**ABSTRACT**

**TOPIC: ONLINE TICKET BOOKING SYSTEM**

The ONLINE TICKET BOOKING SYSTEM is a web-based application that enables users to book transport tickets online. The system is designed to provide a convenient and efficient way of booking transport tickets for various modes of transportation such as buses, trains, and airplanes.

The system is built using a combination of HTML, CSS, JavaScript, Bootstrap, PHP, and MySQL. The front-end of the system is developed using HTML and CSS, while JavaScript is used to add interactivity to the user interface. The back-end of the system is developed using PHP, which handles the logic and functionality of the system. MySQL is used as the database management system to store and retrieve user data.

The system allows users to create an account and login to access the booking system. Once logged in, users can search for available transport options, select their preferred mode of transportation, and book tickets. The system also provides users with information about the schedule and availability of various modes of transportation.

The booking process involves selecting the mode of transportation, specifying the departure and arrival locations, selecting the travel dates, and specifying the number of tickets required. Users can then choose their preferred seats or cabin class and make payment using various payment methods.

The system also includes features such as ticket cancellation, and ticket rescheduling. Users can view their booking history and get notifications about their booking status and updates.

In conclusion, the transport ticket booking system is a web-based application that provides a convenient and efficient way of booking tickets. The system is built using a combination of HTML, CSS, JavaScript, Bootstrap, PHP, and MySQL and provides users with a seamless and user-friendly booking experience.

**TABLE OF CONTENTS**

**INDEX**

|  |  |  |
| --- | --- | --- |
| **SL.NO.** | **DESCRIPTION** | **PAGE NO.** |
| 1 | INTRODUCTION | 4 |
| **2** | SYSTEM STUDY   * 1. Existing System   2. Proposed System   3. Problem Definition and Project Description | 7  7  9 |
|  |  |
| **3** | SYSTEM ANALYSIS   * 1. Requirements Specification   2. Feasibility Study | 12  13 |
| 4 | SYSTEM DESIGN   * 1. Architectural Design   2. Data Flow Diagram   3. Data Dictionary   4. User Interface Design   5. Normalization | 16 |
|  | 17  21  30 |
|  | 36 |
|  |  |

**INTRODUCTION**

**1.INTRODUCTION**

An ONLINE TICKET BOOKING SYSTEM is a web application that allows users to book tickets for transportation services such as buses, trains, and flights. The system typically consists of a user-facing frontend and a backend database that stores information about routes, schedules, and bookings.

The frontend of the system is usually built using HTML, CSS, and JavaScript. HTML provides the structure of the web page, CSS is used for styling and layout, and JavaScript provides interactivity and dynamic behaviour.

The backend of the system is typically built using PHP and MySQL. PHP is a server-side scripting language that is used to handle user requests, process data, and communicate with the database. MySQL is a popular database management system that is used to store and retrieve data.

In an ONLINE TICKET BOOKING SYSTEM, the frontend allows users to search for routes, view schedules, and book tickets. The backend manages the business logic of the system, such as checking availability, processing payments, and updating the database.

Overall, The ONLINE TICKET BOOKING SYSTEM requires a combination of front-end and back-end technologies to provide a seamless and user-friendly experience for customers.

**SYSTEM STUDY**

**2. SYSTEM STUDY**

* 1. **EXISTING SYSTEM:**

The following are the disadvantages of the existing system:

* + - **Time**-**consuming**: One major disadvantage of offline transport ticket booking is that it can be time-consuming. Customers have to physically go to a ticket counter or agency to purchase their tickets, which can take a significant amount of time, especially if there are long lines or if the customer has to travel a long distance to reach the ticket counter.
    - **Higher** **prices**: Offline ticket booking may also be more expensive than online booking. This is because offline booking often involves additional fees, such as commission fees charged by agents or transportation companies, which can increase the overall cost of the ticket.
    - **Limited** **information**: Finally, offline ticket booking may also limit the amount of information available to customers. Online booking often provides more detailed information about the transportation, such as schedules, routes, and seat availability, which can help customers make more informed decisions when purchasing tickets.
  1. **PROPOSED SYSTEM:**

The existing manual system is replaced with a web application changing the way of maintaining the ONLINE TICKET BOOKING SYSTEM which proves to be beneficial, improves efficiency, and saves us time.

**The following are the advantages of the proposed system:**

* An ONLINE TICKET BOOKING SYSTEM is a software application that enables users to book transport tickets (such as bus, train, or flight tickets) online. The proposed system would allow users to search for transport options, view schedules and prices, and book tickets online.
* The system has the following features:
* **User registration and login**: Users would need to create an account and login to use the system.
* **Search :** Users should be able to search for available transport options based on their preferences such as date, time, departure and arrival locations, and transport type.
* **Booking and payment**: Once the user selects their preferred transport option, they should be able to book and pay for their ticket online.
* **Confirmation:** After booking and payment, the system should generate a confirmation email or ticket that the user can present to board the transport.
* **Cancellation and refund:** The system should also allow users to cancel their bookings and process refunds according to the cancellation policy.
* **Admin panel:** An admin panel should be available for transport companies to manage their schedules and prices, and to update information on the system.
* Overall, an ONLINE TICKET BOOKING SYSTEM should be user-friendly, secure, and efficient in terms of booking and payment processes, as well as having accurate and up-to-date information on schedules and prices.
  1. **PROBLEM DEFINITION AND PROJECT DESCRIPTION**
  + The transportation industry is one of the largest industries in the world, with millions of people using various modes of transportation every day. However, booking transportation can be a time-consuming and often frustrating experience. Customers often have to call or visit multiple service providers to find available transport options and to book their travel. This can lead to a lot of confusion and inconvenience.
  + To solve this problem, we propose an online transport booking system that allows customers to book their travel online. This system will streamline the booking process and make it easy and convenient for customers to find and book the transportation they need.

* + - * + Admin
        + User
        + Employee

**ADMIN MODULE**

* Manage transportation options, including adding new vehicles and modifying existing ones
* View and manage customer bookings
* View reports on bookings and revenue
* Manage the system's settings and configurations

**USER MODULE**

* Search for available transportation options based on their destination and date of travel
* View details of available transportation options, such as the type of vehicle, departure and arrival times, and fare
* Book their travel online and pay for their booking using a secure payment gateway
* Cancel or modify their booking if necessary
* View their booking history

**SYSTEM ANALYSIS**

**3.REQUIREMENT SPECIFICATION**

**3.1** **SOFTWARE REQUIREMENTS**

**HARDWARE REQUIREMENTS**

Processor : Intel or Ryzen

Ram : 4.00 GB (3.89 GB usable)

Hard disk : 40 GB

Monitor : 15’’mVGA monitor

Keyboard : 104 keys Keyboard

Mouse : Optical mouse

**SOFTWARE REQUIREMENTS**

Operating System : 64-bit OS

Local Host Server : Xampp server

Database Connectivity : PhpMyAdmin

Front End : Html and PHP

Back End : MYSQL

**3.2 FEASIBILITY STUDY**

A feasibility analysis is conducted to decide if the solution considered to meet the criteria is feasible and workable in the software. During the feasibility study, information such as resource A feasibility analysis is conducted to decide if the solution considered to meet the criteria is feasible and workable in the software. During the feasibility study, information such as resource availability, cost estimates for software production, advantages of the software to the enterprise after its development, and cost to be expended on its maintenance is determined. The feasibility study aims to ascertain why developing software is appealing to users, adaptable to change, and compliant with applicable requirements. The system has been tested for feasibility in the following points:

* + - Technical Feasibility
    - Operational Feasibility
    - Economic Feasibility

**Technical Feasibility:**

The technical feasibility of Online Transport Ticket Booking depends on various factors, such as the type of transport, the size of the service area, and the number of users. For example, online ticket booking for local public transport in a small town with a few hundred users may require a different level of technical infrastructure compared to online ticket booking for a national railway network with millions of users.

Some of the technical requirements for online transport ticket booking include:

1.A secure and reliable payment gateway to process payments and prevent fraud.

2.A user-friendly interface for customers to browse available services, select their preferred transport and book tickets.

3.Integration with the transport provider's existing reservation and scheduling systems to ensure accurate availability and pricing information.

4.Robust data management and analysis capabilities to monitor user behavior and system performance.

5.To implement an ONLINE TICKET BOOKING SYSTEM, a team of experienced software developers,

designers, and data analysts may be required. They will need to work closely with the transport provider to understand their requirements and integrate the online booking system with their existing operations.

Overall, the technical feasibility of online transport ticket booking has been well established and it is a proven technology that can help transport providers streamline their operations and offer a more convenient service to customers

**OPERATIONAL FEASIBILITY:**

The proposed system is beneficial only if it can be turned out into an information system that will meet the operating requirements of the organization. The extent to which the required software completes a sequence of steps to address the challenges and requirements of the developer and users respectively is measured by operational viability. The following are the operations carried out by operational feasibility:

* + Determines whether sufficient support for the organization is provided by the users.
  + Ensures proper working of the system if it is being developed and implemented.
  + Checks whether there will be any resistance from the users that will ruin the possible benefits of the application

This ONLINE TICKET BOOKING SYSTEM would ensure the optimal utilization of computer resources and would help in the improvement of performance status.

**ECONOMIC FEASIBILITY:**

A system that can be developed technically and that will be used if installed must still be a good investment for the organization. Economic feasibility needs to consider the expenses made on purchasing, such as hardware purchasing and required activities to carry out software development. It is also necessary to consider the benefits that can be achieved by developing the software. Software is economically feasible when it focuses on the issues listed below.

* + - The expense incurred on software development for achieving long-term gains for an organization.
    - Expenses required to conduct elicitation and requirements analysis
    - Hardware and software cost, development team, and training cost.

This system is economically feasible. Since this system is developed using the existing resources and technologies, there is nominal expenditure which ensures the economic feasibility of the system.

