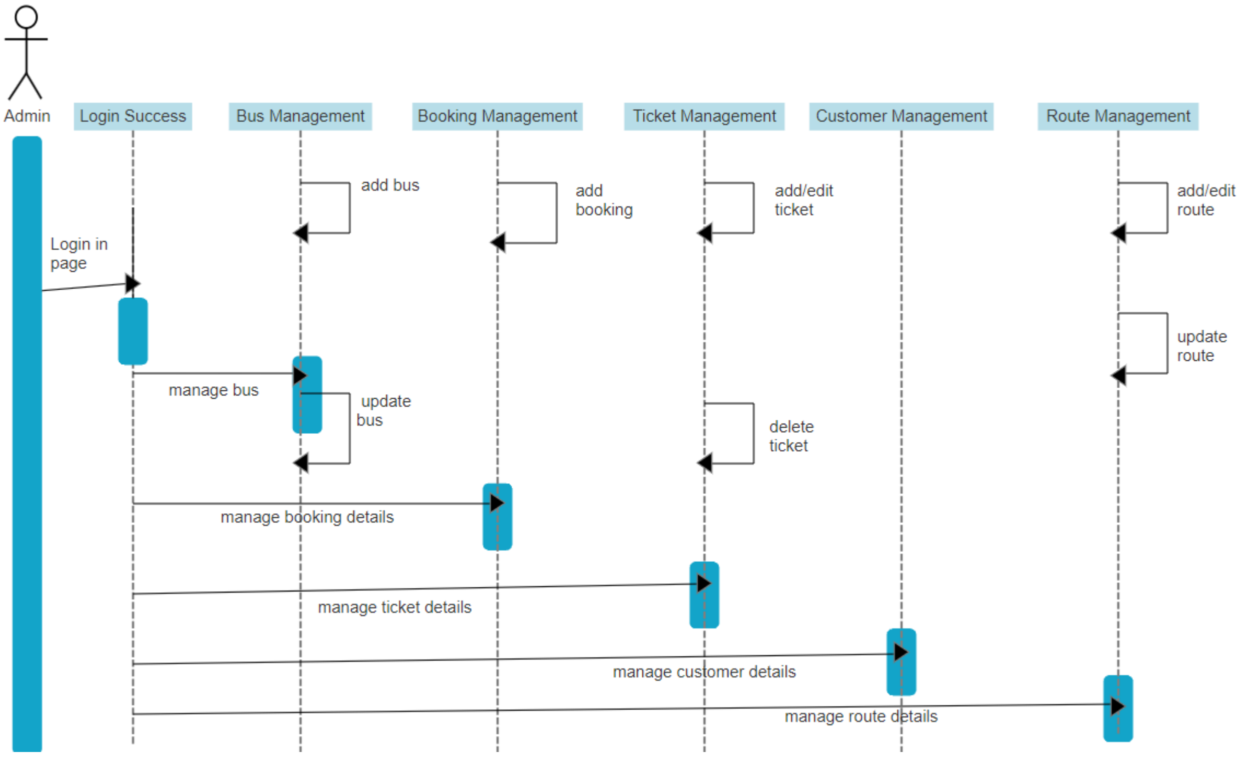
9. DESIGN MODEL

9.1 Sequence Diagram

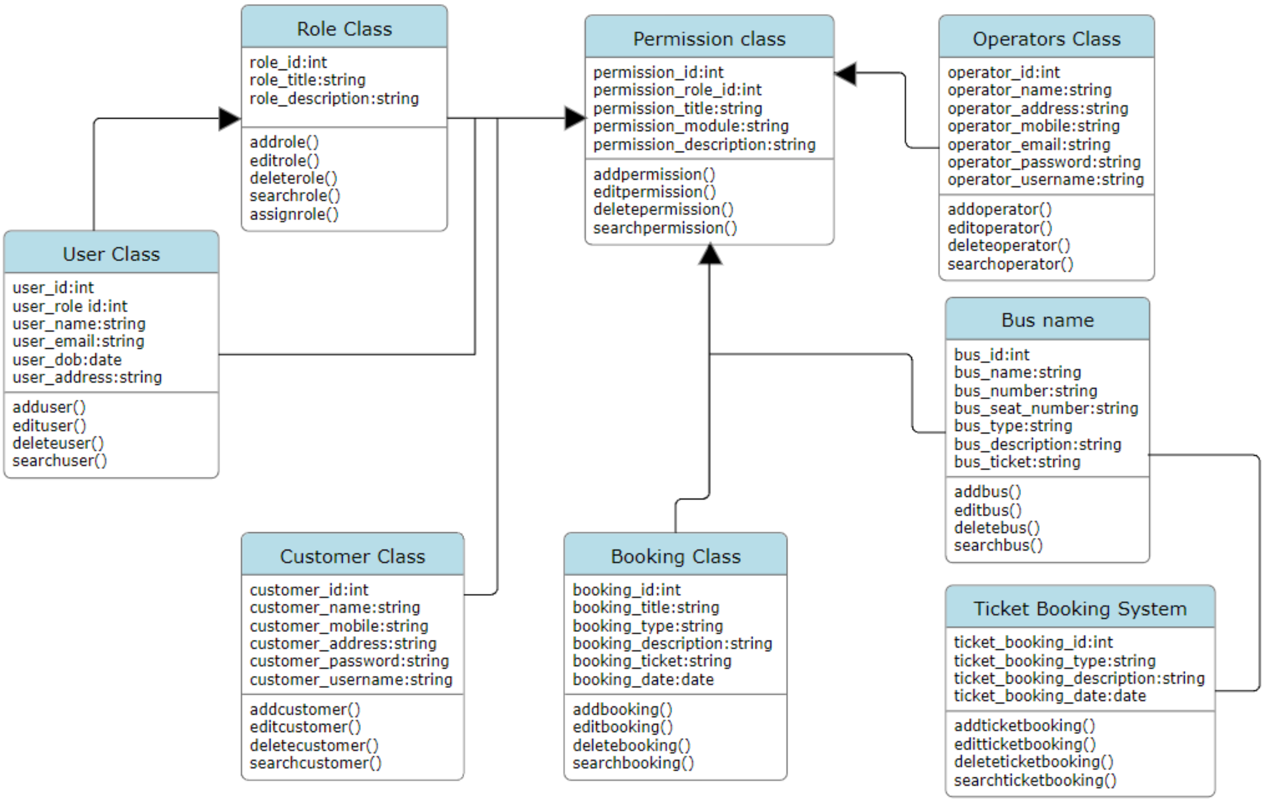
Admin



Users



9.2 Class Diagram



9.3 UI Design

ADMIN:

Login Page: The admin must enter a username and a password to login to the application.

Home Page: Displays the total count of users, admins, properties posted, and new messages.

Listings Page: Admin can view all the properties posted by the users and will have the options to delete and approve them.

Approval Waiting Page: Admin can view all the unapproved properties and approve them.

Users Page: Admin can view/search or delete users.

Admins Page: Admin can add/view/search or delete admins. They also have the option to update their profile and change the password.

Messages Page: Admin can view/search or delete Messages sent by users to admin.

Reports Page: Admin can generate various reports based on Subscriber Payment, Property Owner, Property Pricing, Property Type, Monthly and Yearly Payment as well as a Visualization on Property Types.

Logout Page: By clicking this button, the admin gets logged out of the application.

USER:

Login Page: The user must enter a username and a password to login to the application.

Home Page: The user can view all the existing properties and also allows them to search for property based on different criterias.

Post Property Page: Enables the user to post a property and share property details, including descriptions, images, videos, and 360-degree views. The user is redirected to pay for a subscription before posting a property if the user has already posted a property in the same month.

My Listings Page: The user can view, update or delete all the properties they have posted.

My Requests Page: The user can view or delete all the enquiry requests for the properties they have posted.

All Listings Page: It allows the user to view the properties posted by other users. It also allows the user to send an enquiry to the user who posted it.

Dashboard Page: It gives the user an overview on the count of the Properties Listed and Saved, Requests Sent and Received by them.

Contact Us: It allows the user to send messages and their valuable suggestion to the administrator personally.

Logout Page: By clicking this button, the user gets logged out of the application.

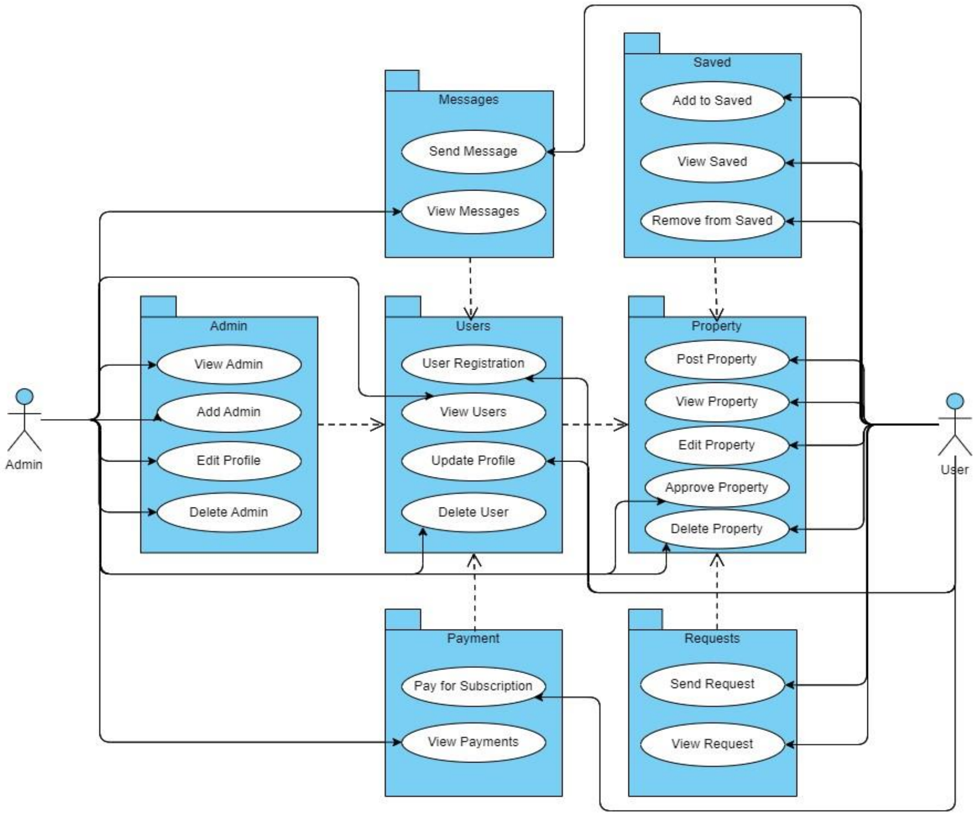
9.4 Theoretical Background

Online Bus Booking System – The Real Estate Portal is developed using one of the widely used front-end tools PHP, HTML, CSS and Bootstrap and at the back-end, we used SQL

