A Project Report

On

BUS TICKET RESERVATION SYSTEM

Submitted in partial fulfilment of the
requirements for the award of the degree
of

MASTER OF COMPUTER APPLICATIONS

SUBMITTED BY

DAMARASINGU.GOWTHAM BHASKAR

(22P31F0009)

Under the Esteemed Guidance of

Ms.Mohammad.Gousia, MCA

Assistant Professor

DEPARTMENT OF MCA

ADITYA COLLEGE OF ENGINEERING AND TECHNOLOGY

(Approved by AICTE, Affiliated to JNTUK & Accredited by NBA, NAAC with ‘A+’Grade)
Recognized by UGC under the sections 2(f) and 12(B) of the UGC act 1956
SURAMPALEM-533437, East Godavari District, ANDHRA PRADESH.
2022-2024

ADITYA COLLEGE OF ENGINEERING AND TECHNOLOGY

(Approved by AICTE, Affiliated to JNTUK & Accredited by NBA, NAAC with 'A+'Grade)

Recognized by UGC under the sections 2(f) and 12(B) of the UGC act 1956

SURAMPALEM-533437, E.G.Dist, ANDHRA PRADESH.

2022-2024

DEPARTMENT OF MCA

CERTIFICATE

This is to certify that the project work entitled, **"BUS TICKET RESERVATION SYSTEM",** is a Bonafede work carried out by **DAMARASINGU.GOWTHAM BHASKAR** bearing Regd.No:**22P31F0009** submitted to the requirements for the award of the Computer Applications in partial fulfilment of the requirements for the award of degree of MASTER OF COMPUTER APPLICATIONS from **Aditya College of Engineering and Technology,** Surampalem during the academic year **2023-2024.**

Project Guide

Ms.Mohammad.Gousia, MCA

Assistant Professor,
Department of MCA,
Aditya College of Engg & Tech,
Surampalem-533437.

Head of the Department

Mr.R.V.V.N.BheemaRao, M.Tech(Ph.D)

Associate Professor,
Department of MCA,
Aditya College of Engg & Tech,
Surampalem-533437.

External Examiner

DECLARATION

I hereby declare that the project entitled **“BUS TICKET RESERVATION SYSTEM”** done on my own and submitted to **Aditya College of Engineering & Technology, Surampalem** has been carried out by me alone under the guidance of **Ms.Mohammad.Gousia.**

Place: Surampalem

Date:

(D.GOWTHAM BHASKAR)

22P31F0009

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that I have now the opportunity to express my gratitude for all of them.

The first person I would like to thank is my guide **Ms.Mohammad.Gousia, MCA, Assistant Professor, Aditya College Of Engineering And Technology, Surampalem,** Her wide knowledge and logical way of thinking have made a deep impression on me. Her understanding, encouragement, and personal guidance have provided the basis for this project. She is a source of inspiration for innovative ideas and his kind support is well known to all her students and colleagues.

I wish to thank **Mr.R.V.V.N.Bheemarao, M.Tech, Head of the Department, Aditya College Of Engineering And Technology, Surampalem,** has extended his support for the success of this project.

I also wish to thank **Dr.D.Sanjay, Principal, Aditya College Of Engineering And Technology, Surampalem,** who has extended his support for the success of this project.

I would like to thank all the **Management & Technical Supporting Staff** for their timely help throughout the project.

ABSTRACT

ABSTRACT

The Bus Ticket Reservation System was developed by using python programming language and Django framework. The system helps travel business provide their clients a platform to reserve a ticket to their desired destination. This system has only a small scope but enough to help you guys understand how to create a ticket reservation system like this. This system also has a simple admin panel. The admin panel is a restricted area of the system which only can be accessed by the admin user using their system credentials. On this side of the system, the admin user can confirm/cancel their client’s reservations.

In these modern days the population has been increased and the travelling has become as one of their daily routines. With the increased number of customers or travellers it would require more man power. When we want to travel by bus, we usually go to the reservation centre and book a ticket for particular day. It is a time-consuming process and now we make this process easier and faster by creating a web application that is used to book tickets by using digital method. Here the people can easily use the web application by simply clicking on the URL that we have provided. After clicking on it the user needed to enter the details about the location that user wanted to go along with that some of their personal details like the user’s name, contact number, address etc.., The user can easily book ticket from anywhere and after a successful booking the registration details will be displayed on the owner’s desktop. Using simple components and programming techniques the web application is created. The users can easily travel from one place to another. It is a time saving process and user-friendly web application.

LIST OF CONTENTS

PAGE NO

LIST OF FIGURES

LIST OF SCREEN SHOTS

CHAPTER-1
INTRODUCTION

1.INTRODUCTION

The ticket machines would end the use of the hefty 1.5-kg ticket racks carried by conductors. It would also end the practice of tearing out tickets and marking fare stages. The Conductor would just have to key in facilitate inspection by the corporation's checking inspectors. The ticket machines would help prevent loss on account of malpractice. It would also help in providing adequate data to the corporation, particularly with the details about the fare stage and the ticket machine would print out the ticket. The machine weighs only 800 grams and is convenient to carry. The parameters are almost like that of a railway ticket, the only difference being that the machine is portable. The machine can print out 2,300 tickets, including the journey report in order to regard to the boarding of passengers from fare stages and important points. This would help the corporation prepare and organize its schedules more efficiently on the basis of traffic demand. Besides, it would provide data on concessions given to various sections. Another additional feature is that the data in the ticket machine could be fed into the computer. More over the depots of the corporation would be fully computerized so we want to add some other modules in our domain for depot’s verification.

1.1 Brief Introduction Of The Project:

The bus ticket reservation system is currently maintaining the project Transport company’s process which is a very time-consuming process. It deals with transport industry ticket booking and transport maintenance, so it becomes a very tedious job for the et booking transporter to look after these particulars to complete the task at right time. the Bus ticket reservation system not only deals with transporters owned vehicles but also takes with consideration about the other type project of the system transport vehicles available with other transporters To develop a software application that support specific to the project travel agency automation that can solve all tedious tasks related to ticket booking in a travel agency This system project is made as user friendly as possible so that any one can use it with little knowledge This system will lead to increase in the ticket booking efficiency of the project staff and members of the ticket booking agency with title throughout of computers .The ticket booking project will reduce the ticket booking tedious job of the system paperwork by keeping all the project details of bus ticket booking, cancelling tickets are stored in the form database in computers hard disk Up-to-date information of the system performance status and enquiry We provide up to dale information that is not possible manually.

1.2 Motivation Of The Project:

This Project Is Suggested by Travel Agent. This Project Is Taken for The Purpose That There Is Lot Of Problem Related to Buses Route, Timing, Ticketing, Trip Details, Bus Details, Bus Stops in Local Areas. More Importantly to Know the Running Time of A Particular Bus And We Want To Manipulate And Stores These Information Successful.

1.3 Objective Of The Project:

The Bus ticket reservation system is currently maintaining the project Transport Company’s process manually which is a very time-consuming process. It deals with transport industry's ticket booking and transport maintenance, so it becomes a very tedious job for the ticket booking transporter to look after these particulars to complete the task at right time. The bus ticket booking system not only deals with transporters owned vehicles but also takes into consideration about the other types project of system transport vehicles available with other transporters. To develop a software application that supports Specific to the project Travel Agency Automation that can solve all tedious tasks related to ticket booking in a travel agency.

- This system will lead to an increase in the ticket booking efficiency of the project Staff and members of the Ticket Booking Agency with little throughput.

- This system project is Mauser-friendly as possible so anyone one can use it with little knowledge of system computers.

- The ticket booking project will reduce the ticket booking tedious job of system paperwork by keeping all the project details of bus ticket booking, cancelling tickets are stored in the form database in the computer's hard disk.

- The objective of my project is to make easy the ticket booking project system of Ticket Booking Agency simple, reliable, user friendly, and corrective. Moreover, less time-consuming as compared to manual work.

1.4 Organization of the Project:

- **System Analysis:** The description of the current system, the planned system, and the required specifications make up the majority of this chapter.
- **System Design:** This chapter is consisting of modules description and algorithms with example and use case diagrams, class diagrams, sequence diagrams, Collaboration diagrams and activity diagrams.
- **Technology Description:** This chapter mainly consists of the technology description of this project.
- **Sample Code:** This chapter consisting of sample code for the few modules.
- **Testing:** This chapter is mainly consisting of testing techniques and test cases for modules.
- **Screen Shots:** This chapter is mainly consisting of output screens of this project.
- **Conclusion:** Main Conclusion of the project is to segmenting the customers based on characteristics.

CHAPTER-2
SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

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.

Draw backs

- Lack of security of data.
- More manpower.
- Time consuming.
- Consumes large volume of pare work.
- Needs manual calculations.
- No direct role for the higher officials.
- Damage of machines due to lack of attention.

2.2 Proposed System

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

- Ensure data accuracy.
- Minimize manual data entry.
- Better Service.
- Greater efficiency.

2.3 Feasibility Study

A Feasibility study evaluates the project’s potential for success; therefore, perceived objectivity is an important factor in the credibility of the study for potential investors and lending institutions. It must therefore, be conducted with an objective, unbiased approach to provide information upon which decisions can be based. Here, we discuss 3 major feasibility studies required for our project.

- Economic Feasibility
- Technical Feasibility
- Operational Feasibility

ECONOMIC FEASIBILITY

Economic Feasibility defines whether the expected benefit equals or exceeds the expected costs. It is also commonly referred to as cost/benefit analysis. The procedure is to determine the benefits and the savings expected from the system and compare them with the costs. A proposed system is expected to outweigh the costs. This is small project with no cost for development. The system is easy to understand use. Therefore, there is no need to spend on training to use the system. This system has the potential to grow by adding functionalities for persons. Hence, the project could have economic benefits in future.

TECHNICAL FEASIBILITY

Technical feasibility is carried out to determine whether the project is feasible in terms of software, hardware, personnel, and expertise, to handle the completion of the project. It considers determining resources for the proposed system. As the system is developed using python, it is platform independent. Therefore, the users of the system can have average processing capabilities, running on any platform. The technology is one of the latest hence the system is also technically feasible.