**SYSTEM DESIGN**

**4.1 ARCHITECTURAL DESIGN:**

Online Transport management System (Buses,Flights,Trains)

Ticket Booking (add,view,cancel)

Add,Update profile

Manage Trains

Train Tickets

Add,manage Trains

Add,Update profile

Manage Transports(add,delte,view)

Manage payments

Manage Users

Manage Employees

Manage Routes

Users

Employee

Admin

Flights

Trains

Buses

Payment Ticket

Manage Transport Ticket

(Check booking Tickets)

Logout

**4.2 Data Flow Diagram:**

DataBase

Bus

**DFD Level 0**

User

Train,flight,bus

Employee train

Admin

train,flight,Bus

Database FLight

DataBase

Train

D

DataBase

FLight

DFD Level 1

route

Bus Data Base:

bus

Admin

booking

payment

user

Bus Data Base:

booking

User

bus

payment

user

Train Data Base:

Orrs\_admin

Admin

Orrs\_train

Orrs\_passenger

Orrs\_employee

Orrs\_train\_tickets

Orrs\_employee

Orrs\_train

Employee

Orrs\_passenger

Orrs\_train tickets

Orrs\_passenger

User

Orrs\_train

Orrs\_train\_ticket

Flight Data Base:

customer

Admin

Flight\_details

Payment\_details

ticket\_details

customer

User

Flight\_details

Ticket\_Details

Payment\_details

**4.3 DATA DICTIONARY**

**Table** **Name**: admin (Bus Booking)

**Purpose**: To store the admin name and password

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **Field Name** | **Data Type** | **Size** | **Constraint** | **Description** |
| 1 | id | Int | 11 | Primary key | Admin id |
| 2 | username | Varchar | 50 | Not Null | Admin name |
| 3 | password | Varchar | 50 | Not Null | Admin password |

**Table** **Name**: booking (Bus Booking)

**Purpose**: To store the Ticket Booking Passengers details.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **Field Name** | **Data Type** | **Size** | **Constraint** | **Description** |
| 1 | Id | int | 11 | Primary key | Passenger id |
| 2 | Passenger\_name | varchar | 50 | Not Null | Passenger name |
| 3 | telephone | int | 10 | Not Null | telephone |
| 4 | email | varchar | 50 | Not Null | Email |
| 5 | boarding\_place | varchar | 50 | Not Null | Start place |
| 6 | Yout\_destination | varchar | 50 | Not Null | Ending place |

**Table** **Name**: bus

**Purpose**: To store the details of buses

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **Field Name** | **Data Type** | **Size** | **Constraint** | **Description** |
| 1 | id | int | 11 | Primary Key | Bus id |
| 2 | Bus\_Name | varchar | 50 | Not Null | Bus name |
| 3 | Tel | int | 10 | Not Null | Phone number |

**Table** **Name**: payment

**Purpose**: To store the payment details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **Field Name** | **Data Type** | **Size** | **Constraint** | **Description** |
| 1 | Id | int | 11 | Primary Key | User Id |
| 2 | amount | float | 10 | Not Null | Amount |
| 3 | name | varchar | 50 | Not Null | User name |
| 4 | email | varchar | 50 | Not Null | Email |
| 5 | address | varchar | 50 | Not Null | User address |
| 6 | city | varchar | 50 | Not Null | City |
| 7 | state | varchar | 50 | Not Null | State |
| 8 | zip\_code | int | 6 | Not Null | Zip code |
| 9 | card\_name | Varchar | 50 | Not Null | Card name |
| 10 | card\_number | int | 16 | Not Null | Card number |
| 11 | exp\_month | Varchar | 20 | Not Null | Exp month |
| 12 | exp\_year | Varchar | 20 | Not Null | Exp year |
| 13 | cvv | int | 4 | Not Null | cvv |

**Table** **Name**: route (Bus Booking)

**Purpose**: To store the route details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **Field Name** | **Date Type** | **Size** | **Constraint** | **Description** |
| 1 | Id | Int | 11 | Primary Key | User Id |
| 2 | city | varchar | 50 | Not Null | Starting place |
| 3 | Destination | varchar | 50 | Not Null | Ending place |
| 4 | Bus\_name | varchar | 50 | Not Null | Bus name |
| 5 | Departure\_date | Date |  | Not Null | Travel date |
| 6 | Departure\_time | Time | 6 | Not Null | starting time |
| 7 | Cost | varchar | 50 | Not Null | Price of bus |

**Table** **Name**: Users (Bus Booking)

**Purpose**: To store the User details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.NO** | **Field Name** | **Data Type** | **Size** | **Constraint** | **Description** |
| 1 | Id | Int | 11 | Primary Key | Number of user |
| 2 | User\_id | Int | 11 | Unique | Trans side id |
| 3 | First\_Name | Varchar | 50 | Not Null | First name |
| 4 | Last\_Name | Varchar | 50 | Not Null | Last name |
| 5 | Username | Varchar | 50 | Not Null | User name |
| 6 | Email | Varchar | 50 | Not Null | Email |
| 7 | Password | varchar | 50 | Not Null | Password |

**Train tables**

**Table** **Name**: Orrs\_admin ( Train)

**Purpose** :To store the Admin details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **Field Name** | **Data Type** | **Size** | **Constraint** | **Description** |
| 1 | Admin\_id | Int | 10 | Primary Key | Admin number |
| 2 | Admin\_fname | Varchar | 50 | Not Null | First name |
| 3 | Admin\_lname | varchar | 50 | Not Null | Last name |
| 4 | Admin\_email | Varchar | 50 | Not Null | Email |
| 5 | Admin\_uname | varchar | 50 | Not Null | User name |
| 6 | Admin\_pwd | Varchar | 50 | Not Null | Password |
| 7 | Admin\_dpic | varchar | 50 | Not Null | image |

**Table** **Name**: Orrs\_train (Train)

**Purpose**: To store the User details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **Field Name** | **Data Type** | **Size** | **Constraint** | **Description** |
| 1 | Id | Int | 20 | Primary Key | Number of Train |
| 2 | name | varchar | 50 | Not Null | Train name |
| 3 | Route | varchar | 50 | Not Null | Route of train |
| 4 | Current | Varchar | 50 | Not Null | Start place |
| 5 | Destination | Varchar | 50 | Not Null | End place |
| 6 | Time | time |  | Not Null | Time |
| 7 | passenger | Varchar | 50 | Not Null | Passenger size |
| 8 | Number | varchar | 10 | Not Null | Train number |
| 9 | fare | int | 10 | Not Null | Rate |

**Table** **Name**: Orrs\_User (Train)

**Purpose**: To store the user details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.NO** | **Field Name** | **Data Type** | **size** | **Constraint** | **Description** |
| 1 | Pass\_id | Int | 10 | Primary Key | Passenger id |
| 2 | Pass\_fname | varchar | 50 | Not Null | First name |
| 3 | Pass\_lname | varchar | 50 | Not Null | Last name |
| 4 | Pass\_phone | Number | 50 | Not Null | Phone |
| 5 | Pass\_addr | Varchar | 50 | Not Null | Address |
| 6 | Pass\_email | Varchar | 50 | Not Null | Email |
| 7 | Pass\_pwd | Varchar | 50 | Not Null | Password |
| 8 | Pass\_dpic | Varchar |  | Not Null | Image |
| 9 | Pass\_username | Varchar | 50 | Not Null | Username |
| 10 | Pass\_train\_number | Varchar | 50 | Not Null | Train number |
| 11 | pass\_train\_name | Varchar | 50 | Not Null | Train name |
| 12 | Pass\_dep\_station | Varchar | 50 | Not Null | Start place |
| 13 | Pass\_dep\_time | Varchar | 50 | Not Null | Time |
| 14 | Pass\_arr\_station | Varchar | 50 | Not Null | End place |
| 15 | Pass\_train\_fare | Varchar | 50 | Not Null | price |
| 16 | Pass\_fare\_payment\_code | varchar | 50 | Not Null | Price code |

**Table** **Name**: Orrs\_train\_ticket (Train)

**Purpose**: To store booking ticket details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.NO** | **Field Name** | **Data Type** | **Size** | **Constraint** | **Description** |
| 1 | Ticket\_id | int | 20 | Primary Key | Number of ticket |
| 2 | Pass\_name | varchar | 50 | Not Null | Name |
| 3 | Pass\_email | varchar | 50 | Not Null | Mail id |
| 4 | Pass\_addr | varchar | 50 | Not Null | Address |
| 5 | Train\_name | varchar | 50 | Not Null | Name of trajn |
| 6 | Train\_no | varchar | 50 | Not Null | Train no |
| 7 | Train\_dep\_state | varchar | 50 | Not Null | Start place |
| 8 | Train\_dep\_arr | varchar | 50 | Not Null | End place |
| 9 | Train\_fare | varchar | 50 | Not Null | fare |
| 10 | fare\_payment\_code | varchar | 50 | Not Null | Payment code |
| 11 | confirmtion | varchar | 50 | Not Null | confirmation |

**Table** **Name**: Orrs\_password\_reset (Train)

**Purpose**: To store employee resets passwords

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.NO** | **Field Name** | **Data Type** | **Size** | **Constraint** | **Description** |
| 1 | Pwd\_id | Int | 20 | Primary Key | Password |
| 2 | Email | Varchar | 50 | Not null | Email |
| 3 | status | varchar | 50 | Not null | Approved  Or  Removed |

**Table** **Name**: Orrs\_employee (Train)

**Purpose**: To store employee details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **Field Name** | **Data Type** | **Size** | **Constraint** | **Description** |
| 1 | Emp\_id | Int | 20 | Primary Key | Emp id |
| 2 | Emp\_fname | varchar | 50 | Not Null | First name |
| 3 | Emp\_lname | varchar | 50 | Not Null | Last name |
| 4 | Emp\_nat\_idno | int | 50 | Not Null | number |
| 5 | Emp\_phone | Int | 10 | Not Null | Mobile number |
| 6 | Emp\_addr | varchar | 50 | Not Null | Address |
| 7 | Emp\_uname | varchar | 50 | Not Null | User name |
| 8 | Emp\_email | varchar | 50 | Not Null | Email |
| 9 | Emp\_pwd | varchar | 50 | Not Null | Password |
| 10 | Emp\_dpic | varchar | 50 | Not Null | Image |
| 11 | Emp\_dept | varchar | 50 | Not Null | department |

**Flight**

**Table** **Name**: admin

**Purpose**: To store admin details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **Field Name** | **Data Type** | **Size** | **Constraint** | **Description** |
| 1 | Admin id | Varchar | 20 | Primary key | Admin id |
| 2 | Pwd | Varchar | 30 | Not Null | Password |
| 3 | Staff\_id | Varchar | 20 | Not Null | Staff id |
| 4 | Name | Varchar | 20 | Not Null | User name |
| 5 | Email | Varchar | 25 | Not Null | Email |

**Table** **Name**: Customer

**Purpose**: To store Customer details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **Field Name** | **Datatype** | **Size** | **Constraint** | **Description** |
| **1** | **Customer\_id** | **varchar** | **20** | **Primary Key** | **Customer id** |
| **2** | **Pwd** | **Varchar** | **20** | **Not null** | **password** |
| **3** | **Name** | **Varchar** | **20** | **Not null** | **Username** |
| **4** | **Email** | **Varchar** | **20** | **Not null** | **Email** |
| **5** | **Phone\_no** | **Varchar** | **20** | **Not Null** | **Phone no** |
| **6** | **Address** | **varchar** | **20** | **Not Null** | **address** |

**Table** **Name**: Flight\_details

**Purpose** :To store Flights details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **Field Name** | **Data Type** | **Size** | **Constraint** | **Description** |
| 1 | Flight\_no | Varchar | 10 | Primary Key | Flight no |
| 2 | From\_city | Varchar | 20 | Not Null | Start city |
| 3 | To\_city | Varchar | 20 | Not Null | End city |
| 4 | Depature\_date | Date |  | Primary Key | Starting date |
| 5 | Arrival\_date | Date |  | Not Null | Ending date |
| 6 | Departure\_time | Time |  | Not Null | Pick up time |
| 7 | Arrival\_time | Time |  | Not Null | Reach time |
| 8 | Seats\_economy | Int | 5 | Not Null | Seats economy |
| 9 | Seat\_business | Int | 5 | Not Null | Seat business |
| 10 | Price\_economy | Int | 10 | Not Null | Price economy |
| 11 | Price\_business | Int | 10 | Not Null | Price business |
| 12 | Jet\_id | varchar | 10 | Not Null | Jet id |

**Table** **Name**: Frequent\_Flight\_details

**Purpose**: To store Frequent Flights details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **Field Name** | **Data Type** | **Size** | **Constraint** | **Description** |
| 1 | Frequent\_Flier\_no | Varchar | 20 | Primary Key | Frequent Flier\_no |
| 2 | Customer\_id | Varchar | 20 | Not Null | Customer\_id |
| 3 | mileage | int | 10 | Not Null | mileage |