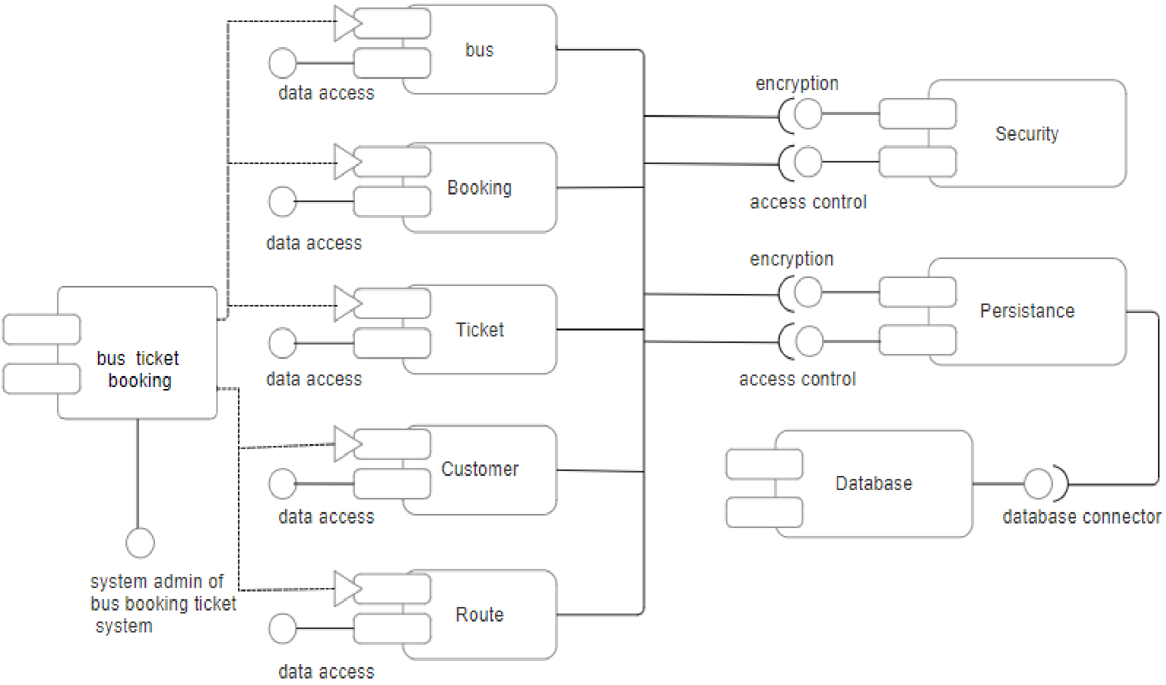
Server. The Visual Studio Code is used as the Integrated Development Environment(IDE). The Operating System used to develop this application is Windows 10.

9.5 Architecture

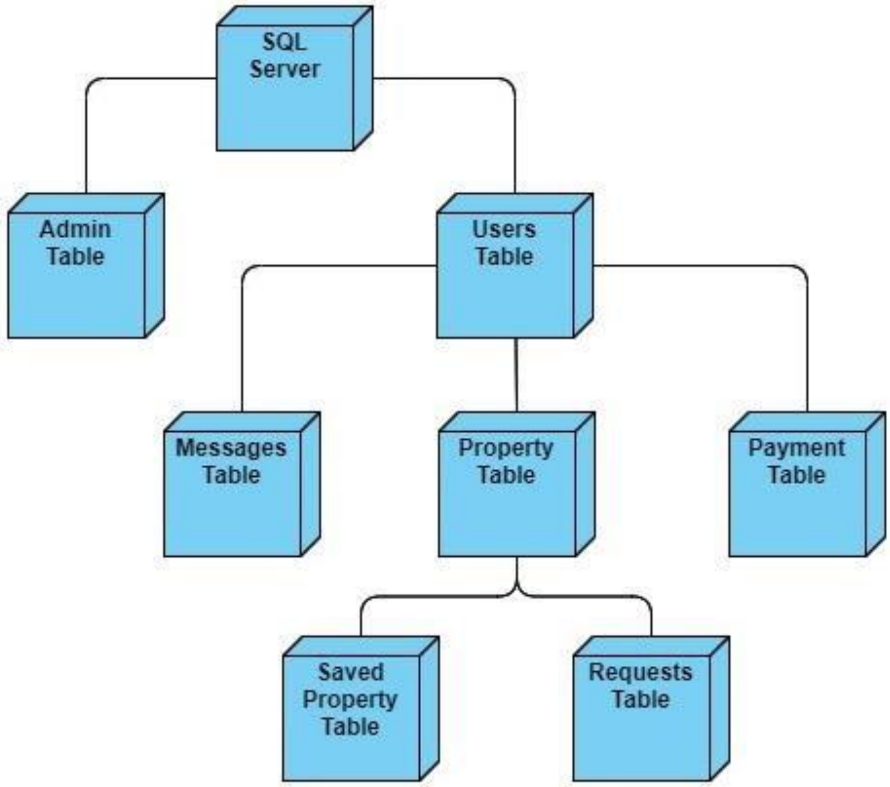
9.5.1 Package diagram



9.5.2 Component diagram



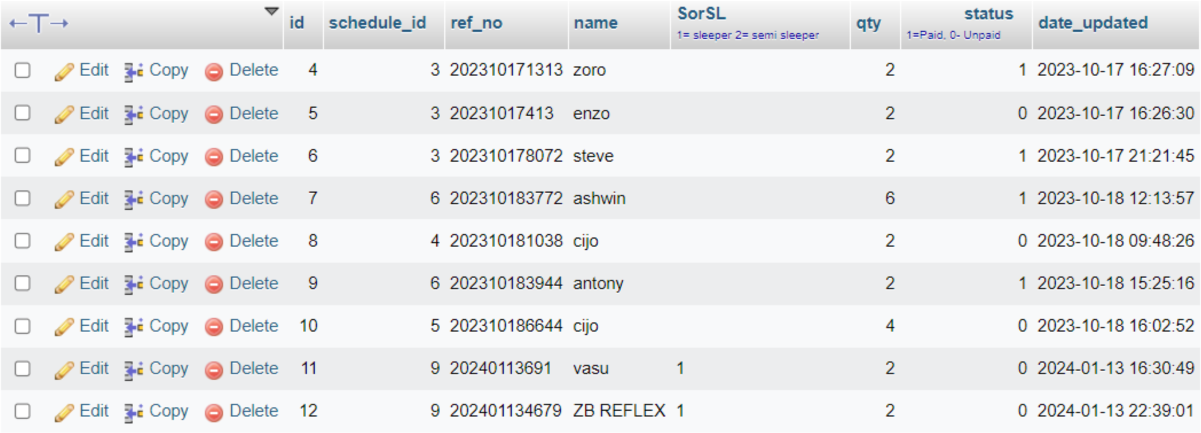
9.5.3 Deployment diagram



9.6 Database Design

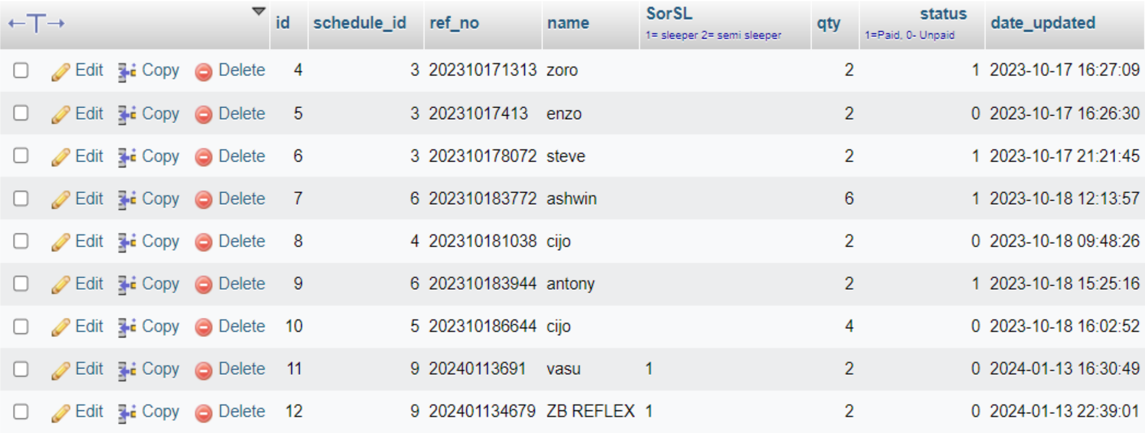
Data Base Name : bus_booking

Booked Table



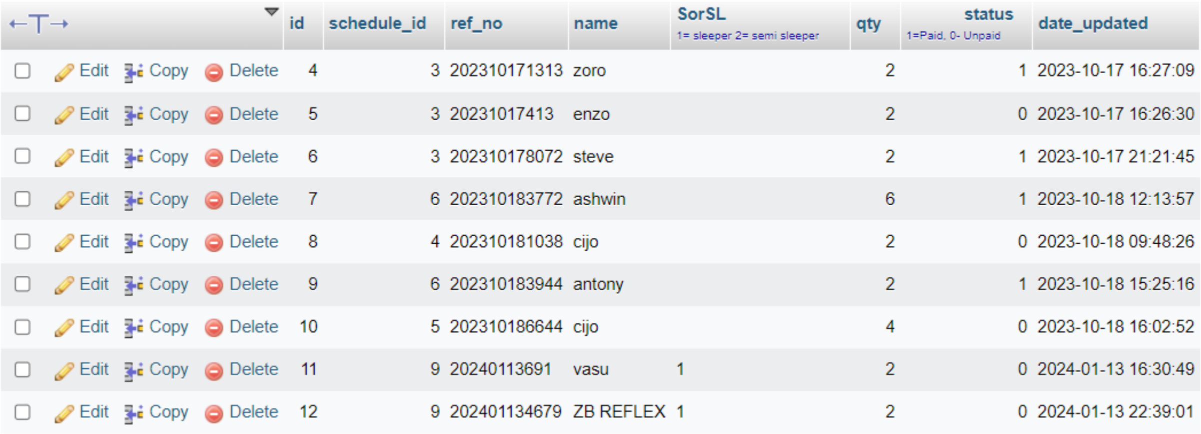
←T→			▼	id	schedule_id	ref_no	name	SorSL 1= sleeper 2= semi sleeper	qty	status 1=Paid, 0- Unpaid	date_updated
□	Edit	Copy	Delete	4	3	202310171313	zoro		2	1	2023-10-17 16:27:09
□	Edit	Copy	Delete	5	3	20231017413	enzo		2	0	2023-10-17 16:26:30
□	Edit	Copy	Delete	6	3	202310178072	steve		2	1	2023-10-17 21:21:45
□	Edit	Copy	Delete	7	6	202310183772	ashwin		6	1	2023-10-18 12:13:57
□	Edit	Copy	Delete	8	4	202310181038	cijo		2	0	2023-10-18 09:48:26
□	Edit	Copy	Delete	9	6	202310183944	antony		2	1	2023-10-18 15:25:16
□	Edit	Copy	Delete	10	5	202310186644	cijo		4	0	2023-10-18 16:02:52
□	Edit	Copy	Delete	11	9	20240113691	vasu	1	2	0	2024-01-13 16:30:49
□	Edit	Copy	Delete	12	9	202401134679	ZB REFLEX	1	2	0	2024-01-13 22:39:01