OPERATIONAL FEASIBILITY

Operational feasibility is the measure of how well a proposed system solves the problems with the users. Operational feasibility is dependent on human resources available for the project and involves projecting whether the system will be used if it is developed and implemented. The project is operationally feasible for the users as nowadays almost all the humans are familiar with digital technology.

2.4 Functional Requirements

- Every online booking needs to be associated with an account.
- One account cannot be associated with multiple users.
- Search results should enable users to find the most recent and relevant booking options.
- System should enable users to book / pay for their tickets only in a timeboxed manner after tickets being added to the cart.
- System should only allow users to move to payment only when mandatory fields such as date, time, location has been mentioned.
- System should consider time zone synchronisation when accepting bookings from different time zones.
- Booking confirmation should be sent to user to the specified contact details.

2.5 Non Functional Requirements

The major non-functional Requirements of the system are as follows

- Use of captcha and encryption to avoid bots from booking tickets.
- System should accept payments via different payment methods, like PayPal, wallets, cards, vouchers, etc

2.6 Software and Hardware requirements

SOFTWARE REQUIREMENTS

• **Operating System**	**:**	Windows 10
• **Backend**	**:**	Python
• **Framework**	**:**	Django
• **Frontend**	**:**	HTML, CSS

HARDWARE REQUIREMENTS

• **Processor**	**:**	Intel Core i5
• **Speed**	**:**	2.2 Ghz
• **RAM**	**:**	8GB
• **Hard Disk**	**:**	500GB

CHAPTER-3
SYSTEM DESIGN

3.SYSTEM DESIGN

3.1 Introduction

The purpose of the design phase is to plan a solution of the problem specified by the requirement document. This phase is the first step in moving from the problem domain to the solution domain. In other words, starting with what is needed, design takes us toward how to satisfy the needs. The design of a system is perhaps the most critical factor affection the quality of the software; it has a major impact on the later phase, particularly testing, maintenance. The output of this phase is the design document. This document is similar to a blueprint for the solution and is used later during implementation, testing and maintenance. The design activity is often divided into two separate phases System Design and Detailed Design.

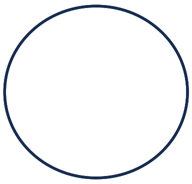
3.2 Data Dictionary

A data dictionary contains metadata and data about the database. The data dictionary is very important as it contains information such as what is in the database, who can access it, where is the database physically stored etc. The users of the database normally don't interact with the data dictionary, it is only handled by the database administrators. The data dictionary in general contains information about the following Names of all the database tables and their schemas.

3.3 Data Flow Diagram

Data flow-oriented techniques advocate that the major data items handled by a system must be first identified and then the processing required on these data items to produce the desired outputs should be determined. The DFD (also called as bubble chart) is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on these data, and the output generated by the system. It was introduced by De Macro (1978), Gane and Sur son (1979). The primitive symbols used for constructing DFD's are:

Symbols used in DFD



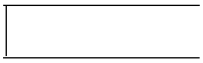
A Circle represents process.

A Rectangle represents
external entity.

A Square defines a Source or destination of the system
data.

An Arrow

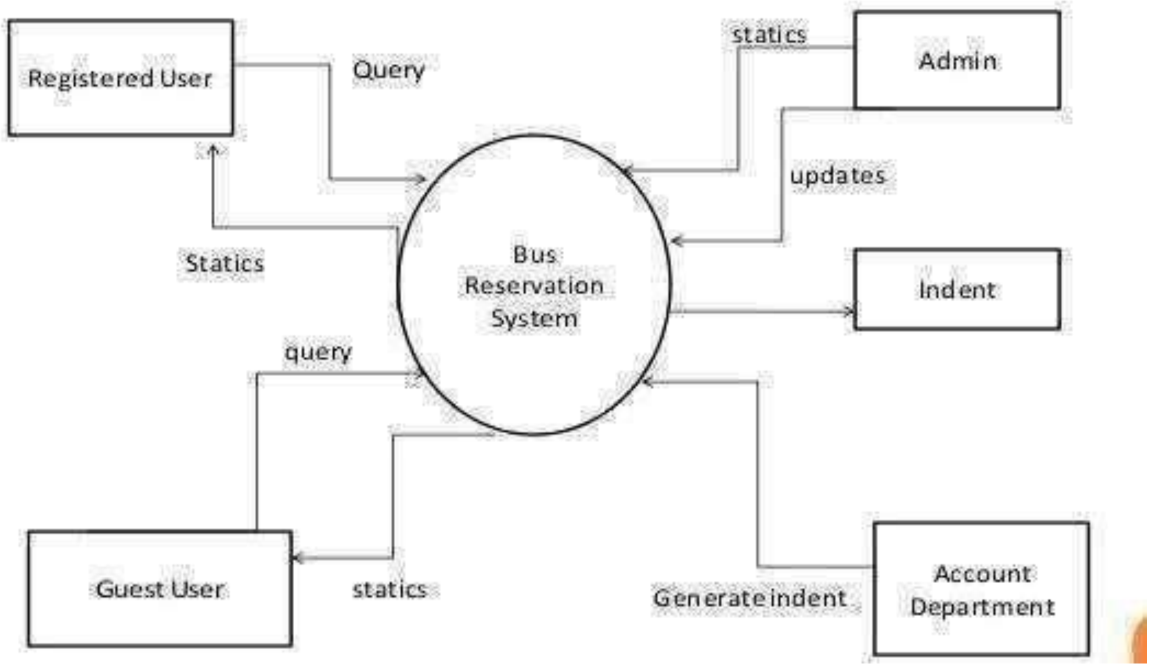
defines Data flow



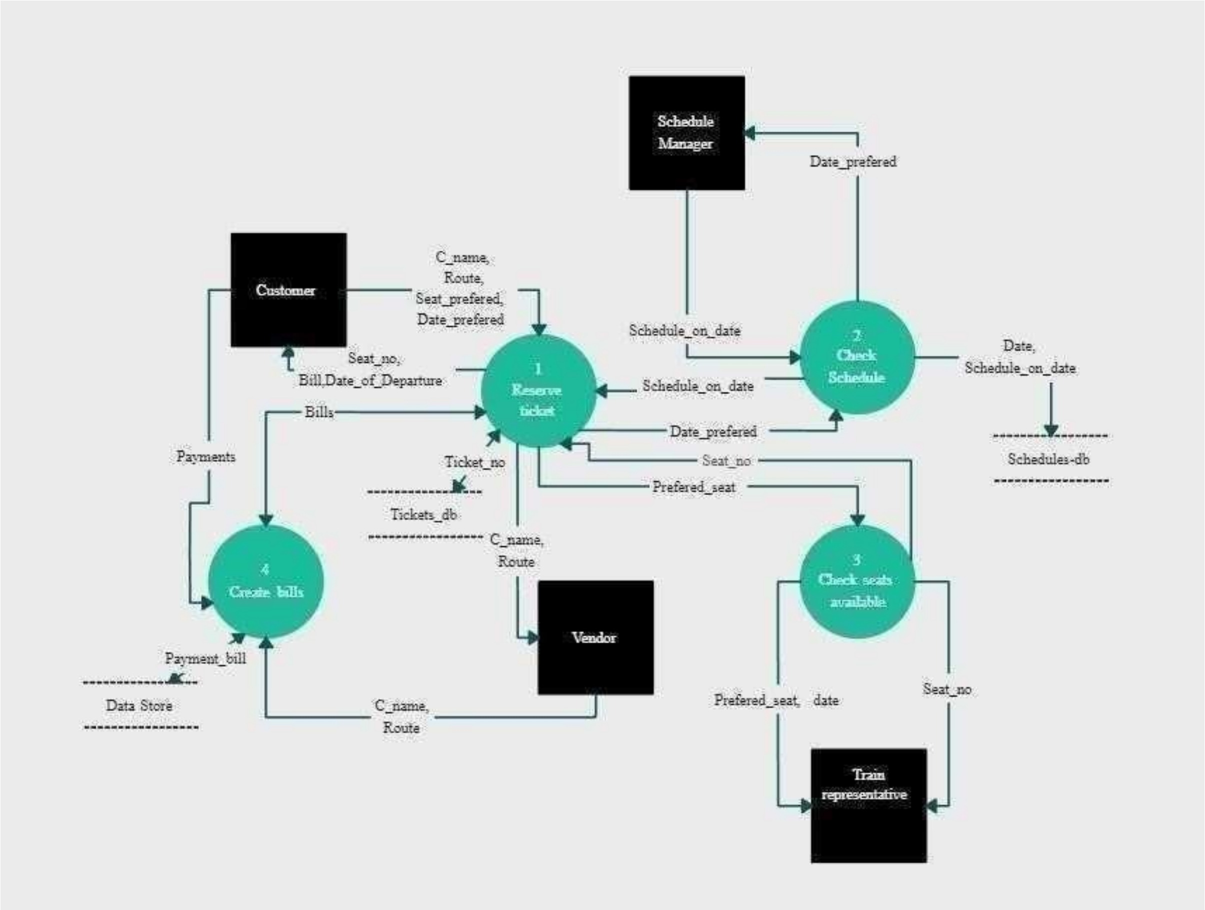
Double line with one end closed indicates data store.