**Table** **Name**:jet\_details

**Purpose**:To store jet\_ details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **Field Name** | **Data Type** | **Size** | **Constriant** | **Description** |
| 1 | Jet\_id | Varchar | 10 | Primary Key | Jet\_id |
| 2 | Jet\_type | Varchar | 10 | Not Null | Jet\_type |
| 3 | Total\_Capacity | Int | 5 | Not Null | Total\_capacity |
| 4 | active | varchar | 5 | Not Null | active |

**Table** **Name**: Payment\_details

**Purpose**: To store Payment\_details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.NO** | **Field Name** | **Data Type** | **Size** | **Constraint** | **Description** |
| 1 | Payment\_id | Varchar | 20 | Primary Key | Payment\_id |
| 2 | Pnr | Varchar | 15 | Primary Key | Pnr |
| 3 | Payment\_date | Date |  | Not Null | Payment\_date |
| 4 | Payment\_amount | Int | 6 | Not Null | Payment\_amount |
| 5 | Payment\_mode | varchar | 15 | Not Null | Payment\_mode |

**Table** **Name**: passenger

**Purpose**: To store Passengers\_details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.NO** | **Field Name** | **Data TYpe** | **Size** | **Constraint** | **Description** |
| 1 | Passenger\_id | Int | 10 | Primary Key | Passenger id |
| 2 | Pnr | Varchar | 15 | Primary key | Unique pnr |
| 3 | Name | Varchar | 20 | Not Null | User name |
| 4 | Age | Int | 3 | Not Null | Age |
| 5 | Gender | Varchar | 8 | Not Null | gender |
| 6 | Meal choice | varchar | 5 | Not Null | Meals |
| 7 | Frequent\_flier\_no | varrchar | 20 | Not Null | Frequent flier no |

**Table** **Name**: ticket\_details

**Purpose**: To store ticket\_details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **Field Name** | **Data Type** | **Size** | **Constraint** | **Description** |
| 1 | Pnr | Varchar | 5 | Primary Key | Pnr |
| 2 | Date\_of\_reservation | Date |  | Not Null | Date of reservation |
| 3 | Flight\_no | Varchar | 10 | Unique Key | Flight number |
| 4 | Journey\_date | Date |  | Unique Key | Journey date |
| 5 | Class | Varchar | 10 | Not Null | class |
| 6 | Booking\_status | Varchar | 20 | Not Null | Booking status |
| 7 | No\_of\_passengers | int | 5 | Not Null | Number of passenger |
| 8 | lounge\_access | Varchar | 5 | Not Null | Lounge\_access |
| 9 | Priority\_checkin | Varchar | 5 | Not Null | Priority check in |
| 10 | Insurance | varchar | 5 | Not Null | Insurance |
| 11 | Payment\_id | Varchar | 20 | Not Null | Payment id |
| 12 | Customer\_id | varchar | 20 | Uniique Key | Customer id |

**4.4 USER INTERFACE DESIGN**

**Home page:**

**Home Buses Flights Trains User Map**

Buses:

**IRCTC OFFICIAL Home Help Login About US Services**

**User Login:**

**Login For IRCTC**

**Username**

****

**Passsword**

****

**Sign Up**

**Login**

**Admin Login:**

**Admin Login**

**Username**

****

**Passsword**

****

****

**Register Form:**

**SIGN UP**

|  |
| --- |
| First Name Last Name  Email-Address  Username  Password  Confirm Password  REGISTER NOW |

**User**

|  |
| --- |
| image  Ticket Booking  Profile  Cancel  Logout |

**Booking Form**

|  |
| --- |
| Passenger Name  Telephone  E-mail  Board Place  Your destination  Booking NOW |

**Payment Form**

|  |
| --- |
| Amount You Pay: Card Name:  Name : Credit Card Number:  Email : Exp Month:  Address : Exp Year  City:  cvv  State : Zip Code:  Pay now |

\

**Update Bus Route Form**

|  |
| --- |
| Id :  City:  Destination:  Departure Date Bus Name:  Depature Time Cost  Update Route Now |

**Update Buses**

|  |
| --- |
| Id:  Bus\_Name:  Telephone:  Update Bus |

**Update Bus Passenger Booking**

|  |
| --- |
| id :  Passenger\_name  Telephone  Boarding place Destination    Email  Update Passenger Now |

**Flights Home Page:**

|  |
| --- |
| Home Main page Book Tickets Check PNR Login Sign Up |

**Flight Login:**

|  |
| --- |
| Username:  Password:    User Type:  Customer Admin  Login |

**Search Flights:**

|  |
| --- |
| Enter the origin Enter the Destination  To:    From:  Enter the Departure Date Enter the No of Passengers    Eg.1  dd-mm-yyyy  Enter the Class  Search for Available Flights |

**View Booked Flights Tickets**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| PNR | DATE OF RESERVATION | FLIGHT  No: | JOURNEY  DATE | CLASS | BOOKING  STATUS | NO OF  PASSENGEERS | PAYMENT  ID |
| 2931786 | 2023-05-05 | IRCTC10 1 | 2023-05-06 | BUSINESS | CONFIRMED | 1 | 6574738 |
| 9412543 | 2023-05-06 | IRCTC102 | 2023-05-07 | ECONOMY | CONFIRMED | 2 | 8769574 |

**Print Ticket:**

|  |
| --- |
| Enter Your PNR Number  Print |

**Cancel Tickets:**

|  |
| --- |
| Enter the PNR:    Cancel Ticket |

**Admin:**

**Add Flight Schedule Details**

|  |
| --- |
| Flight Number  Origin Destination    Departure Date Arrival Date    dd-mm-yyyy  dd-mm-yyyy  Departure Time Arrival Time    Number Of Seats in Economy Class Number Of Seats In Business Class    Ticket Price(Economy Class) Ticket Price(Business Class)    Jet Id    Submit |

**Delete The Flight Schedule:**

|  |
| --- |
| Enter a valid Flight No: Enter the Departure Date    Delete |

**Train:**

|  |
| --- |
| menu      Home  Main Page  Make Reservation  Employee Login  Admin Login  Menu |

**Login:**

|  |
| --- |
| IRCTC  LOG IN  REGISTER  PASSWORD  GMAIL |

**Book Train**

|  |
| --- |
| nalla  First Name    perumal  Last Name  9094077295  Phone Number    South street  Address  IRCTC101  Train Number    Chennai Express  Train Name  Chennai  Departure  Bangalore  Arrival  7.00 A.m  Departure Time  900  Train Fare    Cancel  Book Train |

**Admin:**

**Add Train**

|  |
| --- |
| Train Number  Train Name  Departure  Arrival  Departure Time  Number of Passenger  Train Fare    Cancel  Add Train |

**Maintain Train**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Train  Number | Train | Route | Departure | Arrival | Dep Time | Fare | Total  Passenger | Action |
| IRCTC101 | Chennai  Express | Chennai  -  Bangalore | Chennai | Bangalore | 7.00 A.m | $1200 | 200 | Update  Delete |
| IRCTC102 | Bangalore  Express | Bangalore  -  Chennai | Bangalore | Chennai | 8.00  A.m | $1400 | 300 | Update  Delete |

**Payment Conform**:

|  |
| --- |
| Passenger Name  Email  Address  Booked Train Number  Booked Train Name  Departure  Arrival  Train Fare  Payment Code    Confirmation  Cancel  Confirm Payment |

**Pending Ticket:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Passenger | Email | Address | Train  Number | Departure | Arrival | Fare | Payment  Code | Action |
| Nalla  Perumal | nalla@mail  .com | North  Street | IRCTC101 | Chennai | Bangalore | $900 | 12345 | Approve |

**4.5 NORMALIZATION**

Normalization is the process of organizing the data in the database. Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate undesirable characteristics like Insertion, Update, and Deletion Anomalies. Normalization divides the larger table into the smaller table and links them using a relationship. The normal form is used to reduce redundancy from the database table.

**First Normal Form (1NF)**

For a table to be in the First Normal Form, it should follow the following 4 rules:

* + - It should only have single(atomic) valued attributes/columns
    - Values stored in a column should be of the same domain
    - All the columns in a table should have unique names
    - And the order in which data is stored does not matter

**Second Normal Form (2NF)**

For a table to be in the Second Normal Form,

* It should be in the First Normal form
* And, it should not have Partial Dependency. Partial Dependency occurs when a non-prime attribute is functionally dependent on part of a candidate key

**Third Normal Form (3NF)**

A table is said to be in the Third Normal Form when,

* + - It is in the Second Normal form.
    - And, it doesn't have Transitive Dependency

**Boyce and Codd Normal Form (BCNF)**

Boyce and Codd Normal Form is a higher version of the Third Normal form. This form deals with a certain type of anomaly that is not handled by 3NF. A 3NF table that does not have multiple overlapping candidate keys is said to be in BCNF. For a table to be in BCNF, the following conditions must be satisfied:

* R must be in 3rd Normal Form
* For each functional dependency ( X → Y ), X should be a super Key.

**Fourth Normal Form (4NF)**

A table is said to be in the Fourth Normal Form when,

* It is in the Boyce-Codd Normal Form.
* And, it doesn't have Multi-Valued Dependency

**SYSTEM TESTING**

**5.1 TYPES OF TESTING: 5. SYSTEM TESTING**

Testing is the major quality measure technique employed during software development process. After the coding phase, computer programs are available that can be executed for testing purpose. Testing not only has to uncover errors introduced during coding, but also locates errors committed during the previous phase. Thus the aim of testing is to uncover requirements, design or coding errors in the program.

The basic types of testing are:

* + - Unit testing
    - Integration testing
    - Validation testing
    - Output testing
    - User Acceptance testing

### **UNIT TESTING**

This is the first level of testing. In this different modules are tested against the specification produced during the design of the modules. Unit testing is done for the verification of code produced during the coding of single program module in an isolated environment. Unit testing first focuses on the modules independently of one another to locate errors.

### **INTEGRATION TESTING**

After the modules are tested individually, they must be tested in combination with each other to be sure that the interfaces are correct. This is known as integration testing. Hence, we consider interfacing of various modules. Thus in the integration testing step, all the errors uncovered are corrected for the next testing steps.

### **VALIDATION TESTING**

Validation testing gives the final assurances that the software meets all functional, behavioural and performance requirements. The software is completely assembled as a package. Validation succeeds when the software functions in a manner in which the user expects. Validation refers to the process of using software in a live environment in order to find errors. If the password was given wrongly by customers then it shows the check password error. Then if the username and password are not typed correct then it shows check username and password error. In the field, medicine quantity if the customers type any character other than numbers then it displays a warning message to give only numbers.

### **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 generated or considered in to two ways, one is on screen and another is printed format. The output format on the screen is found to be correct as the format was designed in the system design phase according to the user needs. If the user gives their correct username and password then it logins to the corresponding page.

### **USER ACCEPTANCE TESTING**

User acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes where required.

**5.2 TYPES OF VALIDATIONS**

A Validation control enables to validate an input and display an error message if necessary.

Validation types are given below

### **REQUIRED FIELD VALIDATION**

The Required Field Validator is actually very simple, and yet very useful. One can use it to make sure that the user has entered something in a Text Box control. In every form required field validator is assigned to fulfil all the specification.