Location List table



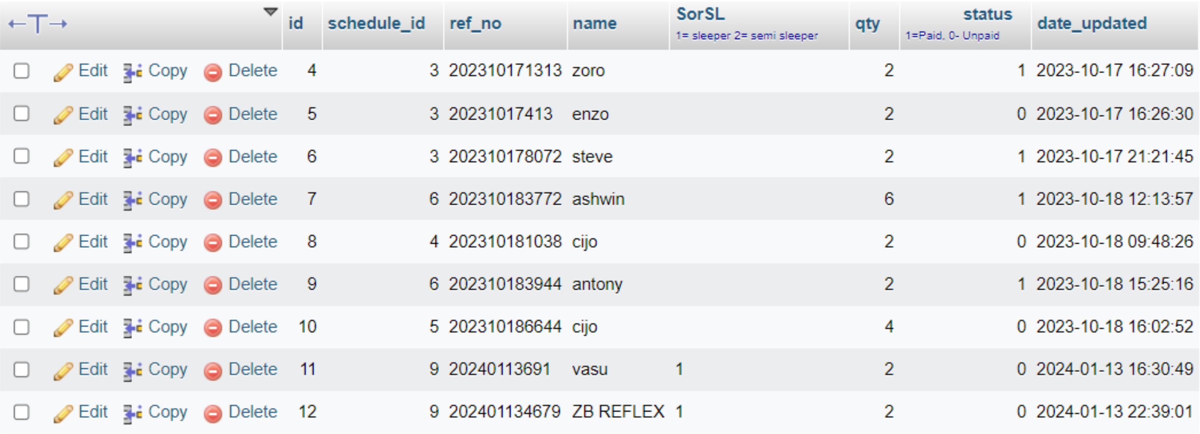
←T→			▼	id	schedule_id	ref_no	name	SorSL 1= sleeper 2= semi sleeper	qty	status 1=Paid, 0- Unpaid	date_updated
□	Edit	Copy	Delete	4	3	202310171313	zoro		2	1	2023-10-17 16:27:09
□	Edit	Copy	Delete	5	3	20231017413	enzo		2	0	2023-10-17 16:26:30
□	Edit	Copy	Delete	6	3	202310178072	steve		2	1	2023-10-17 21:21:45
□	Edit	Copy	Delete	7	6	202310183772	ashwin		6	1	2023-10-18 12:13:57
□	Edit	Copy	Delete	8	4	202310181038	cijo		2	0	2023-10-18 09:48:26
□	Edit	Copy	Delete	9	6	202310183944	antony		2	1	2023-10-18 15:25:16
□	Edit	Copy	Delete	10	5	202310186644	cijo		4	0	2023-10-18 16:02:52
□	Edit	Copy	Delete	11	9	20240113691	vasu	1	2	0	2024-01-13 16:30:49
□	Edit	Copy	Delete	12	9	202401134679	ZB REFLEX	1	2	0	2024-01-13 22:39:01

Schedule List Table



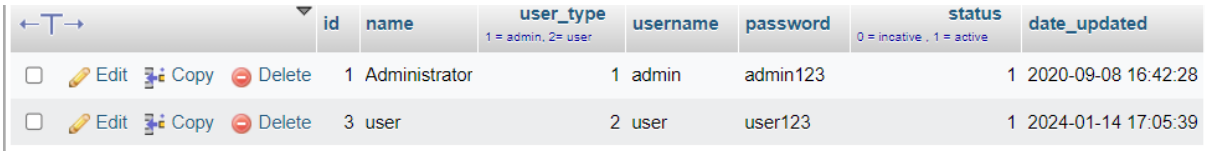
←T→			▼	id	schedule_id	ref_no	name	SorSL 1= sleeper 2= semi sleeper	qty	status 1=Paid, 0- Unpaid	date_updated
□	Edit	Copy	Delete	4	3	202310171313	zoro		2	1	2023-10-17 16:27:09
□	Edit	Copy	Delete	5	3	20231017413	enzo		2	0	2023-10-17 16:26:30
□	Edit	Copy	Delete	6	3	202310178072	steve		2	1	2023-10-17 21:21:45
□	Edit	Copy	Delete	7	6	202310183772	ashwin		6	1	2023-10-18 12:13:57
□	Edit	Copy	Delete	8	4	202310181038	cijo		2	0	2023-10-18 09:48:26
□	Edit	Copy	Delete	9	6	202310183944	antony		2	1	2023-10-18 15:25:16
□	Edit	Copy	Delete	10	5	202310186644	cijo		4	0	2023-10-18 16:02:52
□	Edit	Copy	Delete	11	9	20240113691	vasu	1	2	0	2024-01-13 16:30:49
□	Edit	Copy	Delete	12	9	202401134679	ZB REFLEX	1	2	0	2024-01-13 22:39:01

Bus Table



←T→	id	schedule_id	ref_no	name	SorSL 1= sleeper 2= semi sleeper	qty	status 1=Paid, 0- Unpaid	date_updated
□ Edit Copy Delete	4	3	202310171313	zoro		2	1	2023-10-17 16:27:09
□ Edit Copy Delete	5	3	20231017413	enzo		2	0	2023-10-17 16:26:30
□ Edit Copy Delete	6	3	202310178072	steve		2	1	2023-10-17 21:21:45
□ Edit Copy Delete	7	6	202310183772	ashwin		6	1	2023-10-18 12:13:57
□ Edit Copy Delete	8	4	202310181038	cijo		2	0	2023-10-18 09:48:26
□ Edit Copy Delete	9	6	202310183944	antony		2	1	2023-10-18 15:25:16
□ Edit Copy Delete	10	5	202310186644	cijo		4	0	2023-10-18 16:02:52
□ Edit Copy Delete	11	9	20240113691	vasu	1	2	0	2024-01-13 16:30:49
□ Edit Copy Delete	12	9	202401134679	ZB REFLEX	1	2	0	2024-01-13 22:39:01

Users table



←T→	id	name	user_type 1 = admin, 2= user	username	password	status 0 = incative , 1 = active	date_updated
□ Edit Copy Delete	1	Administrator	1	admin	admin123	1	2020-09-08 16:42:28
□ Edit Copy Delete	3	user	2	user	user123	1	2024-01-14 17:05:39

10.TESTING

10.1 Test cases

Test Scenario: Checking Login Functionality

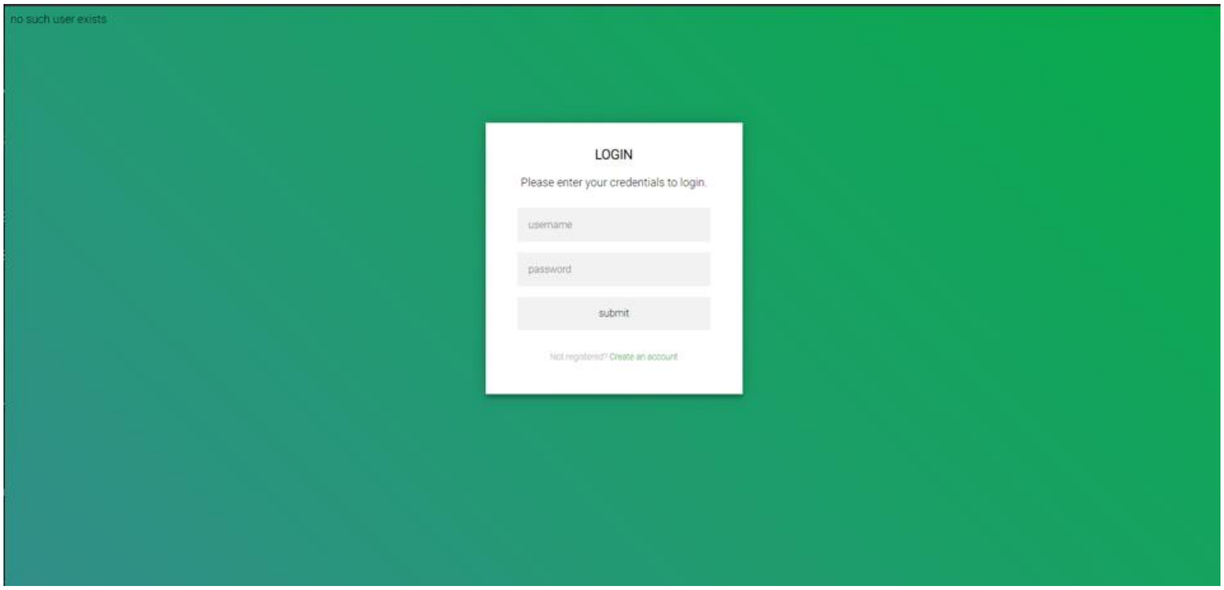
Test Case 1: Invalid User Login

An unregistered or unauthorized login attempt must be blocked.

Precondition: Unauthorized users do not have valid credentials to log in . **Assumption:** Only authorized users have access to valid credentials for logging in.**Test Steps:**

1. Go to Login Page
2. Enter credentials
3. Submit the credentials

Expected Result: A login attempt with invalid or wrong input results in an unsuccessful login message.



no such user exists

LOGIN

Please enter your credentials to login.