Data Flow Diagram

DFD DIAGRAM "0 LEVEL"

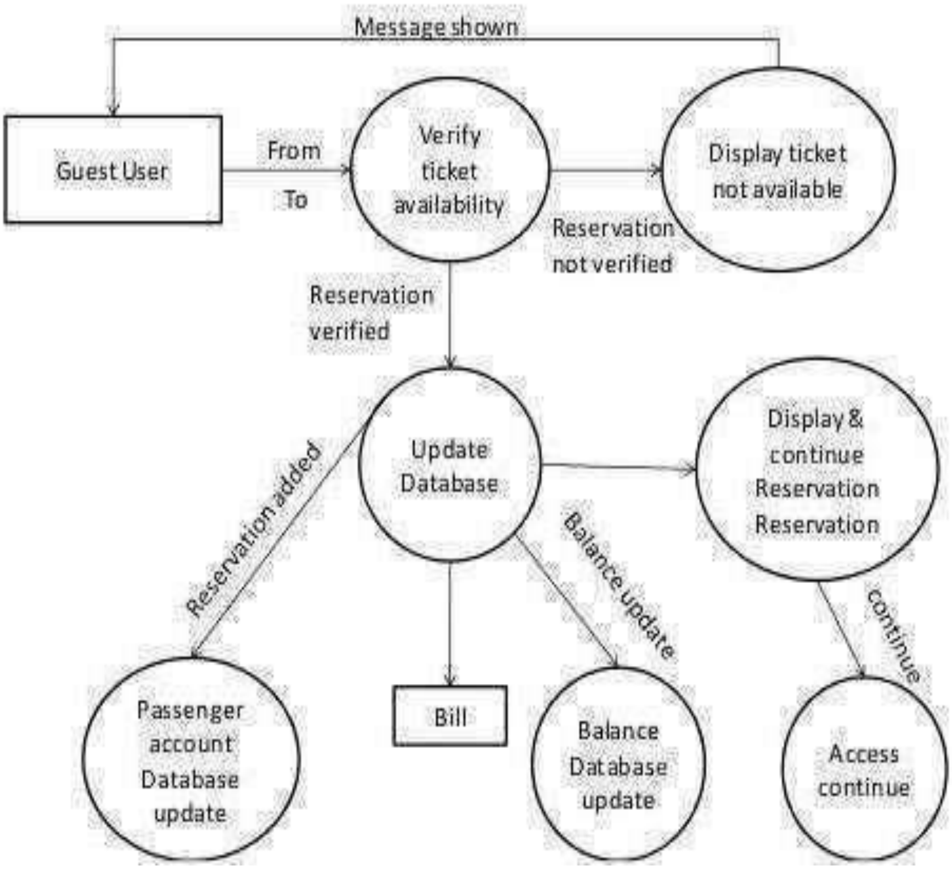


LEVEL-1


Schedule Manager
Date_prefered
Customer
C_name,
Route,
Seat_prefered,
Date_prefered
Schedule_on_date
2
Check
Schedule
Date,
Schedule_on_date
Seat_no,
Bill,Date_of_Departure
1
Reserve
ticket
Schedule_on_date
Bills
Date_prefered
Schedules-db
Payments
Ticket_no
Seat_no
Prefered_seat
Tickets_db
C_name,
Route
4
Create bills
3
Check seats
available
Vendor
Payment_bill
Data Store
C_name,
Route
Prefered_seat, date
Seat_no
Train
representative

LEVEL-2

DFD FOR GUEST USER



3.4 UML Diagrams

The unified modelling language allows the software engineer to express an analysis model using the modelling notation that is governed by a set of syntactic semantic and pragmatic rules. A UML system is represented using five different views that describe the system from distinctly different perspective. UML is specifically constructed through two different domains UML Analysis modelling, this focuses on the user model and structural model views of the system UML design modelling, which focuses on the behavioural modelling, implementation modelling and environmental model views. These are divided into the following types.

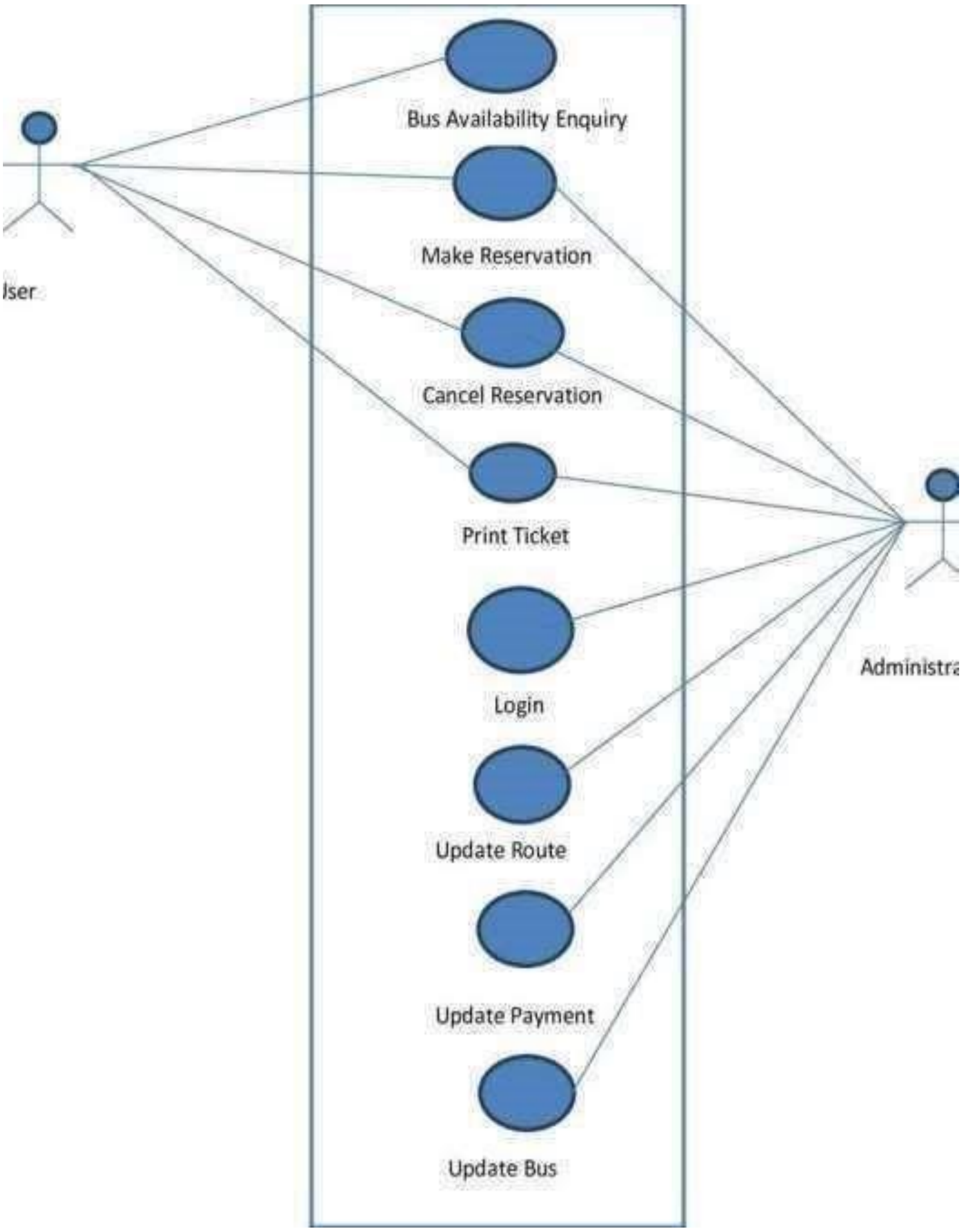
• Use case Diagram • Class Diagram

3.4.1 Use case Diagram

Use Case diagrams identify the functionality provided by the system (use cases), the users who interact with the system (actors), and the association between the users and the functionality. Use Cases are used in the Analysis phase of software development to articulate the high-level requirements of the system. The primary goals of Use Case diagrams include:

- Providing a high level view of what the system does.
- Identifying the users ("actors") of the system. • Determining the areas needed human-computer interfaces.