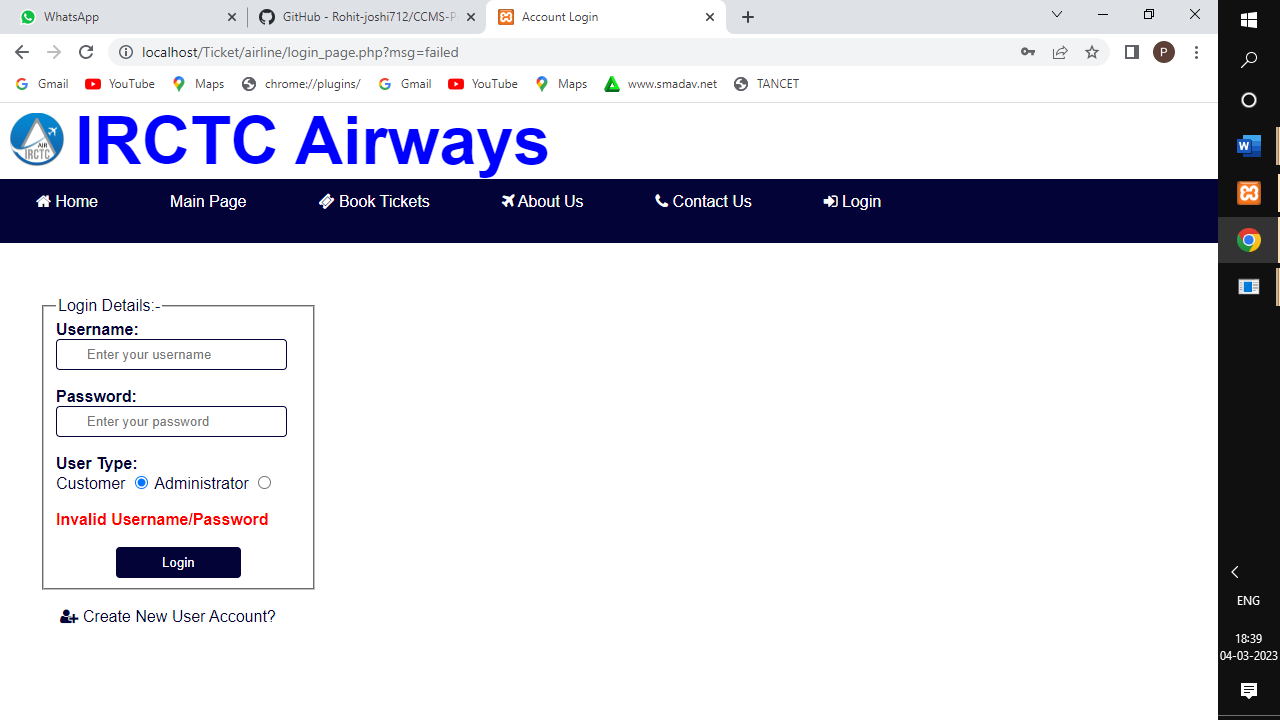
### **REGULAR EXPRESSION VALIDATION**

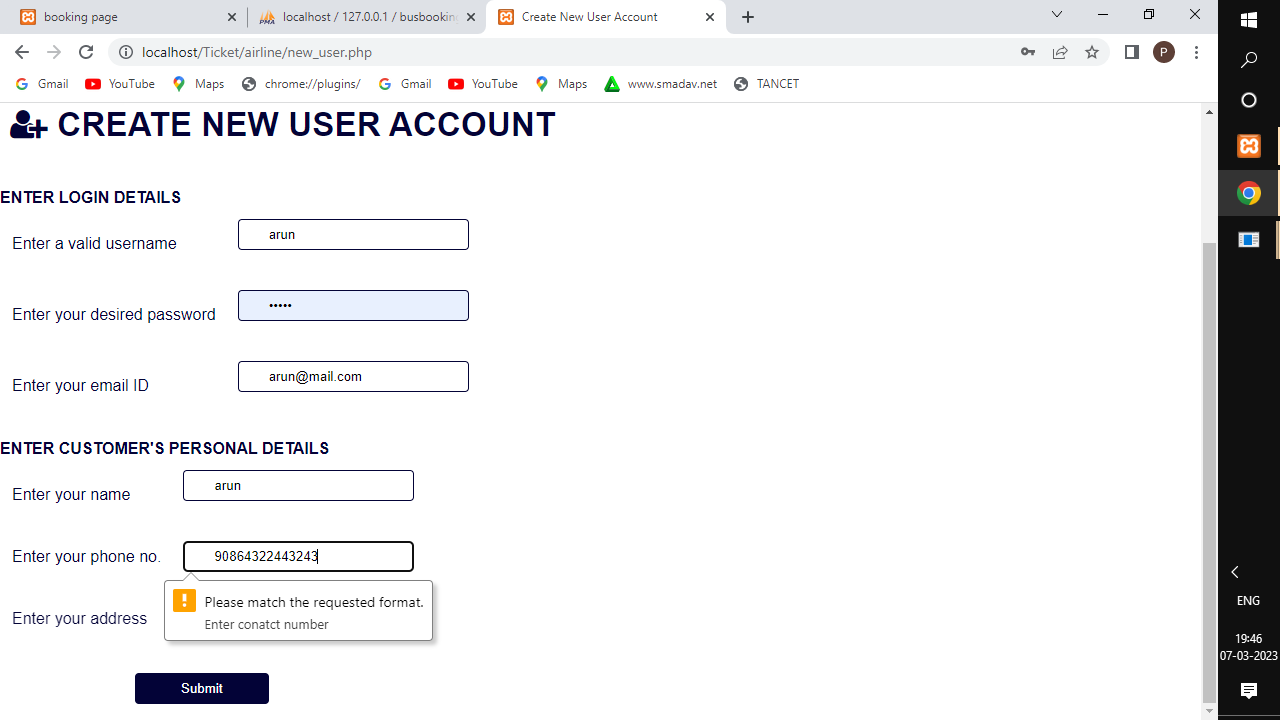
Regular Expression Validator is one of the most useful validators, because it can be used to check the validity of any kind of string. In this project regular expression validator is assigned for email checking entries.

### **RANGE VALIDATION**

The Range Validator does exactly what the name implies; it makes sure that the user input is within a specified range. It is used to validate numbers, strings and dates, which can make it useful in a bunch of cases. In this project, range validator is assigned for checking phone numbers.

**5.3 ERROR MESSAGE**





USER MANUAL

**6. USER MANUAL**

### **INSTALLATION MANUAL**

* + - Download the required software.
    - Unzip the files.
    - Install all the software one by one.
    - Install MYSQL.
    - Install Apache.
    - Open the web browser and type ‘localhost’ as the address.
    - Unzip the PHP file and copy it to the C disk.
    - Find the file named ‘httpd.conf” at ‘C:
    - Open the htdocs folder at ‘C:
    - Open your web browser and enter ‘localhost/test.php’ as the address.

If the browser shows the PHP version and other things, it means the PHP is successfully installed.

### **OPERATIONAL MANUAL**

**User Side:**

**Registration and Login:**

* Users can create an account by providing their personal information such as name, email, password, and contact details.
* Users can log in to their account by providing their email and password.

**Search for Tickets:**

* Users can search for tickets by providing the source and destination city, travel date, and

number of passengers.

* The system should display the available tickets along with the fare, departure and arrival time, and other details.

**Select and Book Tickets:**

* Users can select the desired ticket and book it by providing the passenger details such as name, age, gender, and contact information.
* Users can choose the payment method and pay for the ticket.

**Ticket Management:**

* Users can view their booking history and print the tickets.
* Users can cancel their bookings subject to cancellation policies.

**Admin Side:**

**Login:**

* The first step for an admin is to log in to the system with their username and password

**Manage Bookings**:

* The admin can view, edit, and delete the bookings made by the customers.
* The admin can also search for a particular booking using the booking ID or customer name.

**Manage Shows:**

* The admin can add new shows or modify existing ones.
* They can set the show name, date, time, venue, and price.

**Manage Users:**

* The admin can view and manage the user accounts.
* They can add new users, edit existing user profiles

**Manage Payments**:

* The admin can view the payment status of each booking

**Logout:**

* Finally, the admin can log out of the system after completing their tasks.

**SYSTEM IMPLEMENTATION**

## **7. SYSTEM IMPLEMENTATION**

## **7.**1 SPECIAL FEATURES OF THE LANGUAGES PHP

Hypertext Preprocessor (or simply PHP) is a general-purpose programming language originally designed for web development. It was originally created by Rasmus lerdorf in 1994; the PHP reference implementation is now produced by the PHP group. PHP originally stood for personal home page, but it now stands for the recursive initialism. PHP code may be executed with a command line interface (CLI), embedded into HTML code, or it can be used in combination with various web template systems, web content management systems, and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in a web server or as a common gateway interface executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page.PHP can be used for many programming tasks outside of the web context, such as standalone graphical applications and robotic drone control

The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP license. PHP has been widely ported and can be deployed on most web servers on almost every operation system and platform, free of charge.

The PHP language evolved without a written formal specification or standard until 2014, with the original implementation acting as the de facto standard which other implementations aimed to follow PHP specification.

### **UNIQUE FEATURES OF PHP**

**Simple, Familiar and ease of use:** It is popularly known for its simplicity, familiarity and easy to learn the language as the syntax is similar to that of ‘C’ or Pascal language. So the language is very logical and well organized general-purpose programming language. Even people with a normal programming background can easily understand and capture the use of language. PHP is very advantageous for new users as its a very reliable, fluent, organized, clean, demandable and efficient. The main strength of PHP is the availability of rich pre-defined functions.

**Loosely typed language:** PHP encourages the use of variables without declaring its data types. So this is taken care at the execution time depending on the value assigned to the variable. Even the variable name can be changed dynamically.

**Flexibility:** PHP is known for its flexibility and embedded nature as it can be well integrated with HTML, XML, Java script and many more. PHP can run on multiple operating systems like Windows, Unix, Mac OS, Linux, etc. The PHP scripts can easily run on any device like laptops, mobiles, tablets, and computer. It is very comfortably integrated with various Databases. Desktop applications are created using advanced PHP features. The executable PHP can also be run on command-line as well as directly on the machine. Heavyweight applications can be created without a server or browser. It also acts as an excellent interface with relational databases.

**Cross-platform compatibility:** PHP is multi-platform and known for its portability as it can run on any operating System and windows environments. The most common are XAMPP and LAMP. As PHP is platform-independent, it’s very easy to integrate with various databases and other technologies without re-implementation. It effectively saves a lot of energy, time and money.

**Open Source:** All PHP frameworks are open sources, No payment is required for the users and its completely free. User can just download PHP and start using for their applications or projects. Even in companies, the total cost is reduced for software development providing more reliability and flexibility. It supports a popular range of databases like MySQL, SQLite, Oracle, Sybase, Informix, and PostgreSQL

**Error reporting and exceptions:** PHP supports more errors reporting constants to generate errors and relevant warning at runtime. For example**,** E\_ERROR, E\_WARNING, E\_PARSE, E\_STRICT**.**

**Active community support:** PHP is very rich with many diverse online community developers to help beginners for web-based applications. These worldwide volunteers contribute many features as well as new versions for PHP libraries. Even they contribute a translation in different languages to help out programmers. There is a bundle of third-party open-source libraries which provide basic functionalities. Even the documentation given by the official site helps in implementing new features providing access to a variety of creative imagination.