2: Invalid Admin Login

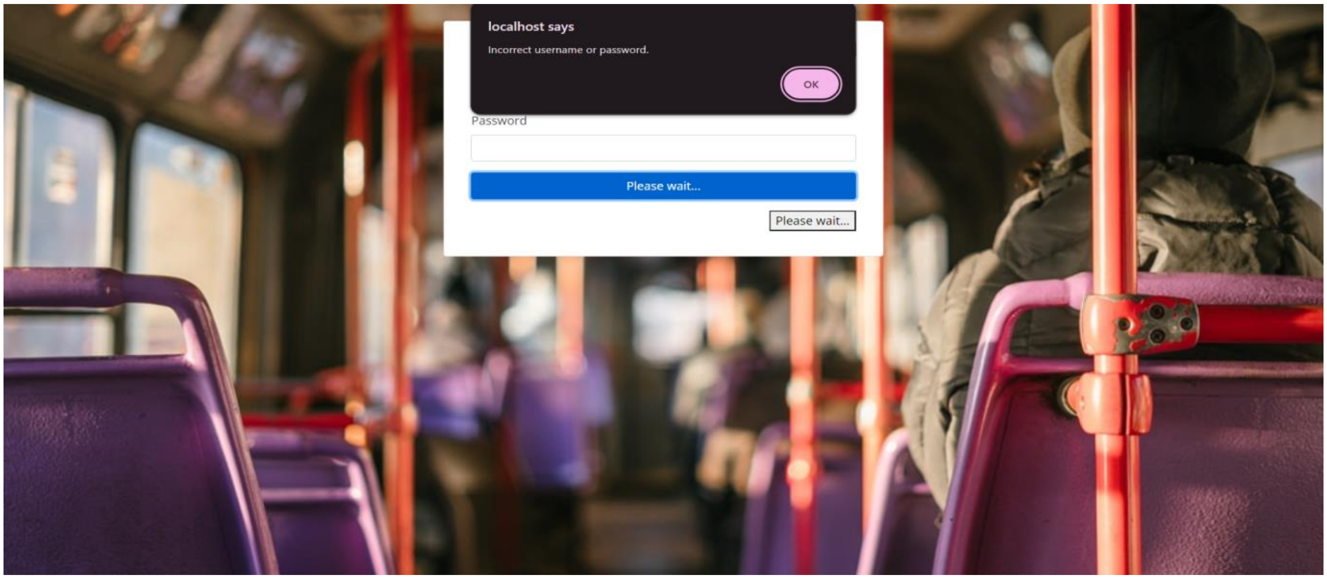
An unregistered or unauthorized login attempt must be blocked.

Precondition: Unauthorized admin do not have valid credentials to log in. **Assumption:** Only authorized admin have access to valid credentials for logging in.

Test Steps:

1. Go to Login Page
2. Enter credentials
3. Submit the credentials

Expected Result: A login attempt by admin with invalid or wrong input results in an unsuccessful login message.



localhost says

Incorrect username or password.

Test Case 3: Check results on not entering new matching passwords by User at Update Profile page

All the attempts to update profile must be blocked if the new passwords entered by the user don’t match.

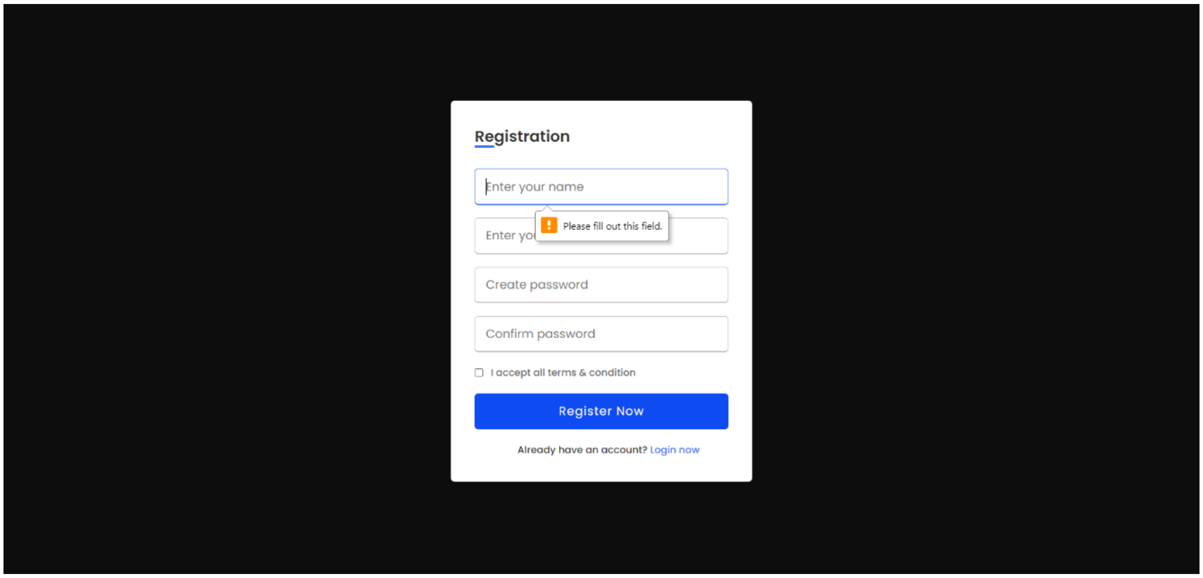
Precondition: User accidentally forgets the new password entered.

Assumption: Only authorized users have access to update their profile.

Test Steps:

1. Log in to the system as User using the correct user’s credentials.
2. Go to the Update Profile page.
3. Submit the updated details of the profile by entering mismatched new passwords.

Expected Result: An attempt by the user to update their profile by entering mismatched new passwords, results in an unsuccessful error message.



Registration

Please fill out this field.

Enter your name
Enter yo
Create password
Confirm password
□ I accept all terms & condition
Register Now
Already have an account? Login now

Test Case

4: results on not correctly entering old password by User at Update Profile page

All the attempts to update profile must be blocked if the old password entered by the user is wrong.

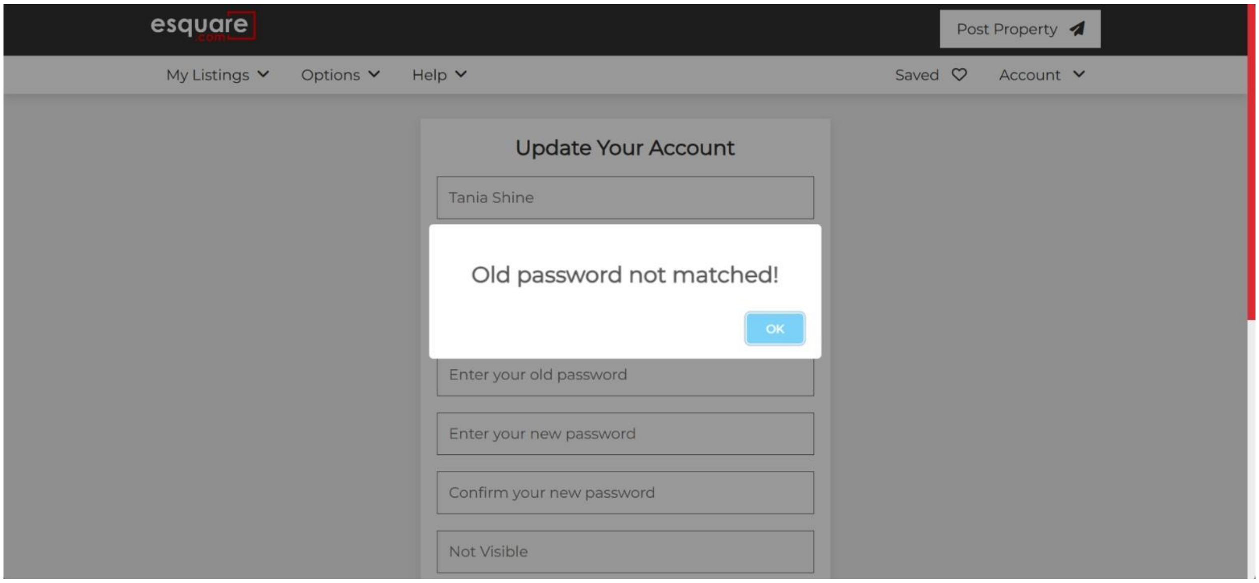
Precondition: User accidentally mistypes or forgets the old password.

Assumption: Only authorized users have access to update their profile.

Test Steps:

1. Log in to the system as User using the correct user's credentials.
2. Go to the Update Profile page.
3. Submit the updated details of the profile by entering wrong old password.

Expected Result: An attempt by the user to update their profile by entering wrong old password, results in an unsuccessful error message.



Update Your Account

Tania Shine

Old password not matched!

Enter your old password

Enter your new password

Confirm your new password

Not Visible

Test Case

5: Check results on not entering new matching passwords by Admin at Update Profile page

All the attempts to update profile must be blocked if the new passwords entered by the Admin don't match.

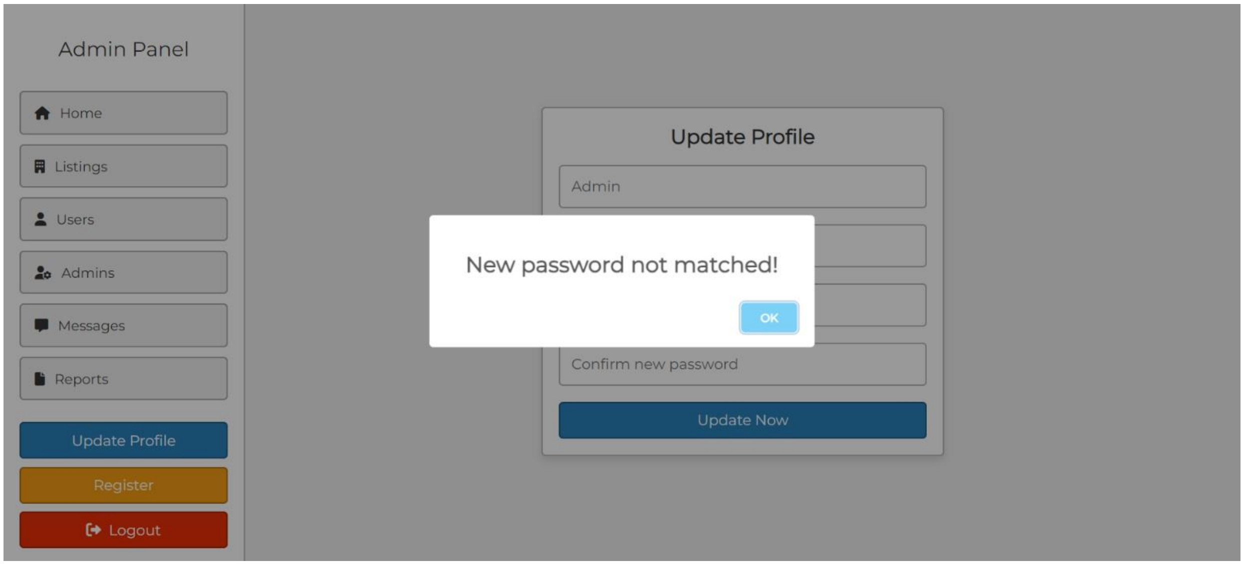
Precondition: Admin accidentally forgets the new password entered.