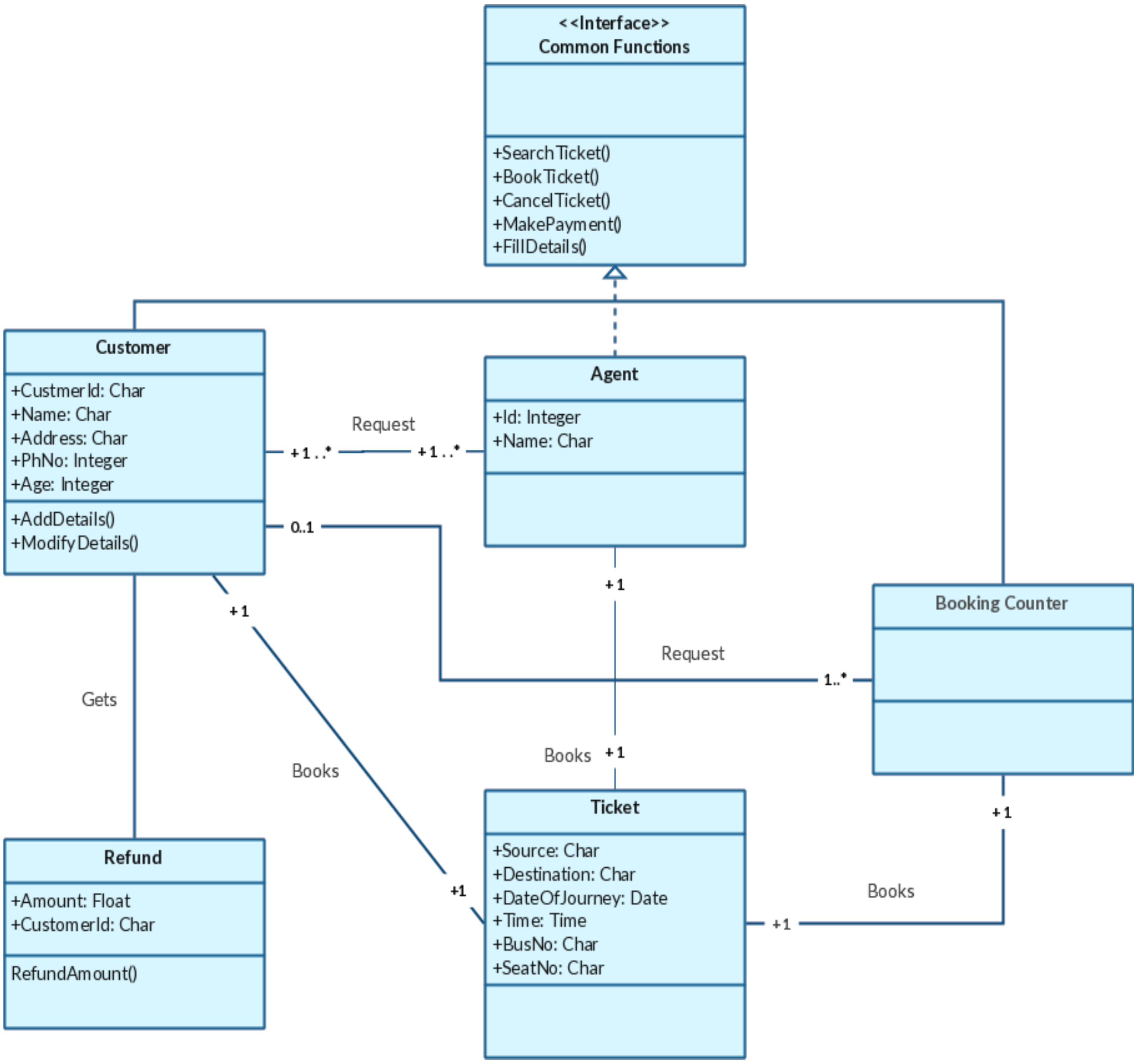
USE CASE DIAGRAM



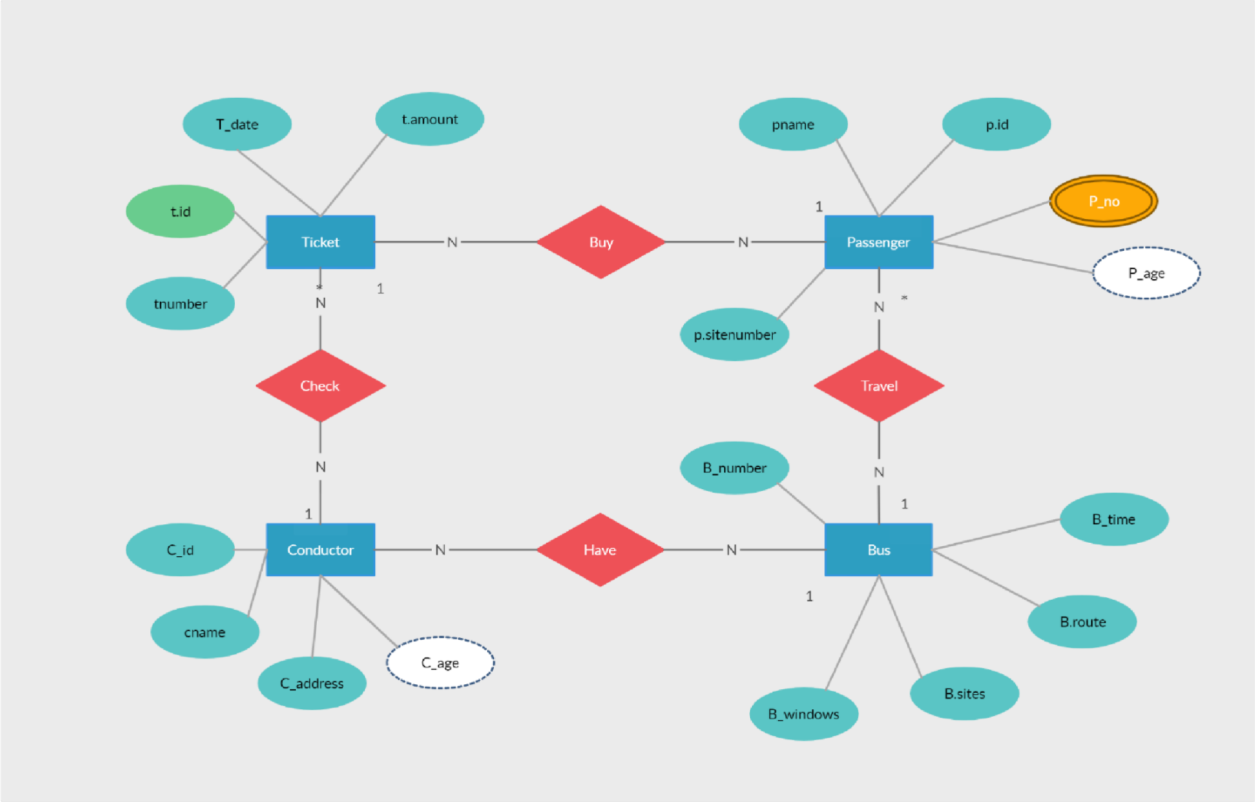
3.4.2 Class Diagram

Class diagrams identify the class structure of a system, including the properties and methods of each class. Also depicted are the various relationships that can exist between classes, such as an inheritance relationship. Part of the popularity of Class diagrams stems from the fact that many CASE tools, such as Rational XDE, will autogenerate code in a variety of languages, these tools can synchronize models and code, reducing the workload, and can also generate Class diagrams from object-oriented code.

CLASS DIAGRAM



3.4.3 ER DIAGRAM



CHAPTER-4
TECHNOLOGY DESCRIPION

4. TECHNOLOGY DESCRIPTION

4.1 HTML Introduction:

Hypertext Mark-up Language (HTML), the languages of the world wide web (WWW), allows users to produce web pages that included text, graphics and pointer to other web pages (Hyperlinks).HTML is not a programming language but it is an applications of ISO standard 8879, SGML (Standard Generalized Mark -up Language), but specialized to hypertext and adapted to the web. The idea behind Hypertext one point to another point. We can navigate through the information based on out interest and preference. A mark-up language is simply a series of enclosed within the elements should be displayed.

Hyperlinks are underlined or emphasized works that load to other documents or some portions of the same document. Html can be used to display any type of document on the host computer, which can be geographically at a different location. It is a versatile language and can be used on any platform or desktop. HTML provides tags (special code) to make the document look attractive. HTML provides or not case –sensitive. Using graphics, fonts, different sizes, colour, etc. can enhance the presentation of the document. Anything that is not a tag is part of the document itself.

4.1.1 Basic HTML Tags:

< ! ----> Specific comments.

<A>….</A> Creates Hypertext links.

<B>….</B> Creates Hypertext links.

<Big>….</Big> Formatted text in large-font.

<Body>…..</Body> contains all tags and text in the Html-document.

<Center>…..</Center> Creates Text.

<DD>….</DD> Definition of a term.

<TABLE>…</TABLE> Creates table.

<Td>………</Td> indicates table data in a table.

<Tr>………..</Tr> designates a table row.

<Th>………..<Th> creates a heading in a table.

ADVANTAGES

- A HTML document is a small and hence easy to send over the net. It is a small because it does not include formatted information.
- HTML is a platform independent.

4.2 CSS Introduction

CSS is used to control the style of a web document in a simple and easy way.**CSS** is the acronym for **"Cascading Style Sheet"**. This tutorial covers both the versions CSS1, CSS2 and CSS3, and gives a complete understanding of CSS, starting from its basics to advanced concepts. **Cascading Style Sheets**, fondly referred to as **CSS**, is a simple design language intended to simplify the process of making web pages presentable.**CSS** is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. I will list down some of the key advantages of learning CSS:

4.3 PYTHON Introduction

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics developed by Guido van Rossum. It was originally released in 1991. Designed to be easy as well as fun, the name "Python" is a nod to the British comedy group Monty Python. Python has a reputation as a beginner-friendly language, replacing Java as the most widely used introductory language because it handles much of the complexity for the user, allowing beginners to focus on fully grasping programming concepts rather than minute details.

Python is used for server-side web development, software development, mathematics, and system scripting, and is popular for Rapid Application Development and as a scripting or glue language to tie existing components because of its high-level, built-in data structures, dynamic typing, and dynamic binding. Program maintenance costs are reduced with Python due to the easily learned syntax and emphasis on readability. Additionally, Python's support of modules and packages facilitates modular programs and reuse of code. Python is an open-source community language, so numerous independent programmers are continually building libraries and functionality for it.

4.4 Modules Description

This project is modularized as the following:

1.Management of Routes Module:

This module includes information about how we can Manage the routes for a particular bus service so in the case of Route management module we must know the details about route number, number of stops, fare stages and running time of the particular bus more over we want to manipulate and stored this information successfully.

2.Trip information Module:

Each journey is identified as a trip. Each ticket must contain the trip no so that calculation of passenger can be done easily. Here in this section, we want to know start time and route no of the bus this information can be manipulate and stored successfully.

3.Bus Detail Module:

In this module all bus details are stored and manipulated, in bus detail module contains minimum charge, type, depot, fare increment, bus number, and passenger's states (child or adult) are manipulated and stored.

4.Bus Stops Module:

Bus Stops module includes information about what are the main bus stops of a particular bus. This module connected to the route of the bus and it is used to store stop number, stop name and fare stages and Route number.

5.Bus Ticketing Module:

Ticketing is the most important module in this Project which uses all the table together and calculates fare for the pass, beginning stop, end stop, ticket number persons (Adult/child) rate, date and time also we want to print the all this information. In order to do the calculation data has to be pulled out from stops, bus, trip and route. Number of passengers & the states are entered by the Venter and to produce the tickets.