**Maintenance:** When dealing with big projects, maintenance of code is also an important aspect of the web development process. There are many PHP frameworks for example MVC (Model View Controller) which makes development and maintenance of code easier. Files belonging to the different module are maintained separately.

**PDO Class:** PHP Data Objects are created by PDO class which gives a good abstraction layer for database drivers. The PDO Classes are enriched with functions which are database independent. It means the same functions are used for similar actions for different databases without re-development as long as it supports PDO. In this way, the application becomes more portable saving lot of time and effort. Use of PDO helps the application from SQL injection attacks.

**Memory and CPU usage information:** PHP can provide memory usage information from functions like memory\_get\_usage() or memory\_get\_peak\_usage(), which can help the developers optimize their code. In the similar way, the CPU power consumed by any script can be retrieved for further optimization.

**Object oriented features:** PHP supports object-oriented programming features, resulting in increased speed and introducing added features like data encapsulation and inheritance at many levels.

### MySql

MySql is an open source relational database management system (RDMBS). It’s name is a combination of “My”, the name of co-founder Michael Widenius’s daughter, and “SQL”, the abbreviation for structured Query Language.

MySql is free and open-source software under the terms of the GNU general public license, and is also available under a variety of proprietary licenses. MySql was owned and sponsored by the Swedish company MySql AB, which was bought by sun Microsystems. In 2010 when oracle acquired sun, Widenius forked the open-source MySql project to create MariaDB.

MySql is a component of the lamp web application software stack, which is an acronym for Linux, Apache, MySql, and Perl. MySql is used by many database-driven web applications, including drupal, joomla, phpBB, and Word Press. MySql is also used by many popular websites, including Google, face book, Twitter and You Tube. The main features of MySql includes

**Easy to use**: MySQL is easy to use. We have to get only the basic knowledge of SQL. We can build and interact with MySQL by using only a few simple SQL statements.

**It is secure**: MySQL consists of a solid data security layer that protects sensitive data from intruders. Also, passwords are encrypted in MySQL.

**Client/ Server Architecture**: MySQL follows the working of a client/server architecture. There is a database server (MySQL) and arbitrarily many clients (application programs), which communicate with the server; that is, they can query data, save changes, etc.

**Free to download**: MySQL is free to use so that we can download it from MySQL official website without any cost.

**Compatible on many operating systems**: MySQL is compatible to run on many operating systems, like Novell NetWare, Windows\* Linux\*, many varieties of UNIX\* (such as Sun\* Solaris\*, AIX, and DEC\* UNIX), OS/2, FreeBSD\*, and others. MySQL also provides a facility that the clients can run on the same computer as the server or on another computer (communication via a local network or the Internet).

**FUTURE ENHANCEMENT**

**8. FUTURE ENHANCEMENT**

* Allow users to view and select their preferred seats and amenities.
* Provide chatbot support for quick and efficient customer service.
* Enable users to easily search and compare prices and schedules.
* Enable users to provide feedback on the accessibility of transport services.
* To implement a range of travel packages and deals to users, including bundled travel and accommodation options.

**CONCLUSION**

**9. CONCLUSION**

The arrival of technology and the escalating use of internet have reduced our burden of work. The Online Ticket booking can be booked and the status of those bookings can be viewed anywhere and at anytime by the user. All the processes are transparent to the admin. In comparison with the manual system, the benefit under a computer system is considerable in saving man power, working hours and efforts.

Various validation techniques have been used to implement accuracy of data in all formats of input. Updating of information becomes so easier. The system has produced all the reports required for the management. It is concluded that the application works well and satisfies the users.

The application is tested very well and errors are properly debugged. The project works according to the restrictions provided to the users respectively. Further enhancements can be made to the application, so that the “Online Ticket Booking System” can be functioned more efficiently than the present one.

**BIBLIOGRAHY**

**10. BIBLIOGRAPHY**

### **BOOK REFERENCES**

* + - Robin Nixon, “Learning **PHP**, MySQL, JavaScript, and CSS: A Step-by-Step Guide to Creating Dynamic Websites”, Second Edition, O’Reilly Publications, 2012
    - Thomas Powell, “The Complete Reference: HTML and XHTML”, Fifth Edition, Tata McGraw Hill Publication, 2010
    - Paul Hudson, “PHP in a Nutshell”, O’Reilly Publications, 2005
    - Roger S. Pressman, “Software Engineering: A Practitioner Approach”, Eighth Edition, McGraw Hill, 2015
    - William E, Perry, “Effective Methods for Software Testing”, Third Edition, John Wiley, 2015

### **WEBSITE REFERENCES**

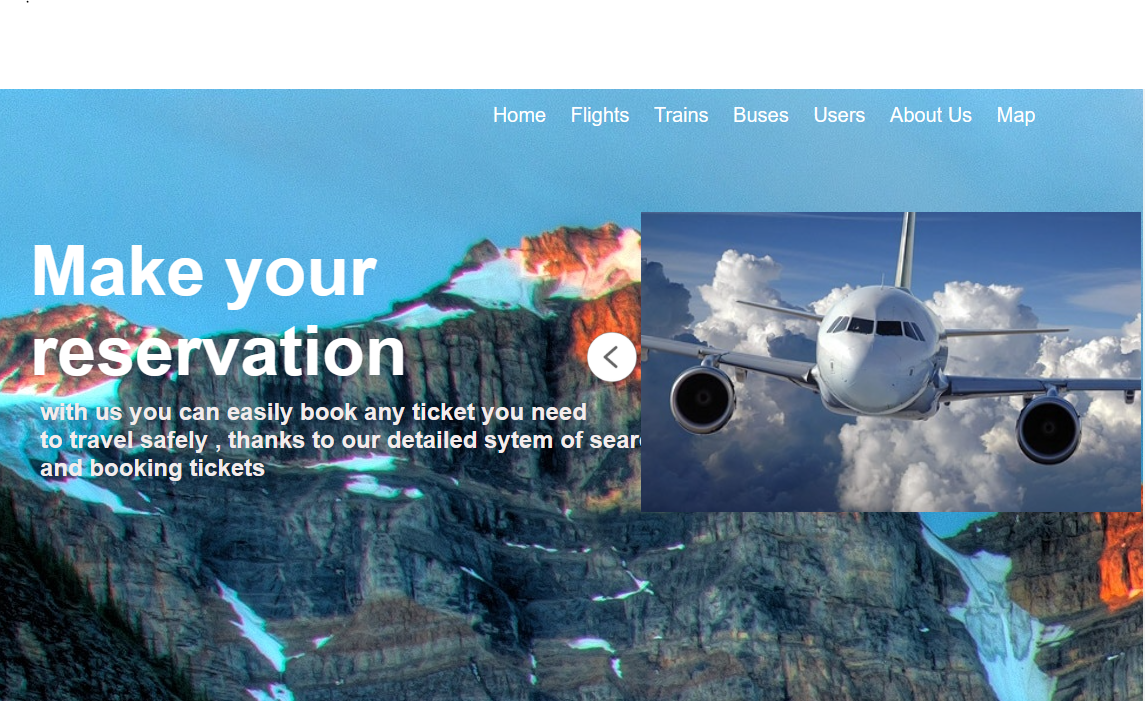
* [www.tutorialspoint.com](http://www.tutorialspoint.com/)
* [www.codexworld.com](http://www.codexworld.com/)
* [www.codecademy.com](http://www.codecademy.com/)
* [www.stackoverflow.com](http://www.stackoverflow.com/)
* [www.w3schools.com](http://www.w3schools.com/)
* [www.Sourcecodester](http://www.Sourcecodester).com
* [www.Projectworlds.com](http://www.Projectworlds.com)
* [www.FreeProjectz.com](http://www.FreeProjectz.com)
* www.vetbossel.com