Assumption: Only authorized Admin have access to update their profile.

Test Steps:

1. Log in to the system as Admin using the correct user's credentials.
2. Go to the Update Profile page.
3. Submit the updated details of the profile by entering mismatched new passwords.

Expected Result: An attempt by the Admin to update their profile by entering mismatched new passwords, results in an unsuccessful error message.



New password not matched!

Test Case Check

6: results on checking files uploaded by User at Post Property page

All the attempts to post property must be blocked if any of the files uploaded exceeds the size limit.

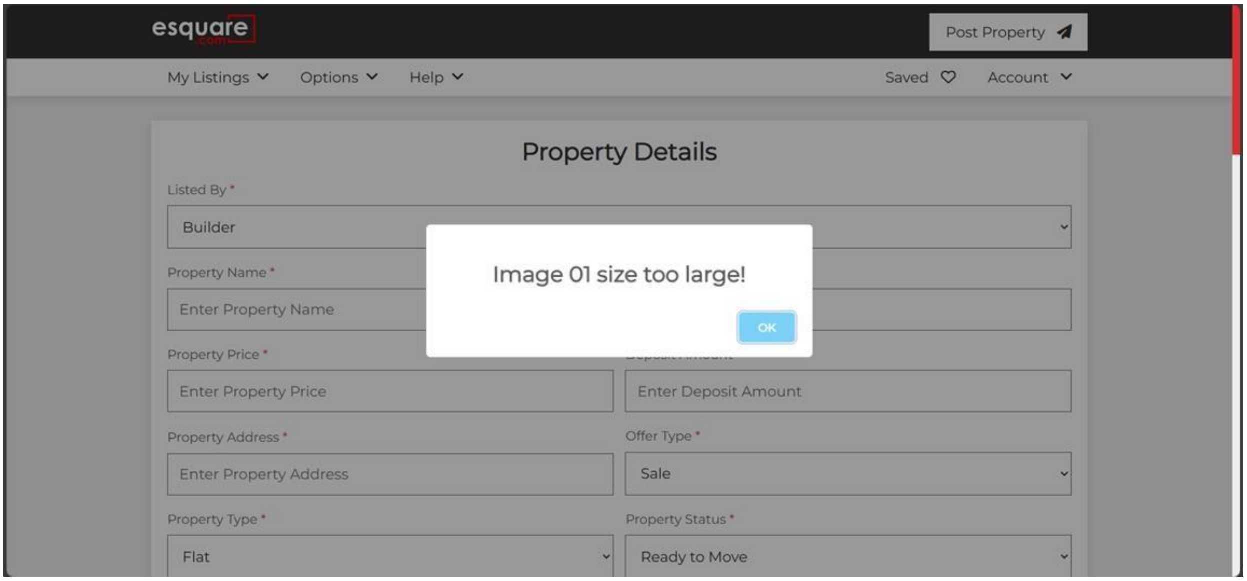
Precondition: User uploads an image of huge file size.

Assumption: Only authorized users have access to post property.

Test Steps:

1. Log in to the system as User using the correct user’s credentials.
2. Go to the Post Property page.
3. Submit the details of the property by uploading a file that exceeds the size limit.

Expected Result: An attempt by the user to post their property by uploading a file that exceeds the size limit, results in an unsuccessful error message.



Property Details

Listed By * Builder

Property Name * Enter Property Name

Image 01 size too large!

Property Price * Enter Property Price

Deposit Amount Enter Deposit Amount

Property Address * Enter Property Address

Offer Type * Sale

Property Type * Flat

Property Status * Ready to Move

10.2 Test Report

In all the test cases, as we got the expected result, we can easily come to the conclusion that the testing process was successful and hence proved that our application is efficient enough to store and process data. Almost about 99.9% of test cases were passed successfully.

10.3 Sample Code used for testing

Test Case 1: Invalid User Login

```
if(isset($_POST['submit'])){
   $email = $_POST['email'];
   $email = filter_var($email, FILTER_SANITIZE_STRING);
   $pass = sha1($_POST['pass']);
   $pass = filter_var($pass, FILTER_SANITIZE_STRING);
   $verify_users = $conn->prepare("SELECT * FROM `users` WHERE email = ? AND
password = ? AND status = 'verified' LIMIT 1");
   $verify_users->execute([$email, $pass]);
   $row = $verify_users->fetch(PDO::FETCH_ASSOC);          if($verify_users-
>rowCount() > 0){
      setcookie('user_id', $row['id'], time() + 60*60*24*30, '/');
header('location:home.php');
   } else {
      $warning_msg[] = 'Incorrect Email or Password, or your account is not verified.';
   }
}
```

Test Case 2: Invalid Admin Login

```
  if(isset($_POST['submit'])){
 $name = $_POST['name'];
  $name = filter_var($name, FILTER_SANITIZE_STRING);
  $pass = ($_POST['pass']);
  $pass = filter_var($pass, FILTER_SANITIZE_STRING);
  $select_admins = $conn->prepare("SELECT * FROM `admins` WHERE name = ? AND
password = ? LIMIT 1");
  $select_admins->execute([$name, $pass]);
    $row = $select_admins->fetch(PDO::FETCH_ASSOC);
 if($select_admins->rowCount() > 0){        setcookie('admin_id',
 $row['id'], time() + 60*60*24*30, '/');
 header('location:dashboard.php');
  }else{
     $warning_msg[] = 'Incorrect username or password!';
  }
}
```

Test Case 3: Check results on not entering new matching passwords by User at Update Profile page

```
if($empty_pass           !=           $old_pass){
if($old_pass != $prev_pass){
          $warning_msg[] = 'Old password not matched!';
      }
      elseif($c_pass != $new_pass){
          $warning_msg[] = 'Confirm password not matched!';
      }
}
```

Test Case 4: Check results on not correctly entering old password by User at Update Profile page

```
if($empty_pass           !=           $old_pass){
if($old_pass != $prev_pass){
```

```
            $warning_msg[] = 'Old password not matched!';
         }
         elseif($c_pass != $new_pass){
            $warning_msg[] = 'Confirm password not matched!';
         }
   }
```

Test Case 5: Check results on not storing all the required fields by User at Contact Us page

```
<form action="" method="post">
      <h3>Get in Touch</h3>
      <input type="text" name="name" required maxlength="50" placeholder="Enter your Name" class="box">
      <input type="email" name="email" required maxlength="50" placeholder="Enter your Email" class="box">
      <input type="number" name="number" required maxlength="10" max="9999999999" min="0" placeholder="Enter your Number" class="box">
      <textarea name="message" placeholder="Enter your Message" required maxlength="1000" cols="30" rows="10" class="box"></textarea>
      <input type="submit" value="Send Message" name="send" class="btn">      </form>
```

11.Transition

11.1 System Implementation

Implementation is the process of having the system personal checks out and put equipment's to use, train the users to use the new system and construct any file that are needed to see it. The final and important phases in the system lifecycle are the implementation of new system. The file conversion is the most time consuming and expensive activity in the implementation stage. System implementation refers to the step necessary to install a new system to put into the operation. The implementation has different meaning, raining from the conversion of basic application to complete replacement of computer system. Implementation includes all these activities that take place to covert from old system to new system. The new system may be totally new replacing and existing manual or automated system or it may be major modification to an existing system. The method of implementation and time scale adopted is found out initially. The system is tested properly and at the same time the users are trained in new procedure. Proper implementation is essential to provide a reliable system to meet organization requirements. Successful implementation may not guarantee improvement in the organization using the new system, but it will prevent improper installation. The implementation involves the following things:-

1. Careful planning.
2. Investigation of the system and constrains 3. Design the methods to achieve the changeover.
4. Train the staff in the changed phase.
5. Evaluation of change over method.

After converting as a package, it has been delivered to the customers where it is implemented and tailored to meet the specific requirements.

11.2 System Maintenance

Like housework, dirty clothes and weeds, system work never seems to an end; users almost always want changes or encounter problems. Thus the system maintenance part of the system process deserves special attention. It is during system maintenance that the analyst: 1. Resolves necessary changes

2. Correct errors.

3. Enhance or modifies the system.

4. Assign staff to perform maintenance activities.

5. Provides for scheduled maintenance.

Most system spends the bulk of their time in the maintenance phase, with constant enhancements and repairs. Studies show that more money is spent in this forth phase than in all of the others combined. Writing system is that require as little maintenance as possible is one of the primary goal as well as one of the benefits of today's modern methodology of software development. Maintenance ids divided into three categories.

1. Corrective maintenance.

2. Adaptive maintenance.

3. Preventive maintenance.

11.2.1 Corrective maintenance

It has to do with the removal of residual errors present in the product when it is delivered as well as errors introduced into the software during its maintenance accounts for about 20% of the maintenance cost.

11.2.2 Adaptive maintenance

It involves adjusting the application to changes in the environment, that is a new release of the hardware or the operating system or anew database system. It also accounts for nearly 20% of the maintenance cost.

11.2.3 Preventive maintenance

It involves changing the software to improve some qualities. It accounts for over 50% of maintenance costs. Here changes are due to the functions offered by the application, and new functions, improve the performance of application etc.

Maintenance is not such a difficult task. The above three maintenance tasks can be easily carried out under this system.