CHAPTER-5
SAMPLE CODE

5. SAMPLE CODE

BASE.HTML

```
<!doctype html>
<html lang="en">
<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  <!-- Bootstrap CSS -->
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
       integrity="sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
crossorigin="anonymous">

  <title>Bus Ticket Reservation System</title>
</head>
<body>

<nav class="navbar navbar-expand-lg navbar-light bg-success">
  <div class="container">
    <a class="navbar-brand" href="#" style="color: rgb(14, 13, 13);"><b>MR.GB
Transports<b></a>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNavAltMarkup"
        aria-controls="navbarNavAltMarkup" aria-expanded="false" aria-label="Toggle
navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <style>
      body {
        margin: 0;
        padding: 0;
        font-family: Arial, sans-serif;
      }
    .homepage {
      background-image:
url("C:\Users\admin\Music\myproject\myapp\static\mysite\images\bg.jpg");
      background-size: cover;
      background-position: center;
```

```
            height: 100vh;
            display: flex;
            flex-direction: column;
            justify-content: center;
            align-items: center;
            text-align: center;
            color: #ffffff;
        }
        header {
            margin-bottom: 20px;
        }
        h1 {
            font-size: 3cm;
            text-shadow: 2px 2px 4px rgba(0,0,0,0.5);
        }

        </style>
        <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
            <div class="navbar-nav" >
                <a class="nav-item nav-link active" href="{% url 'home' %}" style="color: rgb(14, 12, 12);">Home<span class="sr-only">(current)</span></a>
                <a class="nav-item nav-link" href="{% url 'findbus' %}" style="color: rgb(14, 10, 10);">Find Bus</a>
                <a class="nav-item nav-link" href="{% url 'seebookings' %}" style="color: rgb(14, 7, 7);">See Bookings</a>
                {% if request.user.is_active %}
                <a class="nav-item nav-link " href="{% url 'signout' %}" style="color: rgb(10, 7, 7);">Sign out</a>
                {% else %}
                <a class="nav-item nav-link " href="{% url 'signup' %}" style="color: rgb(12, 9, 9);">Sign Up</a>
                {% endif %}


            </div>
        </div>
    </div>
</nav>
{% block content %}
{% endblock %}
<!-- Optional JavaScript -->
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
```

```
    integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
    crossorigin="anonymous"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"
    integrity="sha384-
ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIPm49"
    crossorigin="anonymous"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
    integrity="sha384-
ChfqqxuZUCnJSK3+MXmPNIyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy"
    crossorigin="anonymous"></script>
</body>
</html>
```

SIGNIN.HTML

```
{% extends 'myapp/base.html' %}
{% block content %}
<div class="container">
    <center><h2 style="margin-top: 70px; background-color: rgb(56, 178, 248);color:
white">Welcome to Bus Ticket Reservation</h2></center>
    <div class="row">
      <div class="col-sm-6 mx-auto" style="margin-top: 70px">
        <h2>Log in</h2>

        <form action="{% url 'signin' %}" method="post">
          {% csrf_token %}
          <div class="form-group row">
            <label for="example-email-input" class="col-2 col-form-
label">Username:</label>
            <div class="col-10">
              <input name='name' class="form-control" type="text" id="example-email-
input">
            </div>
          </div>
          <div class="form-group row">
            <label for="example-email-input" class="col-2 col-form-
label">Password:</label>
            <div class="col-10">
              <input name='password' class="form-control" type="password">
            </div>
          </div>
          <div class="pull-right">
```

```
                        <button type="Submit" class="btn btn-success float-right">Sign in</button>
                    </div>

                </form>
                <p>{{error}}</p>
            </div>
        </div>

    </div>

{% endblock %}
```

SIGNUP.HTML

```
{% extends 'myapp/base.html' %}
{% block content %}
<div class="container">
        <div class="row">
            <div class="col-sm-6 mx-auto" style="margin-top: 70px">
                <form action="{% url 'signup' %}" method="post">
                    <h2>Sign up!</h2>
                    {% csrf_token %}
                    <div class="form-group row">
                        <label for="example-email-input" class="col-2 col-form-label">Email:</label>
                        <div class="col-10">
                            <input name='email' class="form-control" type="text" id="example-email-input" required>
                        </div>
                    </div>
                    <div class="form-group row">
                        <label for="example-email-input" class="col-2 col-form-label">Username:</label>
                        <div class="col-10">
                            <input name='name' class="form-control" type="text" required>
                        </div>
                    </div>
                    <div class="form-group row">
                        <label for="example-email-input" class="col-2 col-form-label">Password:</label>
                        <div class="col-10">
                            <input name='password' class="form-control" type="text" required>
                        </div>
                    </div>
                    <div class="pull-right">
                        <button type="submit" class="btn btn-success float-right">Sign up</button>
                    </div>

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

</div>

{% endblock %}
```

BOOKINGS.HTML

```
{% extends 'myapp/base.html' %}
{% block content %}
<h2>Booking Confirmation</h2>
<form action="{% url 'home' %}" method="post">

    {% csrf_token %}
    <h2>Your booking has been confirmed!</h2>
    <h2>Thank you!</h2>
    <h3>Bill details</h3>
    <!-- Button to Open the Modal -->
    <button type="button" class="btn btn-success" data-toggle="modal" data-
target="#myModal">
        Bill details
    </button>

    <!-- The Modal -->
    <div class="modal" id="myModal">
        <div class="modal-dialog">
            <div class="modal-content">

                <!-- Modal Header -->
                <div class="modal-header">
                    <h4 class="modal-title">Modal Heading</h4>
                    <button type="button" class="close" data-dismiss="modal">&times;</button>
                </div>

                <!-- Modal body -->
                <div class="modal-body">
```

```
          <ul class="list-group list-group-flush">
            <li class="list-group-item"><b>Bus name:</b> {{book.bus_name}}</li>
            <li class="list-group-item"><b>Starting point:</b> {{book.source}}</li>
            <li class="list-group-item"><b>Destination point:</b> {{book.dest}}</li>
            <li class="list-group-item"><b>Number of seats:</b> {{book.nos}}</li>
            <li class="list-group-item"><b>Price:</b> {{book.price}}</li>
            <li class="list-group-item"><b>Cost:</b> {{cost}}</li>
            <li class="list-group-item"><b>Date:</b> {{book.date}}</li>
            <li class="list-group-item"><b>Time:</b> {{book.time}}</li>
          </ul>


        </div>

        <!-- Modal footer -->
        <div class="modal-footer">
          <button type="button" class="btn btn-success" data-
dismiss="modal">Close</button>
        </div>

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

  <div class="pull-right">
    <button type="submit" class="btn btn-primary float-right">OK</button>
  </div>


</form>


{% endblock %}
```

BOOKLIST.HTML

```
{% extends 'myapp/base.html' %}
{% block content %}
<h3>{{msg}}</h3>
<h2>List of buses</h2>
<table class="table table-striped">
  <thead style="background-color: blue;color: rgb(238, 231, 231);">
  <td>BOOKING ID</td>
```

```
    <td>USER NAME</td>
    <td>BUS NAME</td>
    <td>SOURCE</td>
    <td>DESTINATION</td>
    <td>NUM OF SEATS</td>
    <td>PRICE</td>
    <td>DATE</td>
    <td>TIME</td>
    <td>STATUS</td>


    </thead>

    {% for row in book_list %}
    <tr>
        <td>{{row.id}}</td>
        <td>{{row.name}}</td>
        <td>{{row.bus_name}}</td>
        <td>{{row.source}}</td>
        <td>{{row.dest}}</td>
        <td>{{row.nos}}</td>
        <td>{{row.price}}</td>
        <td>{{row.date}}</td>
        <td>{{row.time}}</td>
        <td>{{row.status}}</td>


    </tr>
    {% endfor %}
</table>
<form action="{% url 'cancellings' %}" method="post">
    <h3>Choose bus to book</h3>
    {% csrf_token %}
    <div class="col-auto">
        <label for="example-email-input" class="col-2 col-form-label">Bus ID</label>
        <div class="col-5">
            <input name='bus_id' class="form-control" type="number" id="example-email-
input">
        </div>
    </div>

    <br>
    <br>
    <div class="pull-right">
```

```
        <button type="submit" class="btn btn-danger float-left">Cancel bus</button>
    </div>



    {{error}}

</form>

{% endblock %}
```

FINDBUS.HTML

```
{% extends 'myapp/base.html' %}
{% block content %}
<div class="container">
        <div class="row">
            <div class="col-sm-6 mx-auto" style="margin-top: 70px">
                <h2>Find bus</h2>
                <form action="{% url 'findbus' %}" method="post">
                    {% csrf_token %}
                    <div class="form-group row">
                        <label for="example-email-input" class="col-2 col-form-label">From</label>
                        <div class="col-10">
                            <input name='source' class="form-control" type="text" id="example-email-
input">
                        </div>
                    </div>
                    <div class="form-group row">
                        <label for="example-email-input" class="col-2 col-form-
label">Destination</label>
                        <div class="col-10">
                            <input name='destination' class="form-control" type="text">
                        </div>
                    </div>
                    <div class="form-group row">
                        <label for="example-email-input" class="col-2 col-form-label" >Date</label>
                        <div class="col-10">
                            <input name='date' class="form-control" type="date">
                        </div>
                    </div>
                    <div class="pull-right">
                        <button type="submit" class="btn btn-success float-right">Find bus</button>
                    </div>
                    {{error}}

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

    </div>

{% endblock %}
```

LIST.HTML

```
{% extends 'myapp/base.html' %}
{% block content %}
<h3>{{msg}}</h3>
<h2>List of buses</h2>
<table class="table table-striped" >
    <thead style="background-color: blue;color: white;">
    <td>ID</td>
    <td>NAME</td>
    <td>SOURCE</td>
    <td>DESTINATION</td>
    <td>NUM OF SEATS</td>
    <td>NUM OF SEATS REM</td>
    <td>PRICE</td>
    <td>DATE</td>
    <td>TIME</td>

    </thead>

    {% for row in bus_list %}
    <tr>
        <td>{{row.id }}</td>
        <td>{{row.bus_name}}</td>
        <td>{{row.source}}</td>
        <td>{{row.dest}}</td>
        <td>{{row.nos}}</td>
        <td>{{row.rem}}</td>
        <td>{{row.price}}</td>
        <td>{{row.date}}</td>
        <td>{{row.time}}</td>
```

```
  </tr>
  {% endfor %}
</table>
<form action="{% url 'bookings' %}" method="post">
  <h3>Choose bus to book</h3>
  {% csrf_token %}
  <div class="col-auto">
    <label for="example-email-input" class="col-2 col-form-label">Bus ID</label>
    <div class="col-5">
      <input name='bus_id' class="form-control" type="number" id="example-email-
input">
    </div>
  </div>
  <div class="col=auto">
    <label for="example-email-input" class="col-2 col-form-label">Number of
seats</label>
    <div class="col-5">
      <input name='no_seats' class="form-control" type="number">
    </div>
  </div>
  <br>
  <br>
  <div class="pull-right">
    <button type="submit" class="btn btn-success float-left">Book bus</button>
  </div>


  {{error}}

</form>

{% endblock %}
```

CHAPTER-6
TESTING

6.TESTING

Testing is vital to the success of the system. System testing makes a logical assumption that if all parts of the system are corrected, the goal will be successfully achieved. Inadequate testing non testing leads to errors that may not appear until months later.