**APPENDIX**

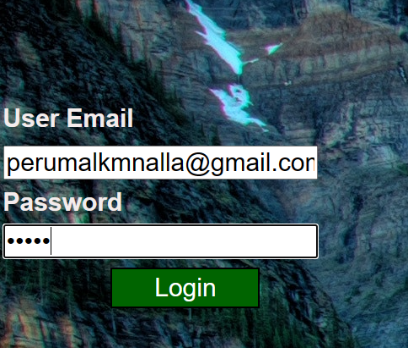
**11. APPENDIX**

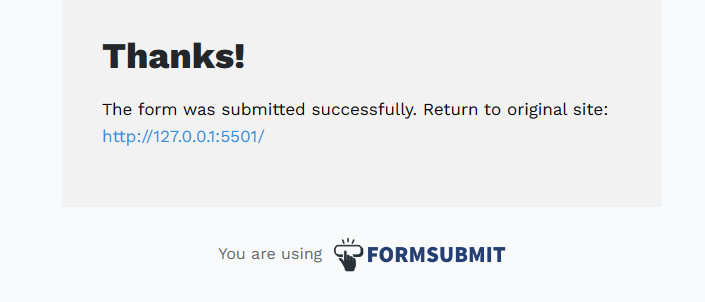
**11.SAMPLE SCREEN LAYOUTS:**

**IRCTC HOME PAGE:**

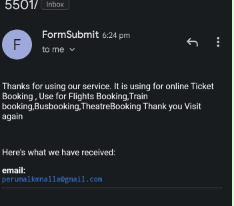
****

**USER LOGIN:**

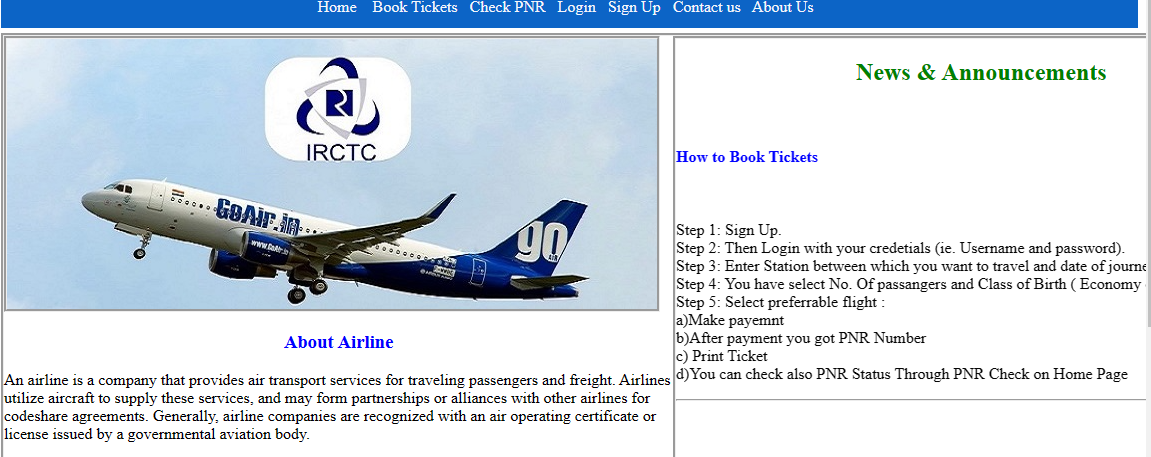




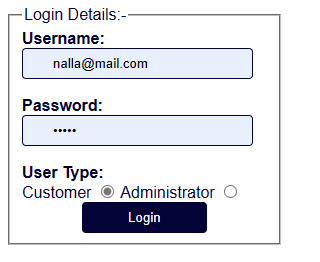
**GMAIL:**



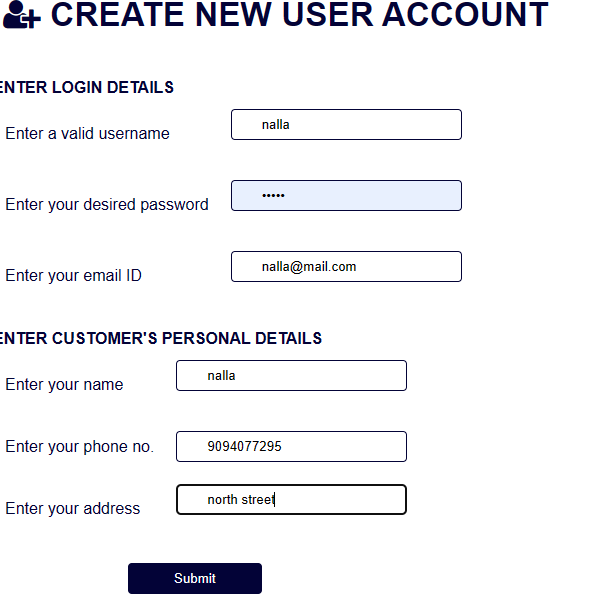
**FLIGHT HOME PAGE:**



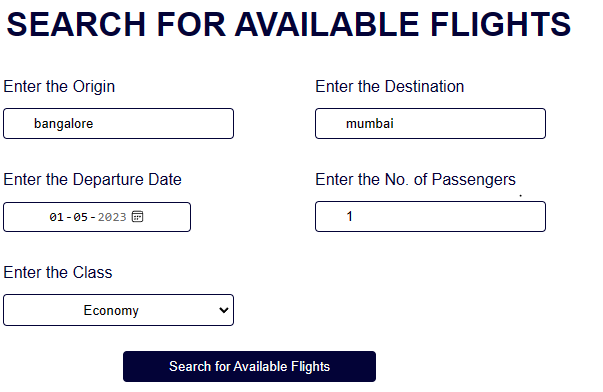
**FLIGHT LOGIN:**

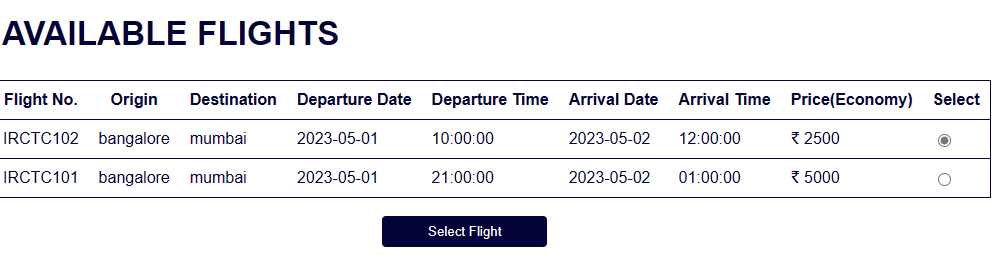


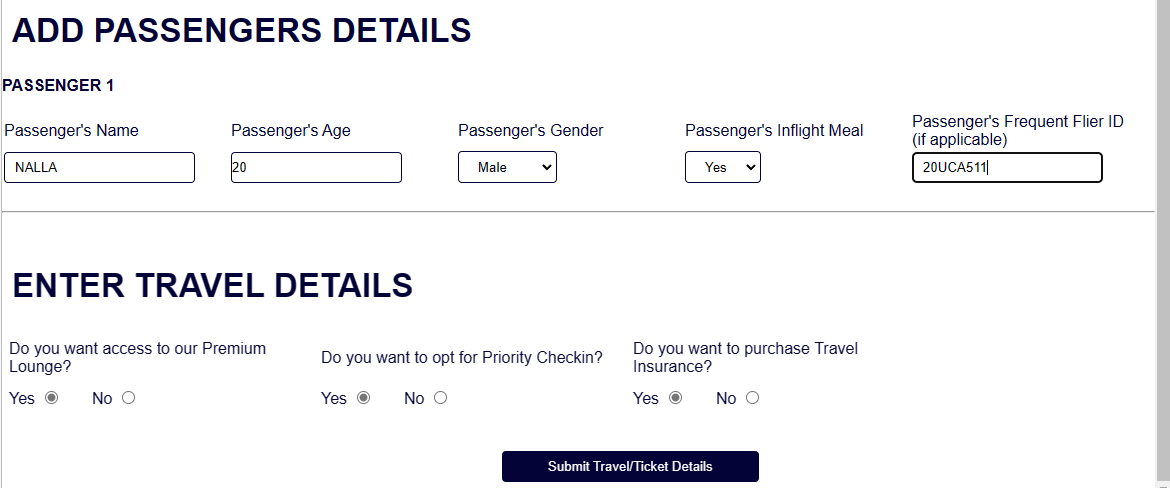
**FLIGHT SIGN UP:**

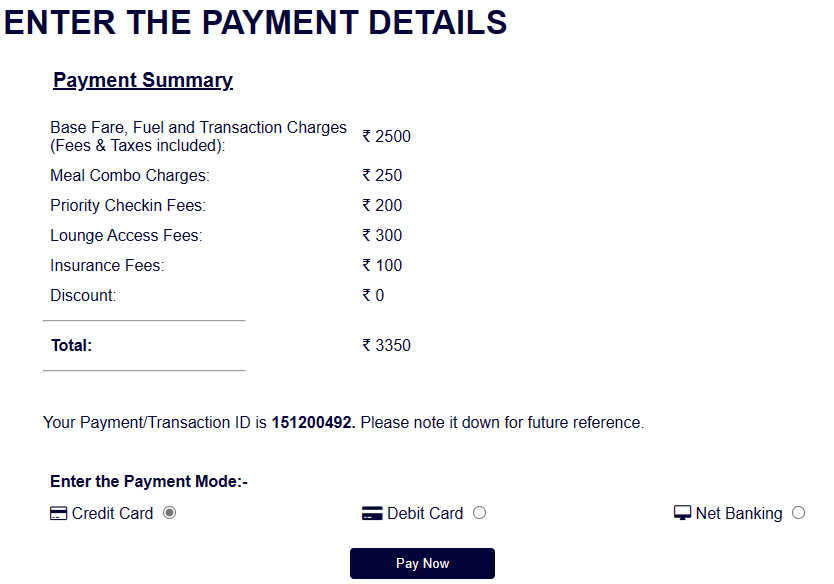


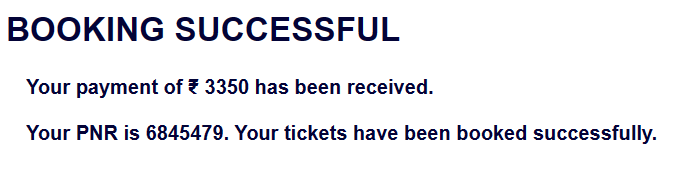
**FLIGHT BOOKING:**



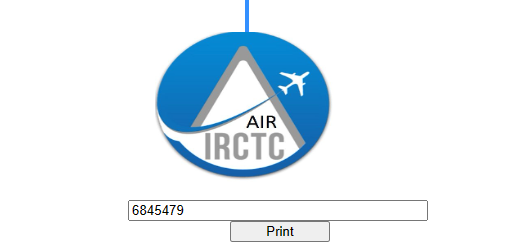




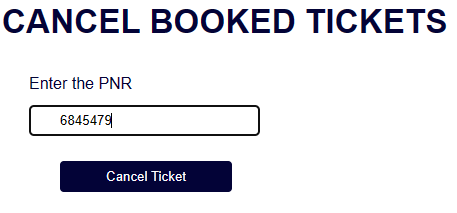


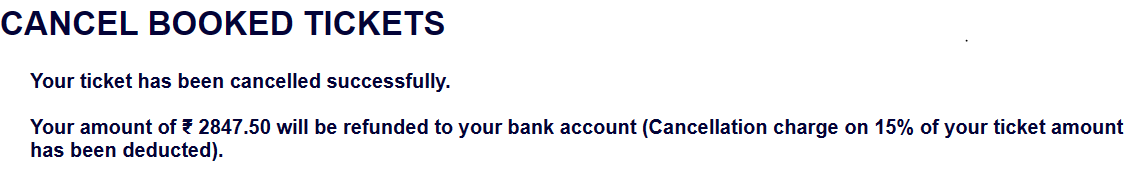


**PRINT TICKET:**

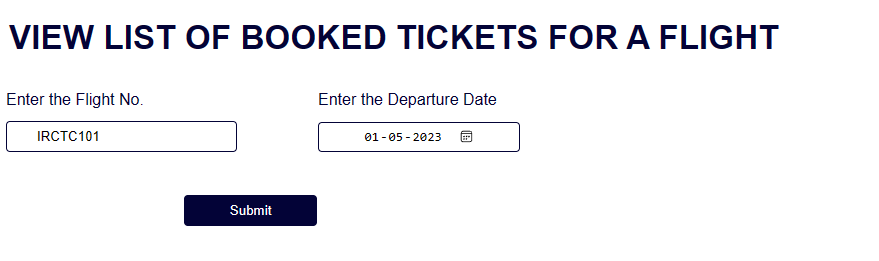




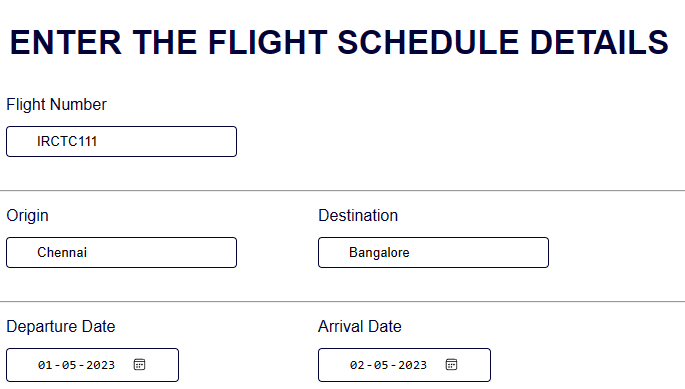




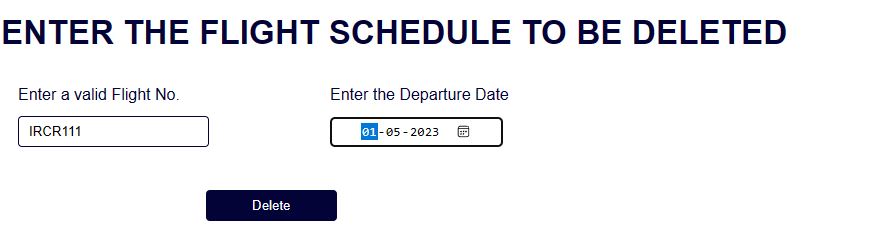
**ADMIN SIDE:**



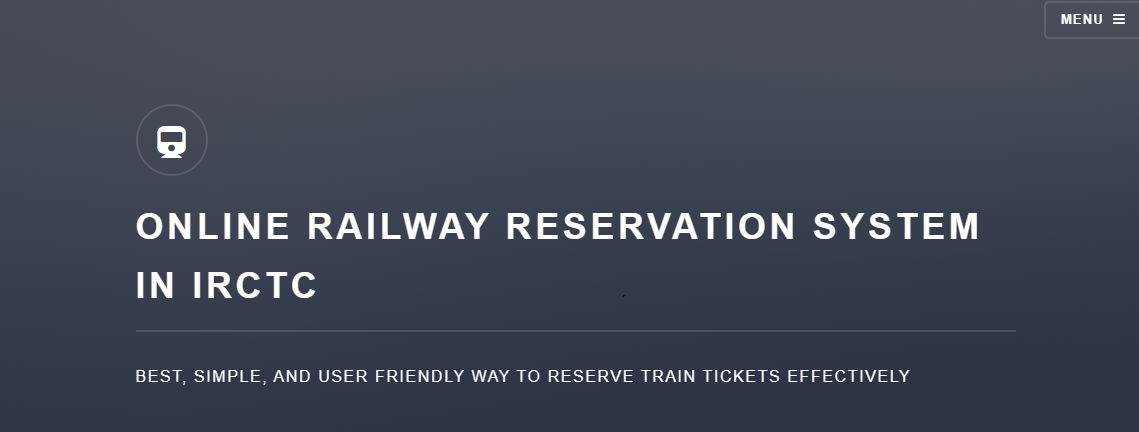


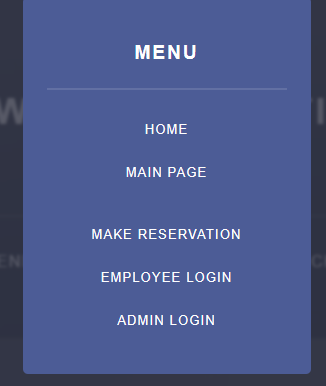






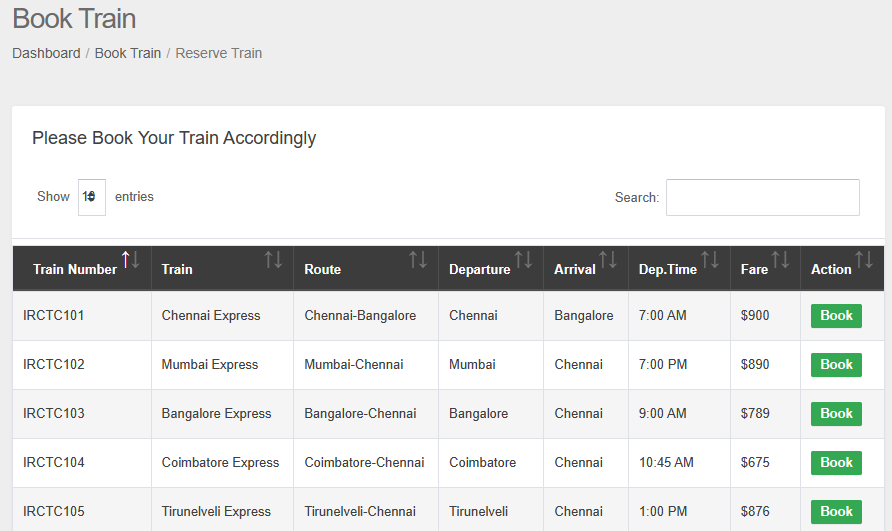
**TRAIN HOME PAGE:**

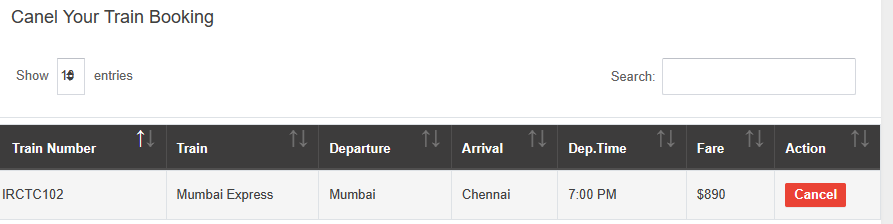




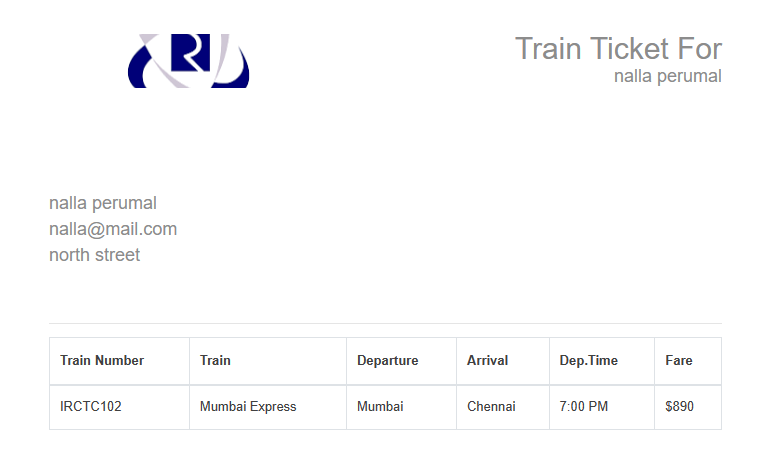
**USER LOGIN:**



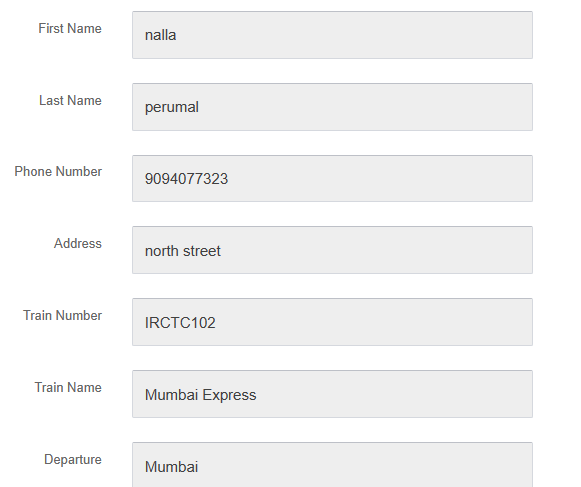


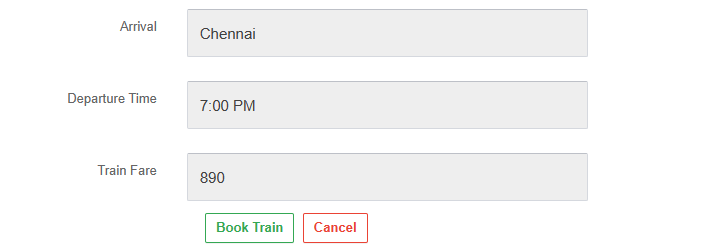


**PRINT TRIAN TICKET:**



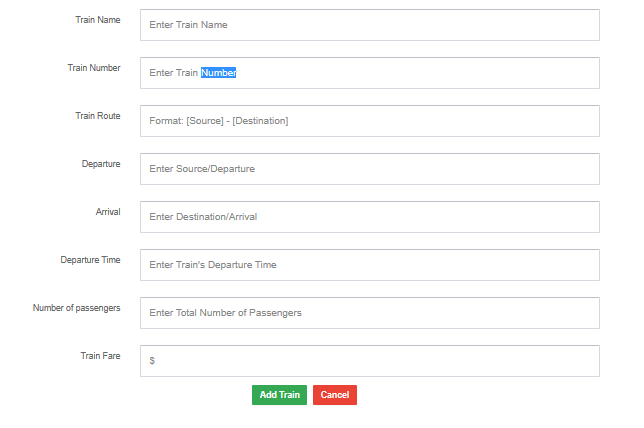
**BOOK TICKET:**



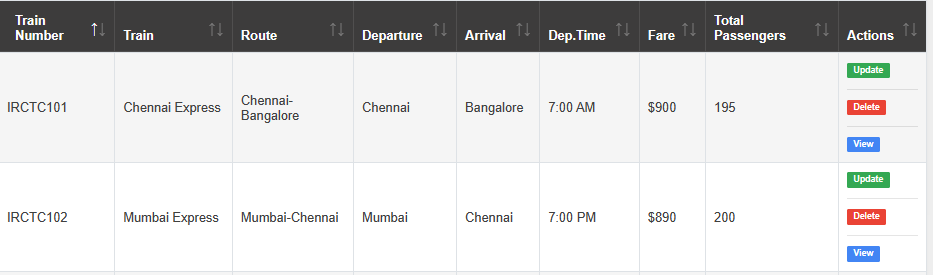


**ADMIN SIDE:**

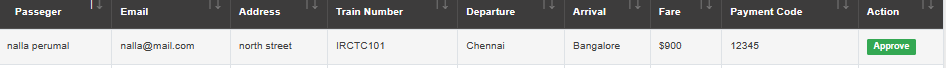
**ADD TRAIN DETAILS:**



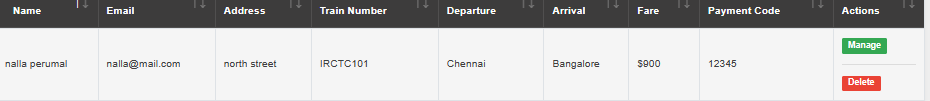
**MANAGE TRAIN:**



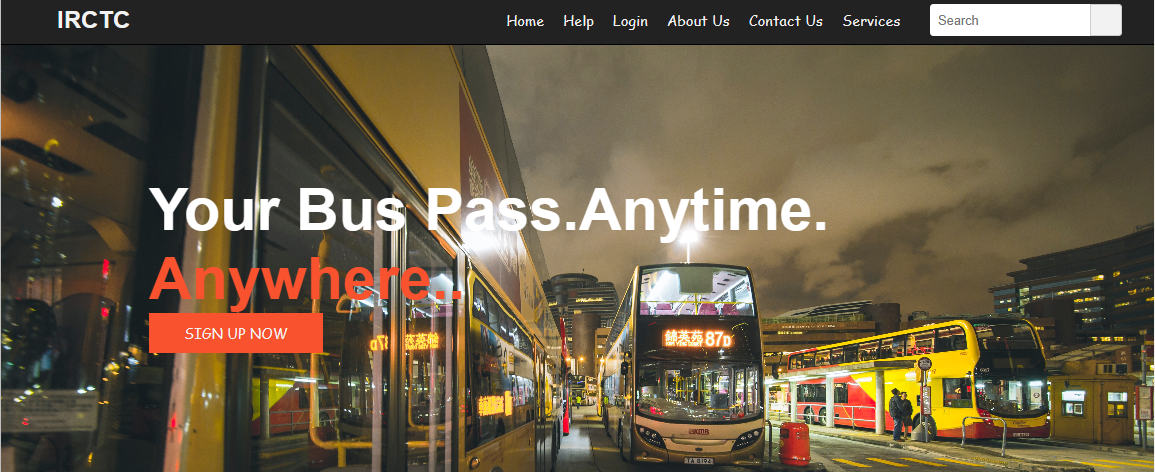
**APPROVE TRAIN TICKET:**



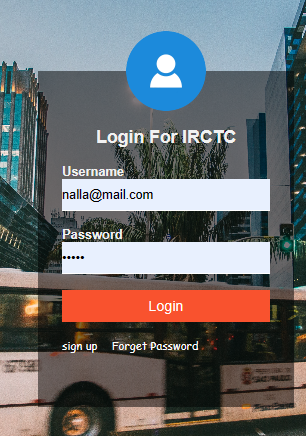
**MANAGE TRAIN TICKET:**



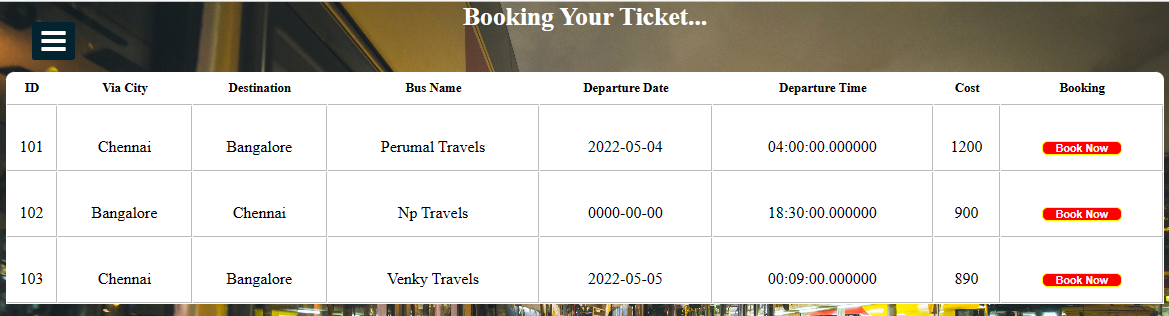
**BUS HOME PAGE:**

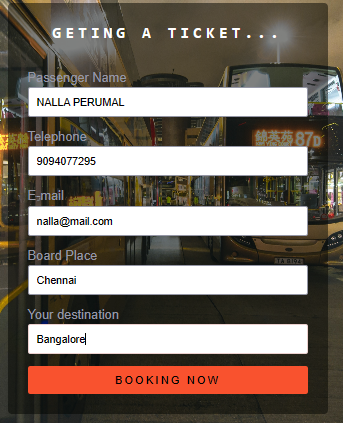


**USER LOGIN:**

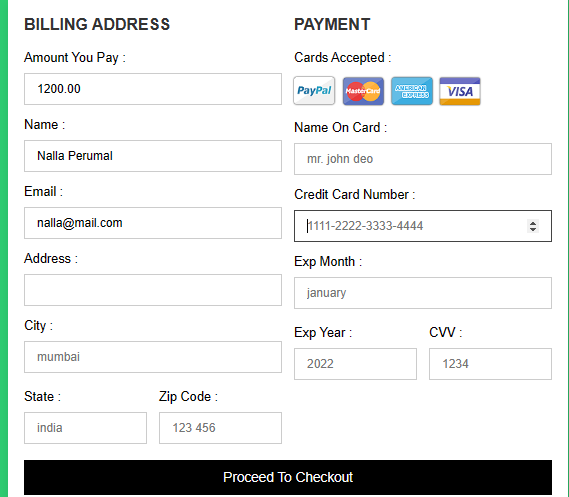


**BOOK BUS:**

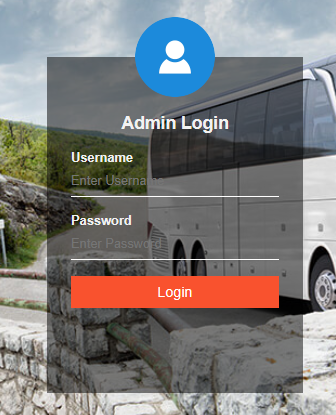


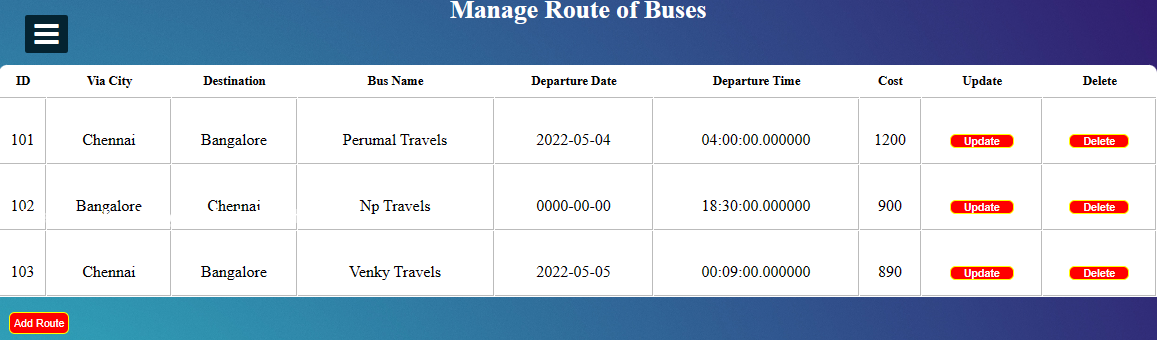


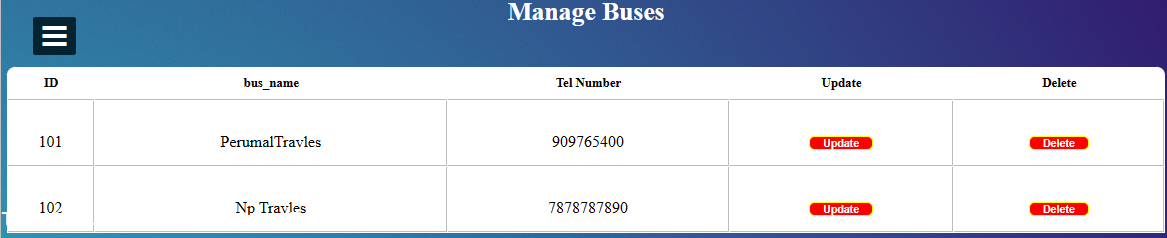
**PAYMENT:**

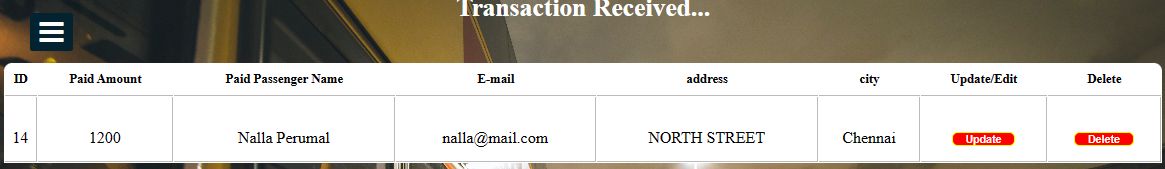