12. ANNEXURE

12.1 References

Websites

[1] YouTube

[2] Random sites https://W3schools.com/

http://www.stackoverflow.com/questions http://tutorialspoint.com/

https://www.geekforgeeks.com

12.2 Annexure I: User Interview Questionnaires

1. How would you approach the system?

2. What about usability of this system?

3. What are normal project requirements?

12.3 CONCLUSION

The Online Bus Booking System aims to enhance the travel experience for users by providing a convenient and efficient platform for booking bus tickets. With its user-friendly interface and advanced features, it is poised to become a reliable solution for both traveller's and bus operators

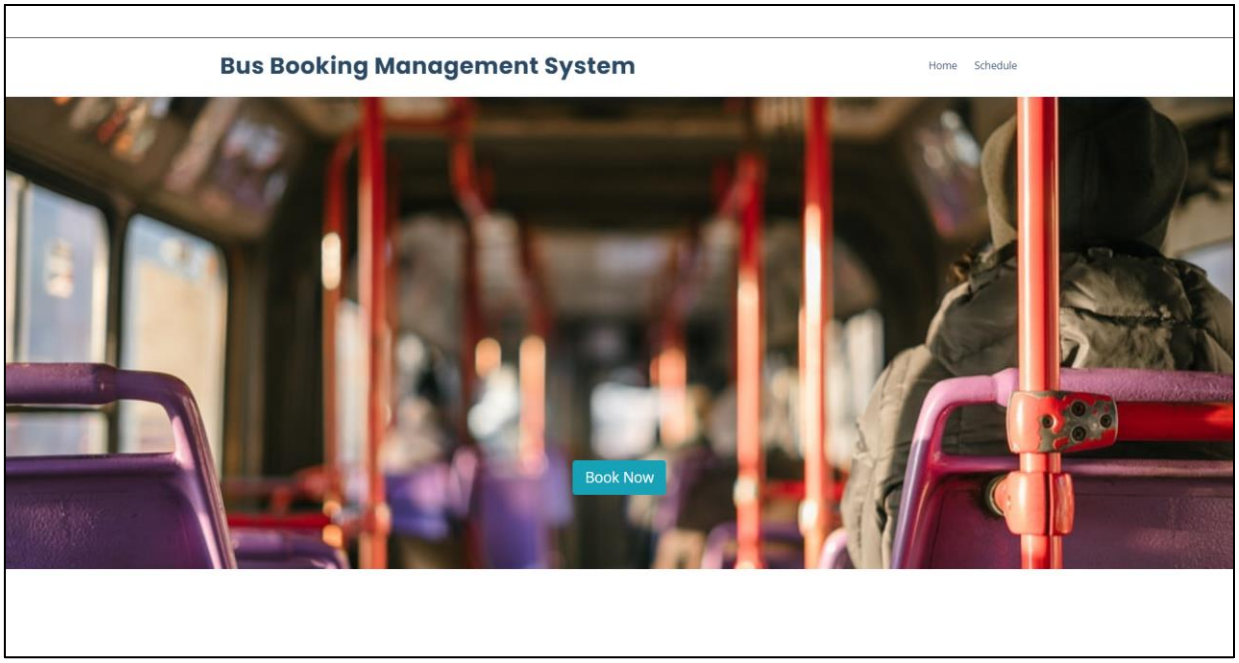
12.4 SAMPLE CODE

- Screenshots

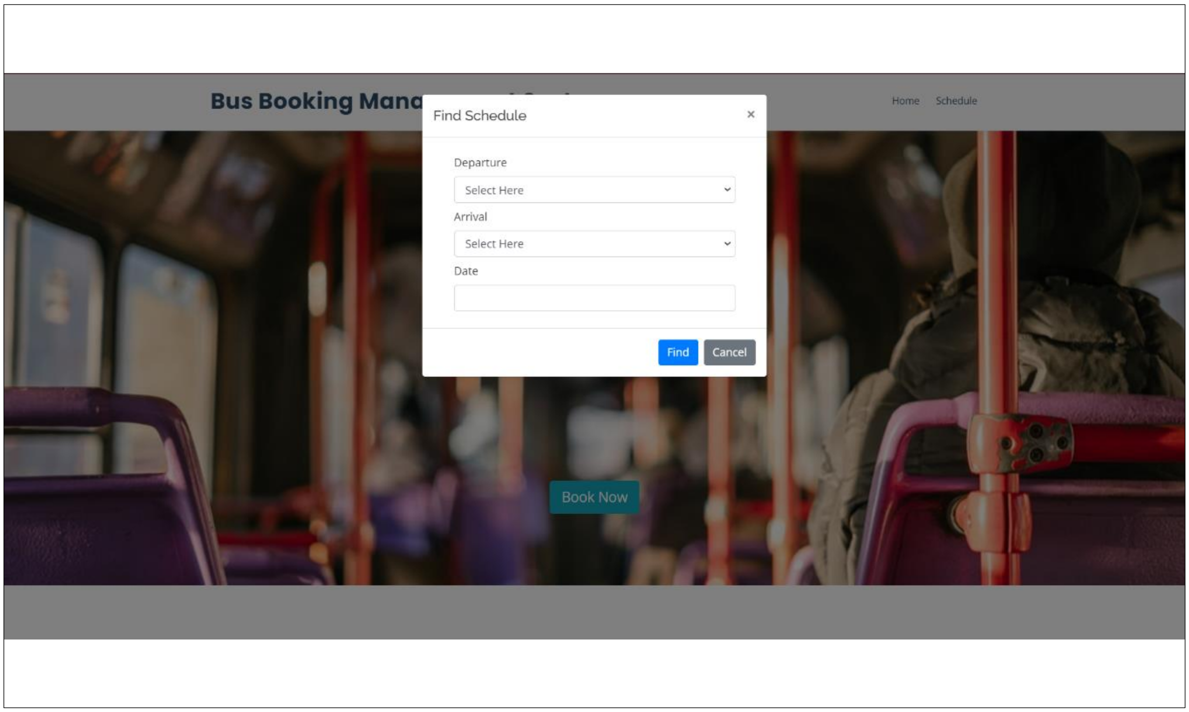
12.4.1. Main

USER HOME PAGE

USER
TIME



USER BOOK SELECTINGPAGE



Find Schedule ×

Departure

Select Here

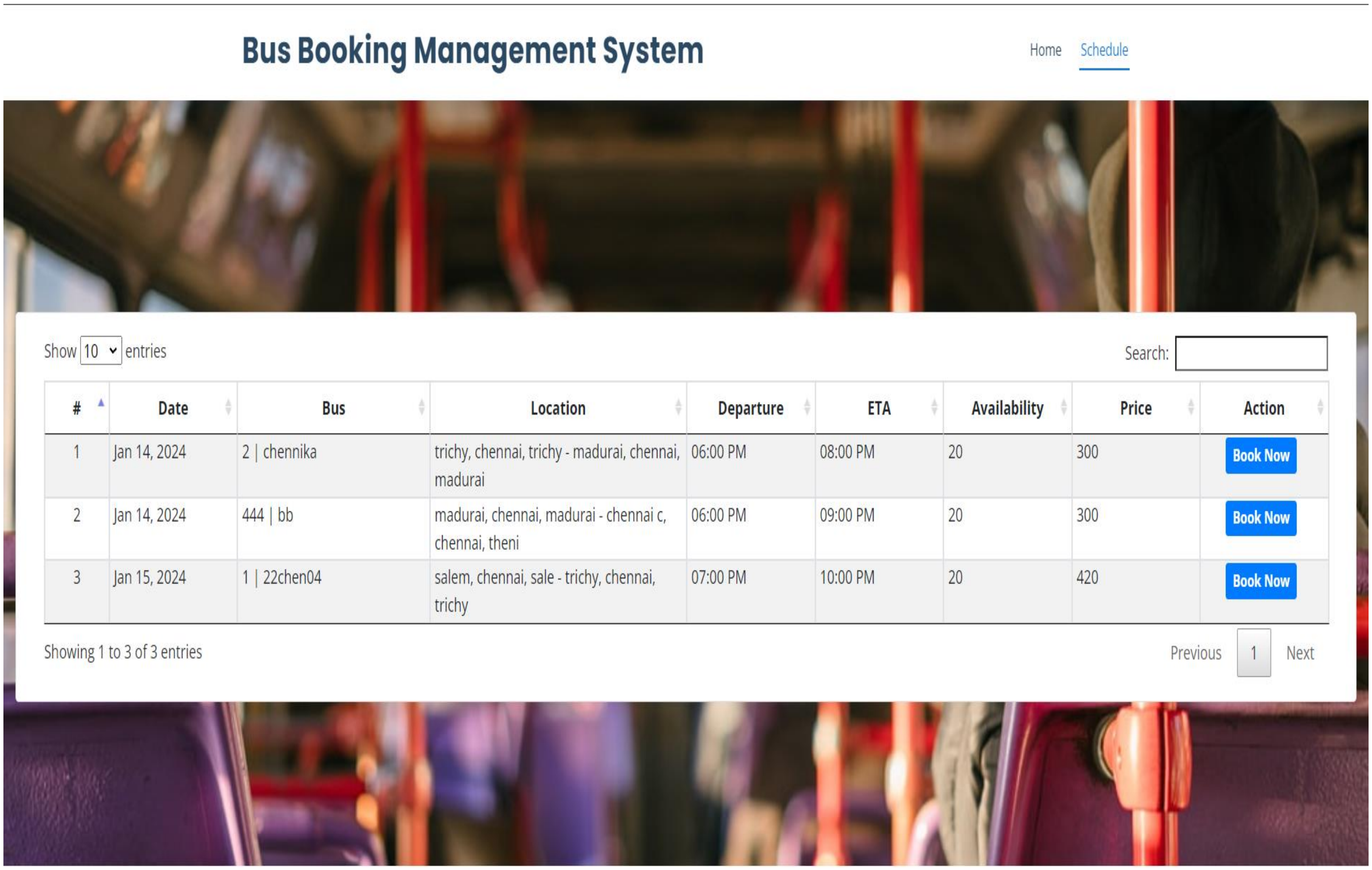
Arrival

Select Here

Date

Find Cancel

SCHEDULE PAGE

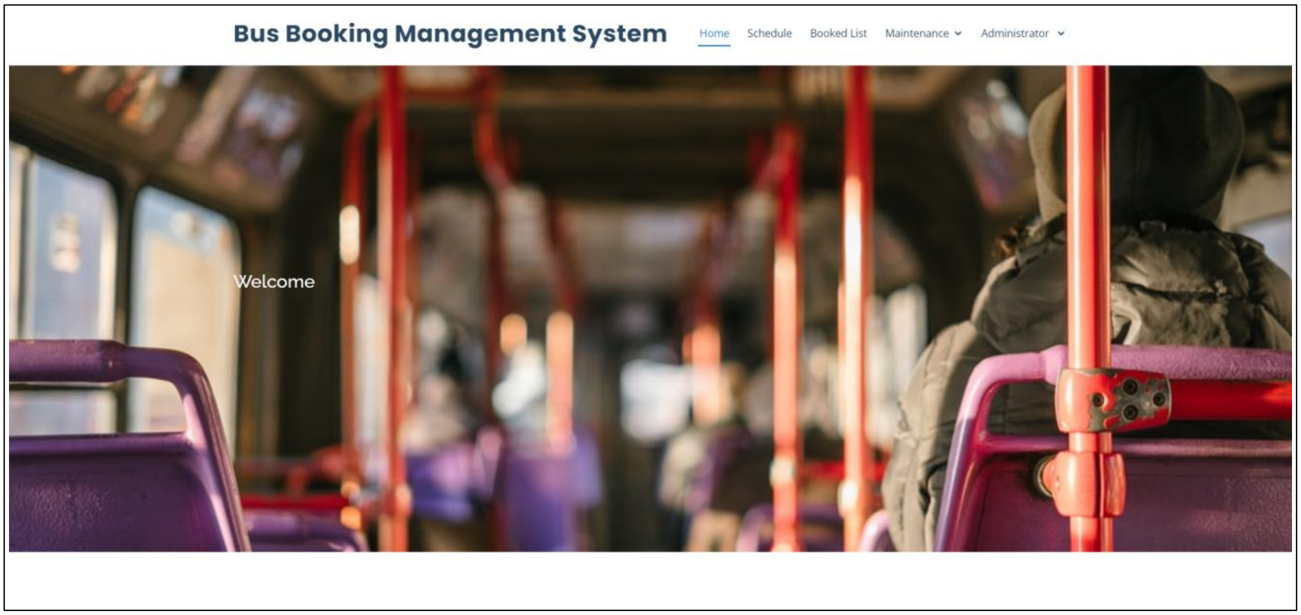