The testing of this project ensures that the data received by the user is accurate. The project gives details of different books available in the book stall according to the customer wish. This is ensured in the testing. Testing should systematically uncover different classes of errors in a minimum amount of time with a minimum amount of efforts. Two classes of inputs are provided to test the process.

A software configuration that includes a software requirement specification, a design specification and source code. A software configuration that includes a test plan and procedure, any testing tool and test cases and their expected results.

Testing is divided into several distinct operations:

TYPES OF TESTING

UNIT TESTING

In computer programming, a unit test is a procedure used to validate that a particular module of source code is working properly. The procedure is to write test cases for all functions and methods so that whenever a change causes a regression, it can quickly be identified and fixed. Ideally, each test case is separate from others.

This project has so many modules like new book insertion, deletion, update of the book details. All the individual modules are tested and validated for checking whether it gives the desired output. After validating these modules, we can say that all the modules of these system are working perfectly and giving the desired output required by the administrator.

WHITE BOX TESTING

Integration testing (sometimes called integration and testing and abbreviated I&T) is the phase of software testing in which individual software modules are combined and tested as a group. It follows unit testing and precedes system testing. Integration testing takes as its input values that have been unit tested, groups them in larger aggregates, applies tests defined in and integrated test plan to those aggregates and delivers as its output the integrated system ready for system testing the purpose of integration testing is to verify functional performance and reliability requirements placed in major design items.

Form validation involves checking all the form constraints like arithmetic, syntax, logical errors. Database validation involves checking constraints like primary key, foreign key, and all the database validations avoiding data redundancy. All the modules of the system are tested and now the modules are combined together and the integrated module is tested and validated for checking whether the combined modules are working perfectly or not.

BASIC PATH TESTING

Validation testing is a concern which overlaps with integration testing ensuring that the application fulfils its specification, is a major criterion of integration testing. Validation, testing also overlaps to a large extend with system testing, where the application is tested with respect to its typical working environment.

Consequently, for many processes no clear division between validation and system testing can be made validation testing.

Specific tests can be performed in either or both stages include the following:

CONDITIONAL TESTING

where this version of the software is tested with the automated test harnesses, used with previous versions to ensure that the required features of the previous versions are still working in the new version.

DATA FLOW TESTING

where the software is deliberately interrupted in a number of ways, for example taking its hard disk off line or even turning the computer off, to ensure that the appropriate techniques for restoring any lost data will function. It is a system that forces the software to fail in a variety of ways and verifies that the recovery is properly performed.

where unauthorized attempts to operate the software or part of it, are attempted. It might also include attempts to obtain access the data or harm the software installations or even the system software. A with all types of security it is recognized that someone sufficiently determined will be able to obtain unauthorized access and the best that can be achieved is to make this process as difficult as possible. It attempts to verify that protection mechanisms built into a system will in fact protect it from improper penetration. The system's security must of course be tested from in vulnerability form frontal attack.

LOOP TESTING

where abnormal demands are made upon the software by increasing the rate at which it is asked to produce information. More complex tests may attempt to create very large data sets or cause the software to make excessive demands on the operating system. Stress tools are designed to confront programs with abnormal situations. Stress testing executes a system in a manner that demands resources in abnormal quantity and volume.

PERFORMANCE TESTING

Performance requirements, if any, are checked. These may include the size of the software when installed the amount of main memory and/or secondary storage it requires and the demands made of the operating system when running within normal limits of the response Time.

BLACK BOX TESTING

Black box testing is done to find out the following information as shown in below:

1. Incorrect or missing functions.
2. Interface errors.
3. Errors or database access.
4. Performance error.
5. Termination error.

The mentioned testing is carried out successfully for this application according to the user's requirement specification.

TEST DATA OUTPUT

After preparing test data, the system under study is tested using the test data. While testing the system using test data, errors are again uncovered and corrected by using above testing and corrections are also noted for future use.

IMPLEMENTATION

Implementation is the process of converting the system design into code, testing the system and giving the user training. Implementation of a new system design is a crucial phase in the system development life cycle.

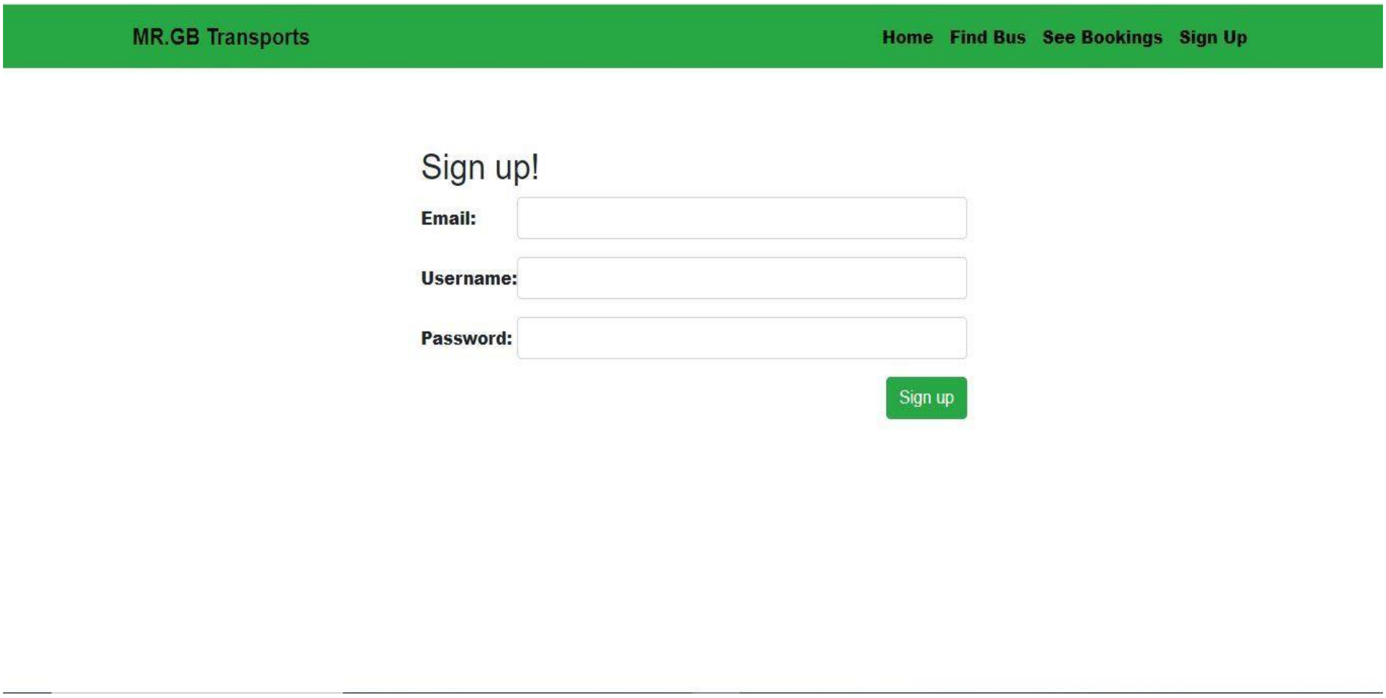
TESTING TABLE

ID	TEST CASE	EXPECTED OUTPUT	ACTUAL OUTPUT	CONCLUSION
01	To show main page.	Any user can access this page and view the list of main sections provided on the platform.	User can access this page and view the list of main sections provided on the platform properly.	This function works effectively.
02	To give the users permission to view the login page and enter their details.	Any user can view the login page and input their details based on their user type/role.	Users can view the login page and input their details properly.	This function works effectively.
03	To give the users permission to view the registration page and enter their details.	Any user can view the registration page.	Users can view the registration page and input their details properly.	This function works effectively.