**ADMIN SIDE BUS:**









**11.2 SAMPLE CODING:**

**HOME PAGE:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Ticket Booking</title>

<link rel="stylesheet" href="styles.css">

</head>

<body background = "inature.jpg">

<header>

<nav>

<ul>

<li id= "hi">Home</li>

<li onclick="location.href='http://localhost/Ticket/airline/index.html';" >Flights</li>

<li onclick="location.href='http://localhost/Ticket/train/train.html';">Trains</li>

<li onclick="location.href='http://localhost/Ticket/Busbooking/bus.html';">Buses</li>

<li id="login">Users</li>

<li id="about">About Us</li>

<li onclick="location.href='http://localhost/Ticket/Home/map.html';">Map</li>

</ul>

</nav>

<h1 style="position:relative ;left: 110px;top:1500px; color:rgb(100, 53, 231);">.</h1>

</header>

<h1 class="heading">Make your<br> reservation </h1>

<h2 class="subheading">with us you can easily book any ticket you need<br>

to travel safely , thanks to our detailed sytem of searching<br> and booking tickets</h2>

<section>

<li id="back" class="back" ><img src = "ileft.png" style="height:50px;"></li>

<li><div class="box box1"><img id="box" src="iplane.jpg"></div></li>

<li id = "next" class="next" ><img src = "iright.png" style="height: 50px;" ></li>

</section>

<form action="https://formsubmit.co/nallaperumalkalayan@gmail.com" method="POST">

<section class="userDetails">

<h3 class="userEmail"> User Email </h3>

<input id="iUserEmail" class="iUserEmail" name="email" type="email" placeholder="Enter Useremail">

<h3 class="userPassword"> Password </h3>

<input class="iUserPassword" type="password" id="password" placeholder="Enter Password">

<button id = "submit" class="submit" onclick="getUserDetails()">Login</button>

<h3 id = "check" class="check"></h3>

<input type="hidden" name="\_autoresponse" value="Thanks for using our service.<br>

It is using for online Ticket Booking ,<br>

Use for Flights Booking,Train booking,Busbooking,TheatreBooking <br>

Thank you Visit again ">

</section>

</form>

<script src="main.js"></script>

<footer class="about">

<table class="table" style="width:50%">

<tr>

<th>Name</th>

<th>Register</th>

</tr>

<tr>

<td>Nalla Perumal</td>

<td>20UCA511</td>

</tr>