Bus Booking Management System

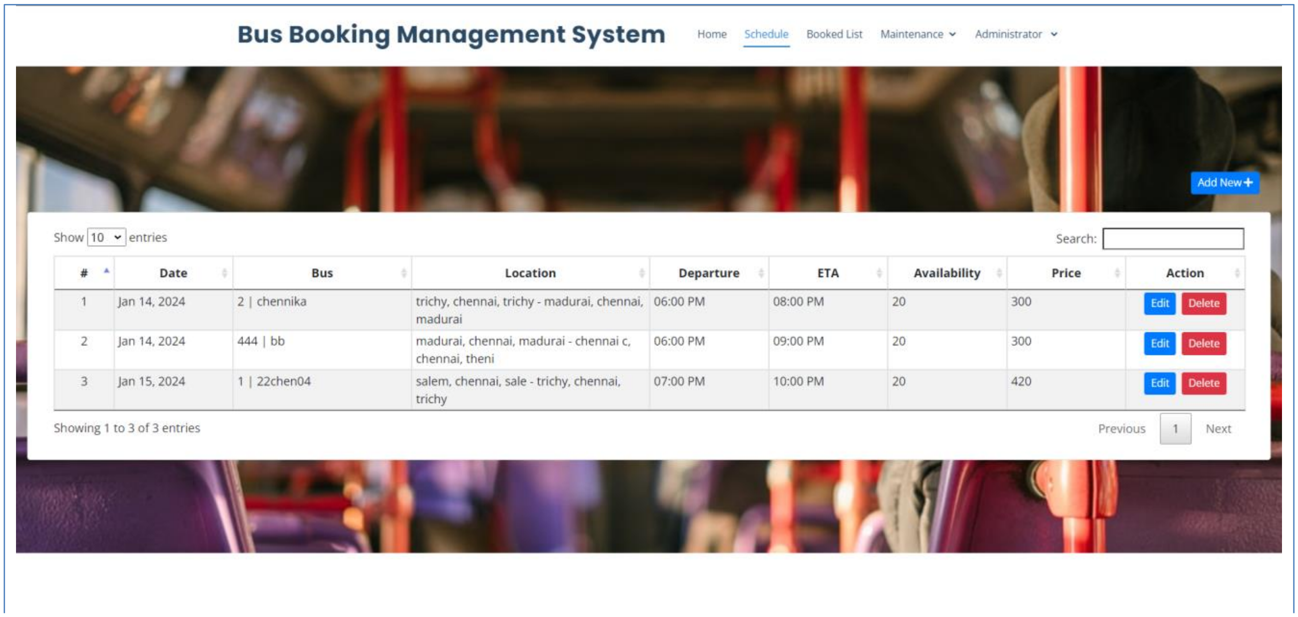
#	Date	Bus	Location	Departure	ETA	Availability	Price	Action
1	Jan 14, 2024	2 | chennika	trichy, chennai, trichy - madurai, chennai, madurai	06:00 PM	08:00 PM	20	300	Book Now
2	Jan 14, 2024	444 | bb	madurai, chennai, madurai - chennai c, chennai, theni	06:00 PM	09:00 PM	20	300	Book Now
3	Jan 15, 2024	1 | 22chen04	salem, chennai, sale - trichy, chennai, trichy	07:00 PM	10:00 PM	20	420	Book Now

Showing 1 to 3 of 3 entries

ADMIN LISTINGS PAGE



ADMIN SCHEDULE LIST PAGE



Show 10 entries

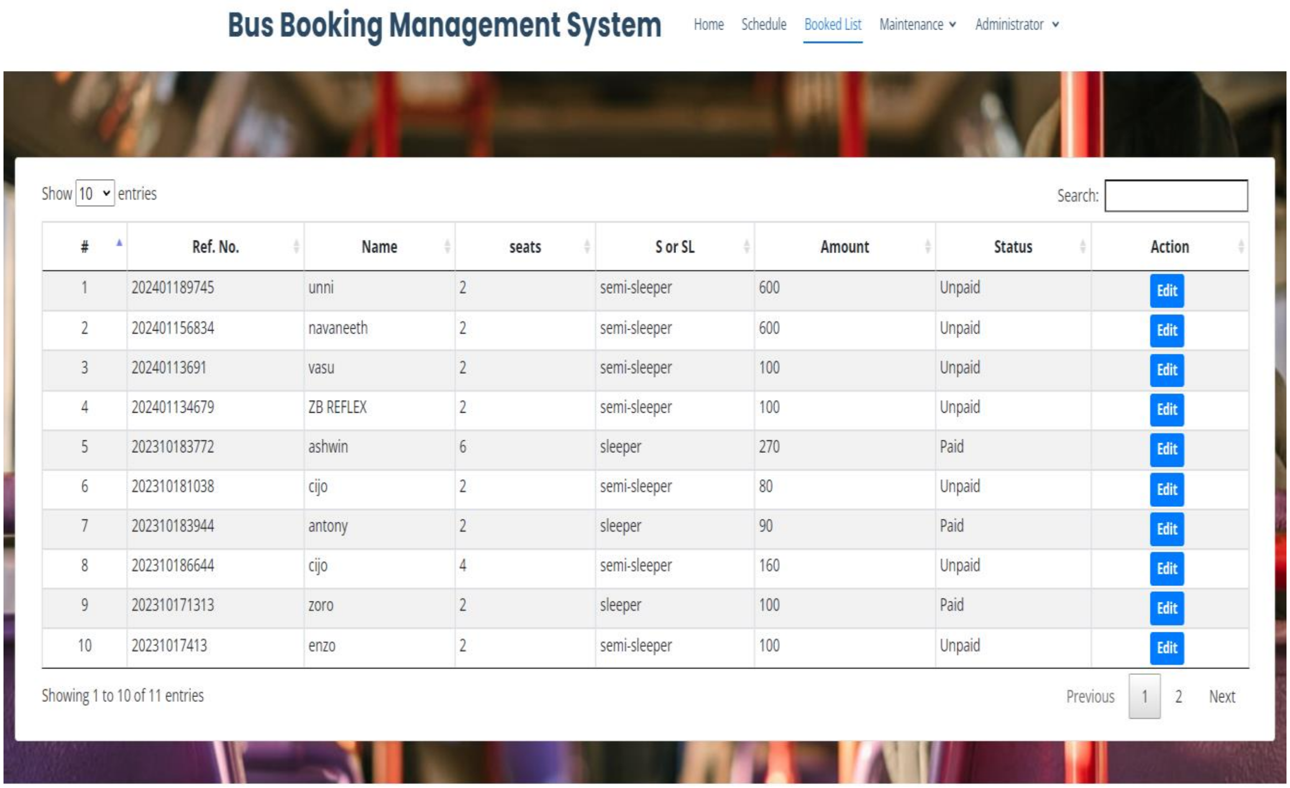
Search:

#	Date	Bus	Location	Departure	ETA	Availability	Price	Action
1	Jan 14, 2024	2 | chennika	trichy, chennai, trichy - madurai, chennai, madurai	06:00 PM	08:00 PM	20	300	Edit Delete
2	Jan 14, 2024	444 | bb	madurai, chennai, madurai - chennai c, chennai, theni	06:00 PM	09:00 PM	20	300	Edit Delete
3	Jan 15, 2024	1 | 22chen04	salem, chennai, sale - trichy, chennai, trichy	07:00 PM	10:00 PM	20	420	Edit Delete