CHAPTER-7
SCREENSHOTS

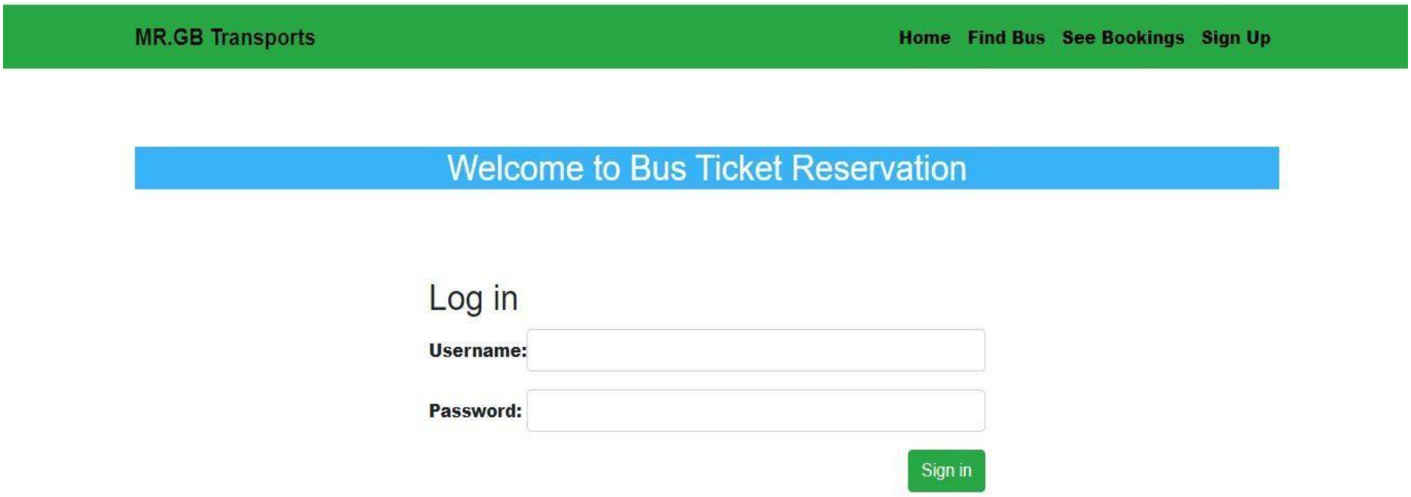
7.SCREEN SHOTS

7.1 SIGNUP PAGE



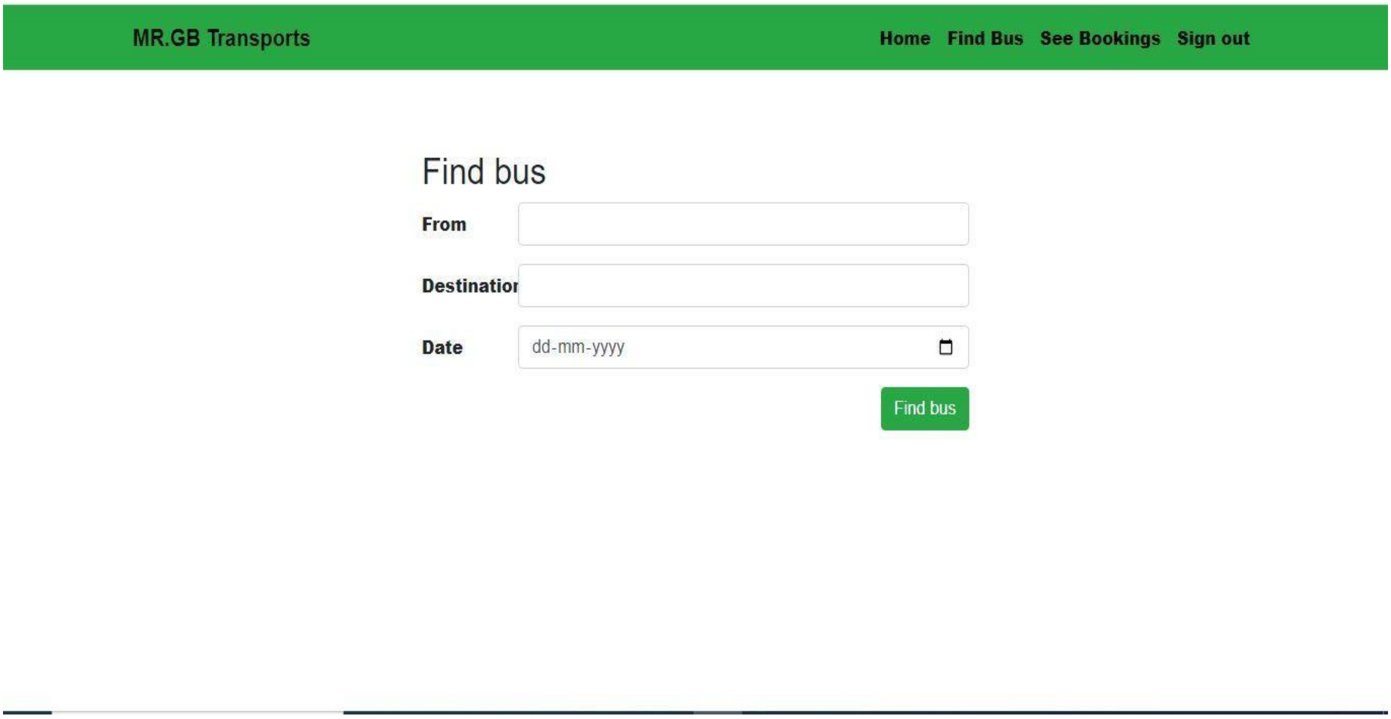
Description:- New user will be able to sign up by giving his/her personal details.

7.2 LOGIN PAGE



Description:- Registered user can login by giving his/her user name and password.

7.3 FINDING BUS PAGE



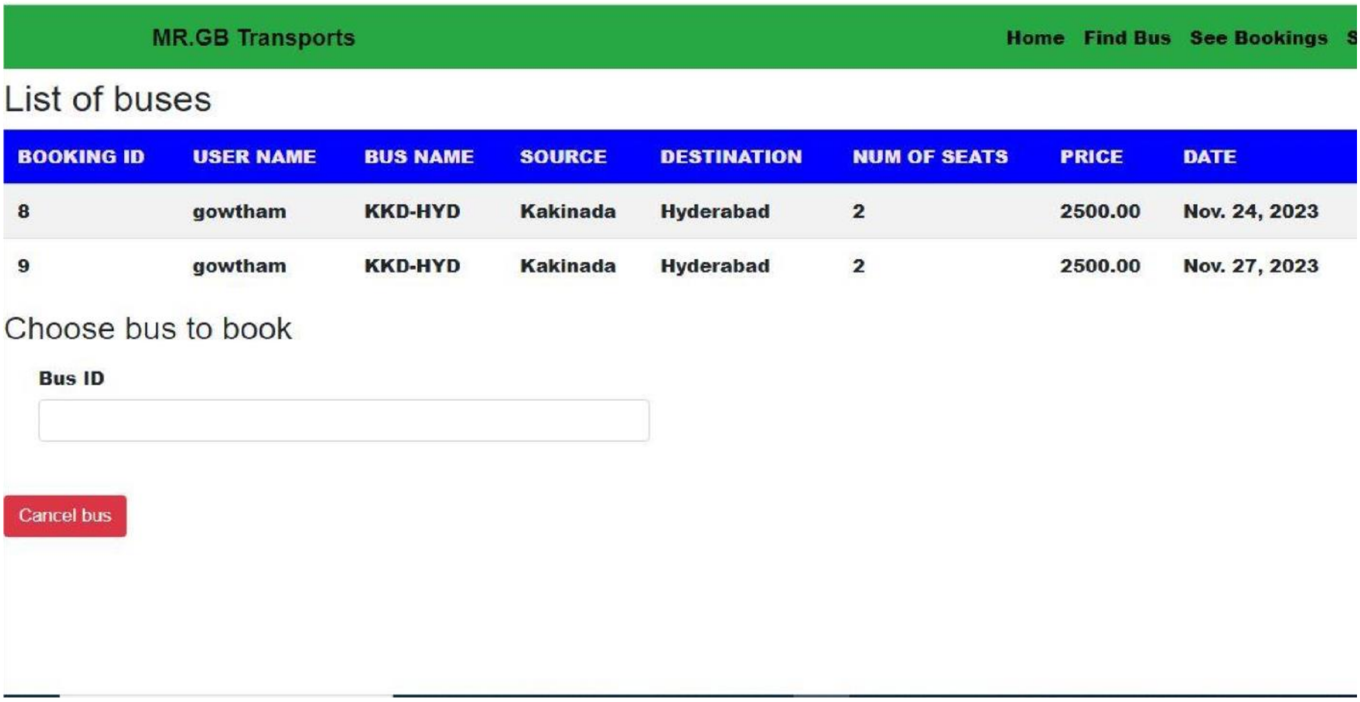
From

Destinatior

Date dd-mm-yyyy

Description:- User can search bus by entering the date and location.

7.4 BOOKINGS PAGE



List of buses

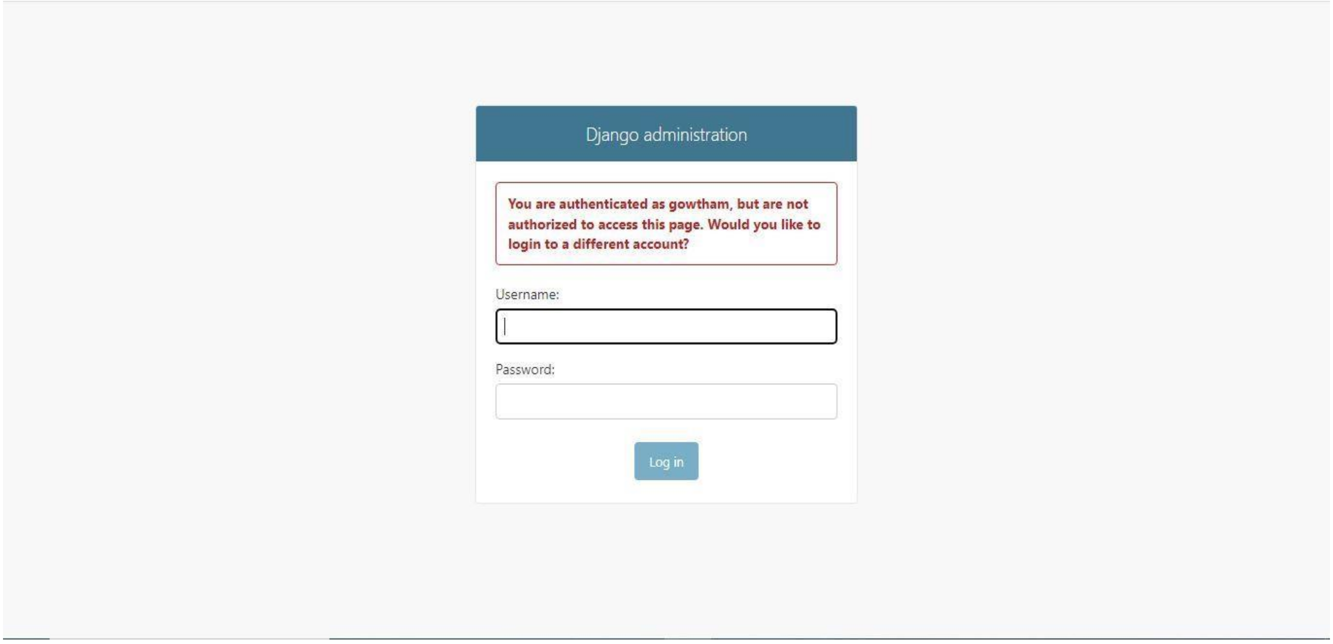
BOOKING ID	USER NAME	BUS NAME	SOURCE	DESTINATION	NUM OF SEATS	PRICE	DATE
8	gowtham	KKD-HYD	Kakinada	Hyderabad	2	2500.00	Nov. 24, 2023
9	gowtham	KKD-HYD	Kakinada	Hyderabad	2	2500.00	Nov. 27, 2023

Choose bus to book

Bus ID

Description:- Here we can see the total bookings done by user.