</table>

</footer>

</body>

</html>

**FLIGHT LOGIN.php:**

<?php

session\_start();

?>

<html>

<head>

<title>

Account Login

</title>

<style>

input {

border: 1.5px solid #030337;

border-radius: 4px;

padding: 7px 30px;

}

input[type=submit] {

background-color: #030337;

color: white;

border-radius: 4px;

padding: 7px 45px;

margin: 0px 60px

}

</style>

<link rel="stylesheet" type="text/css" href="css/style.css"/>

<link rel="stylesheet" href="font-awesome-4.7.0\css\font-awesome.min.css">

</head>

<body>

<img class="logo" src="images/irctc.jpg"/>

<h1 id="title">

IRCTC Airways </h1>

<div>

<ul>

<li><a href="home\_page.php"><i class="fa fa-home" aria-hidden="true"></i> Home</a></li>

<li><a href="http://localhost/Ticket/Home/index.html" >Main Page</a> &nbsp&nbsp</li>

<li><a href="login\_page.php"><i class="fa fa-ticket" aria-hidden="true"></i> Book Tickets</a></li>

<li><a href="home\_page.php"><i class="fa fa-plane" aria-hidden="true"></i> About Us</a></li>

<li><a href="home\_page.php"><i class="fa fa-phone" aria-hidden="true"></i> Contact Us</a></li>

<li><a href="login\_page.php"><i class="fa fa-sign-in" aria-hidden="true"></i> Login</a></li>

</ul>

</div>

<br>

<br>

<br>

<form class="float\_form" style="padding-left: 40px" action="login\_handler.php" method="POST">

<fieldset>

<legend>Login Details:-</legend>

<strong>Username:</strong><br>

<input type="text" name="username" placeholder="Enter your username" required><br><br>

<strong>Password:</strong><br>

<input type="password" name="password" placeholder="Enter your password" required><br><br>

<strong>User Type:</strong><br>

Customer <input type='radio' name='user\_type' value='Customer' checked/> Administrator <input type='radio' name='user\_type' value='Administrator'/>

<br>

<?php

if(isset($\_GET['msg']) && $\_GET['msg']=='failed')

{

echo "<br>

<strong style='color:red'>Invalid Username/Password</strong>

<br><br>";

}

?>

<input type="submit" name="Login" value="Login">

</fieldset>

<br>

<a href="new\_user.php"><i class="fa fa-user-plus" aria-hidden="true"></i> Create New User Account?</a>

</form>

</body>

</html>