Showing 1 to 3 of 3 entries

Previous 1 Next

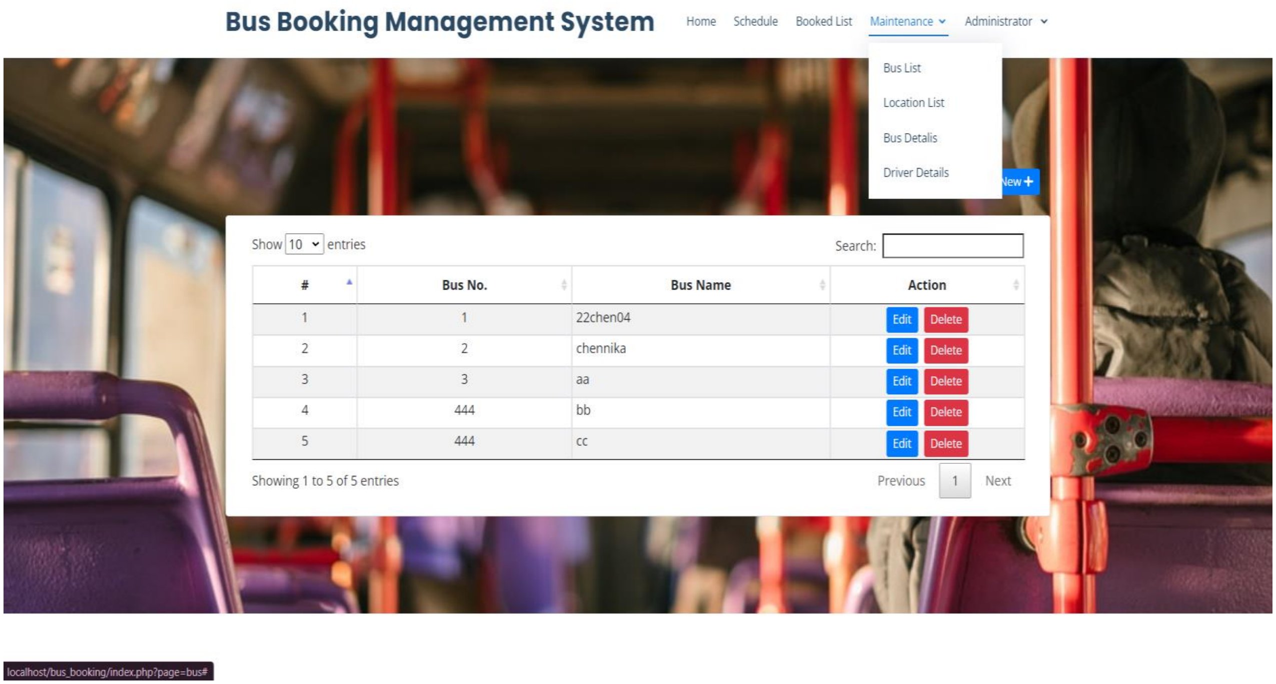
ADMIN BOOKED LIST PAGE



#	Ref. No.	Name	seats	S or SL	Amount	Status	Action
1	202401189745	unni	2	semi-sleeper	600	Unpaid	Edit
2	202401156834	navaneeth	2	semi-sleeper	600	Unpaid	Edit
3	20240113691	vasu	2	semi-sleeper	100	Unpaid	Edit
4	202401134679	ZB REFLEX	2	semi-sleeper	100	Unpaid	Edit
5	202310183772	ashwin	6	sleeper	270	Paid	Edit
6	202310181038	cijo	2	semi-sleeper	80	Unpaid	Edit
7	202310183944	antony	2	sleeper	90	Paid	Edit
8	202310186644	cijo	4	semi-sleeper	160	Unpaid	Edit
9	202310171313	zoro	2	sleeper	100	Paid	Edit
10	20231017413	enzo	2	semi-sleeper	100	Unpaid	Edit

Showing 1 to 10 of 11 entries

ADMIN DETAILS PAGE



#	Bus No.	Bus Name	Action
1	1	22chen04	Edit Delete
2	2	chennika	Edit Delete
3	3	aa	Edit Delete
4	444	bb	Edit Delete
5	444	cc	Edit Delete

Showing 1 to 5 of 5 entries

SAMPLE

12.4.2. Main

INDEX PAGE **index.php**

```
<!DOCTYPE html>

<html lang="en">


<head>

  <meta charset="utf-8">

  <meta content="width=device-width, initial-

scale=1.0" name="viewport">


  <title>Bus Booking Management

System</title>


</head>




<body>

<?php session_start() ?>

<?php


  header('index.php?page=home');

 include 'header.php'; ?>

<?php if(isset($_SESSION['login_id']))

include 'admin_navbar.php'; else include

'navbar.php'; ?>
```

```
<div class="toast" id="alert_toast"
role="alert" aria-live="assertive" aria-
atomic="true">
  <div class="toast-body text-white">
  </div>
</div>

    <?php
    if(isset($_GET['page']) &&
!empty($_GET['page']))
      include($_GET['page'].'.php');
    else
      include('home.php');

    ?>

    <div class="modal fadeIn" tabindex="-1"
id="uni_modal">
      <div class="modal-dialog">
        <div class="modal-content">
          <div class="modal-header">
            <h5 class="modal-title"></h5>
            <button type="button" class="close"
data-dismiss="modal" aria-label="Close">
              <span aria-
hidden="true">&times;</span>
            </button>
          </div>
          <div class="modal-body">
          </div>
          <div class="modal-footer">
            <button type="button" class="btn btn-
primary submit" onclick="$('#uni_modal
form').submit()">
              <?php echo
isset($_SESSION['login_id']) ? 'Save' : 'Find'
?>
          </button>
            <button type="button" class="btn btn-
secondary" data-
dismiss="modal">Cancel</button>
          </div>
        </div>
      </div>
    </div>
    <div class="modal fadeIn" tabindex="-1"
id="confirm_modal">
      <div class="modal-dialog">
        <div class="modal-content">
          <div class="modal-header">
            <h5 class="modal-
title">Confirmation</h5>
            <button type="button" class="close"
data-dismiss="modal" aria-label="Close">
              <span aria-
hidden="true">&times;</span>
            </button>
          </div>
          <div class="modal-body">
          </div>
          <div class="modal-footer">
            <button type="button" class="btn btn-
primary" id="confirm"
onclick="">Continue</button>
            <button type="button" class="btn btn-
default" data-
dismiss="modal">Cancel</button>
          </div>
        </div>
      </div>
    </div>

    <div class="modal fadeIn" tabindex="-1"
id="book_modal">
      <div class="modal-dialog">
        <div class="modal-content">
          <div class="modal-header">
            <h5 class="modal-title"></h5>
            <button type="button" class="close"
data-dismiss="modal" aria-label="Close">
            <span aria-
hidden="true">&times;</span>
            </button>
          </div>
          <div class="modal-body">

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





 <div id="preloader"></div>
 <a href="#" class="back-to-top"><i
class="icofont-simple-up"></i></a>
 <script src="assets/js/main.js"></script>

</body>
<script>
 window.uni_modal =
function($title=",$url=",$book = 0){
  $('#uni_modal .modal-title').html($title);
    start_load();

  $.ajax({
   url:$url,
   error:err=>console.log(err),
   success:function(resp){
    $('#uni_modal .modal-body').html(resp)
    if('<?php echo
!isset($_SESSION['login_id']) ?>' == 1){
        if($book == 1){
                $('#uni_modal
.submit').html('Book')
              }else{
                $('#uni_modal
.submit').html('Find')

              }
            }
        $('#uni_modal .modal-footer').show()
        $('#uni_modal').modal('show')
      },
      complete:function(){
        end_load();

      }
    })
  }
  window._conf =
function($msg='',$func='',$params = []){
     $('#confirm_modal
#confirm').attr('onclick',$func+"("+$params.jo
in(',')+")")
     $('#confirm_modal .modal-
body').html($msg)
     $('#confirm_modal').modal('show')
  }
  window.start_load = function(){
    $('body').prepend('<di
id="preloader2"></di>')
```

```
 }
 window.end_load = function(){
  $('#preloader2').fadeOut('fast', function() {
    $(this).remove();
   })
 }
 window.alert_toast= function($msg =
'TEST',$bg = 'success'){
   $('#alert_toast').removeClass('bg-success')
   $('#alert_toast').removeClass('bg-danger')
   $('#alert_toast').removeClass('bg-info')
   $('#alert_toast').removeClass('bg-warning')

  if($bg == 'success')
   $('#alert_toast').addClass('bg-success')
  if($bg == 'danger')
   $('#alert_toast').addClass('bg-danger')
  if($bg == 'info')
   $('#alert_toast').addClass('bg-info')
  if($bg == 'warning')
   $('#alert_toast').addClass('bg-warning')
  $('#alert_toast .toast-body').html($msg)

$('#alert_toast').toast({delay:3000}).toast('sho
w');
 }
 $(document).ready(function(){
 })
</script> </html>
```

ADMIN PAGE **admin_navbar.php**

```
   <header id="header" class="fixed-top">
   <div class="container d-flex align-items-center">

   <h1 class="logo mr-auto"><a href="./index.php?page=home">Bus Booking Management System</a></h1>

 <nav class="nav-menu d-none d-lg-block" id='top-nav'>
 <ul>
 <li class="nav-home"><a href="./index.php?page=home">Home</a></li>
 <li class="nav-schedule"><a href="./index.php?page=schedule">Schedule</a></li>
 <li class="nav-booked"><a href="./index.php?page=booked">Booked List</a></li>
 <li class="drop-down nav-bus nav-location"><a href="#">Maintenance</a>
 <ul>
 <li><a href="./index.php?page=bus">Bus List</a></li>
 <li><a href="./index.php?page=location">Location List</a></li>
 <li><a href="./index.php?page=busdetail">Bus Detalis</a></li>
 <li><a href="#">Driver Details</a></li>

 </ul>
 </li>
 <li class="drop-down nav-user"><a href="#"><?php echo $_SESSION['login_name'] ?> </a>
 <ul>
 <li><a href="./index.php?page=user">Users</a></li>
 <li><a href="javascript:void(0)" id="manage_account">Manage Account</a></li>
 <li><a href="./logout.php">Logout</a></li>

 </ul>
 </li>
 </ul>
 </nav>



</div>
</header>
<script>
```

```
$(document).ready(function(){
var page = '<?php echo isset($_GET['page']) ? $_GET['page'] : '' ?>';
if(page != ''){
$('#top-nav li').removeClass('active')
$('#top-nav li.nav-'+page).addClass('active')
}
$('#manage_account').click(function(){
uni_modal('Manage Account','manage_account.php')
})
})

</script>
```

admin.php

```
<!DOCTYPE html>
<html>
                <head>
                        <?php include('header.php') ?>
    <?php
    // session_start();
    // if(isset($_SESSION['login_id'])){
     //header('Location:home.php');
     //}
    ?>
                        <title>Admin Login |Bus Booking</title>
                </head>
<style>
    body {
background-image: url(./assets/img/bus.jpg);
height: 96vh;
background-position: center;
background-repeat: no-repeat;
background-size: cover;
}
</style>
```

```
<body id='login-body' class="bg-light">
                    <div class="card col-md-4 offset-md-4 mt-4">
         <div class="card-header-edge text-white">
            <strong>Login</strong>
         </div>
      <div class="card-body">
             <form id="login-frm">
               <div class="form-group">
                  <label>Username</label>
                  <input type="username" name="username" class="form-control">
               </div>
               <div class="form-group">
                  <label>Password</label>
                  <input type="password" name="password" class="form-control">
               </div>
               <div class="form-group text-right">
                  <button class="btn btn-primary btn-block" name="submit">Login</button>
               </div>
               <div class="form-group text-right">
                  <button type="submit"><a href="index.php">Go back to home</a></button>
               </div>

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


                        </body>


     <script>
      $(document).ready(function(){
         $('#login-frm').submit(function(e){
            e.preventDefault()
            $('#login-frm button').attr('disable',true)
            $('#login-frm button').html('Please wait...')
            $.ajax({
                url:'./login_auth.php',
                method:'POST',
                data:$(this).serialize(),
                error:err=>{
                    console.log(err)
                    alert('An error occured');
                    $('#login-frm button').removeAttr('disable')
                    $('#login-frm button').html('Login')
                },
                success:function(resp){
                    if(resp == 1){
                        location.replace('index.php?page=home')
                    }else{
                        alert("Incorrect username or password.")
                        $('#login-frm button').removeAttr('disable')
                        $('#login-frm button').html('Login')
                    }
                }
            })

        })
    })
</script>
</html>
```