7.5 ADMIN LOGIN PAGE



Django administration

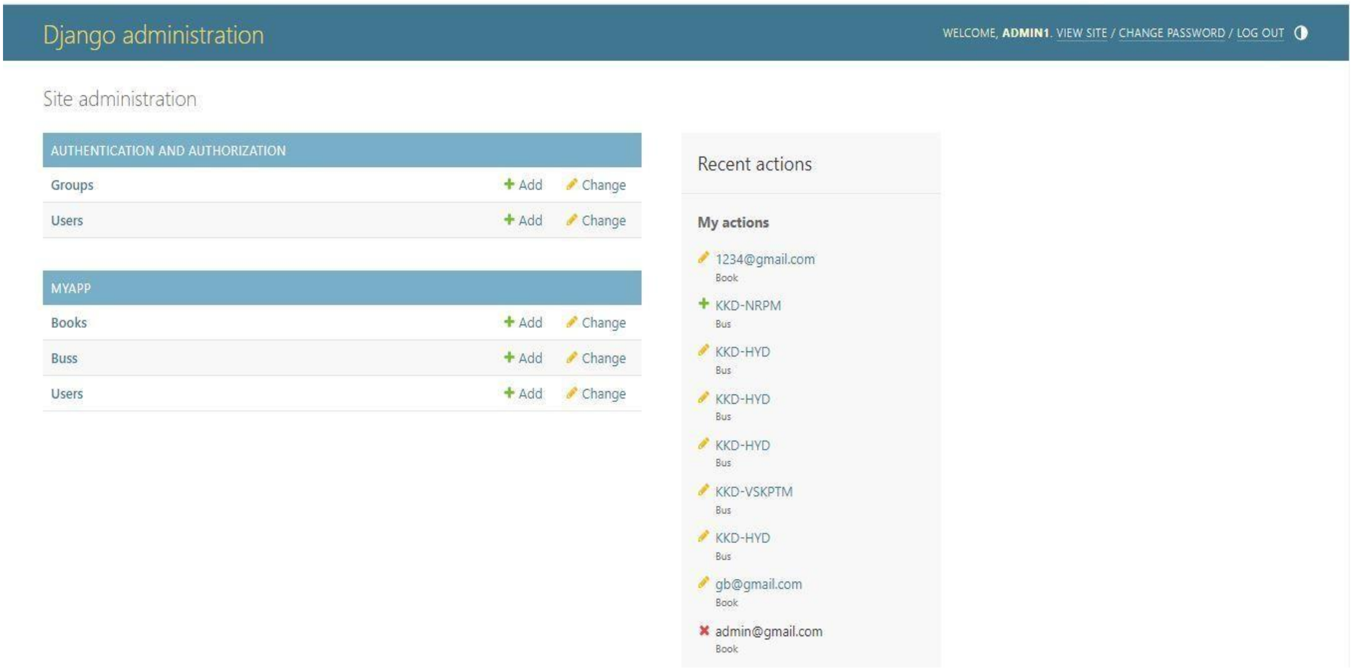
You are authenticated as gowtham, but are not authorized to access this page. Would you like to login to a different account?

Username:

Password:

Description:- It is a limited access area where only admin can login by giving his/her access credentials.

7.6 ADMIN PAGE



Django administration

WELCOME, **ADMIN1**. VIEW SITE / CHANGE PASSWORD / LOG OUT

Site administration

AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	✎ Change
Users	+ Add	✎ Change

MYAPP		
Books	+ Add	✎ Change
Buss	+ Add	✎ Change
Users	+ Add	✎ Change

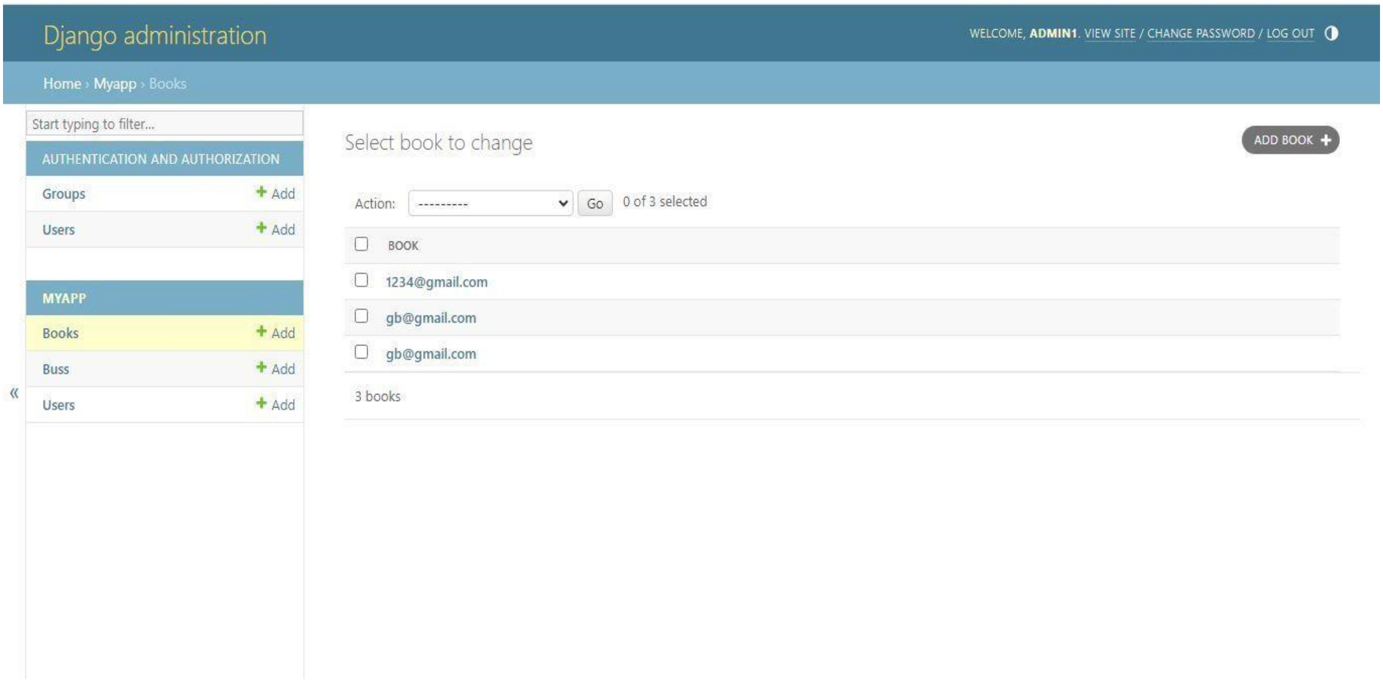
Recent actions

My actions

- ✎ 1234@gmail.com
Book
- + KKD-NRPM
Bus
- ✎ KKD-HYD
Bus
- ✎ KKD-HYD
Bus
- ✎ KKD-HYD
Bus
- ✎ KKD-VSKPTM
Bus
- ✎ KKD-HYD
Bus
- ✎ gb@gmail.com
Book
- ✖ admin@gmail.com
Book

Description:- Interface of entire backend administration data.

7.7 TOTAL BOOKINGS PAGE



Select book to change

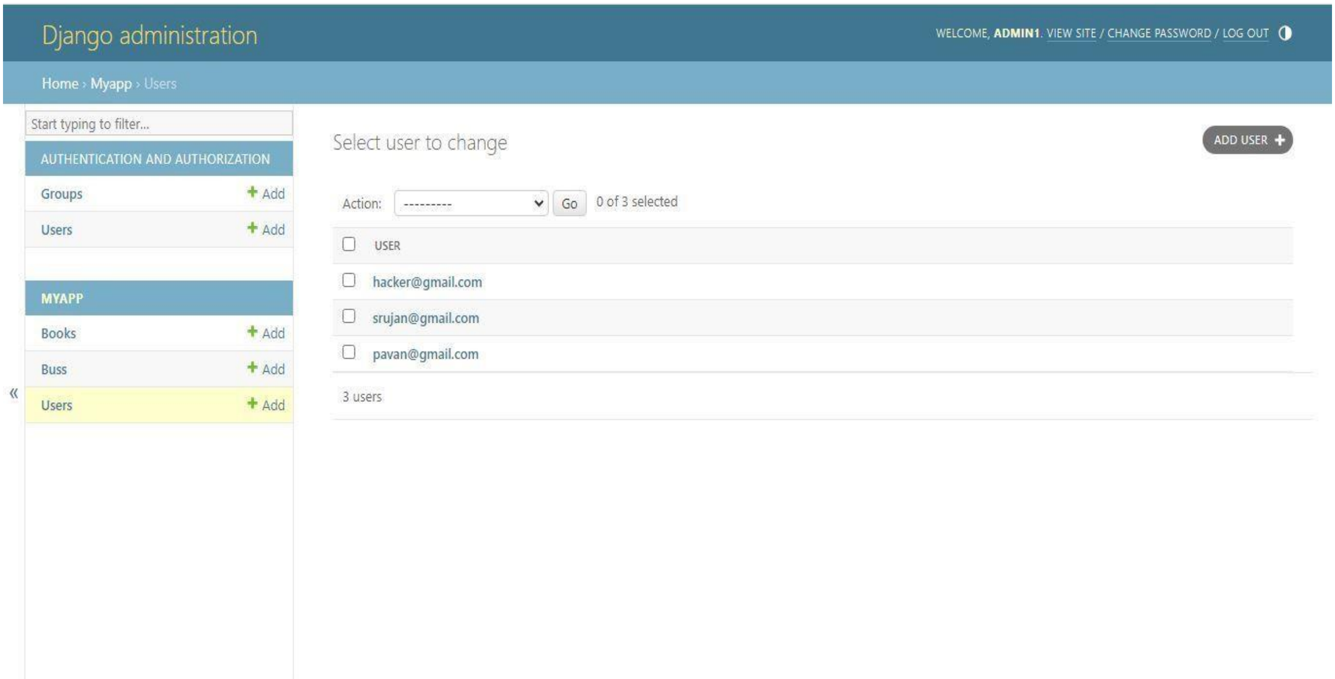
0 of 3 selected

□	BOOK
□	1234@gmail.com
□	gb@gmail.com
□	gb@gmail.com

3 books

Description:- Total number of bookings done by different users.

7.8 USERS PAGE



Select user to change

□	USER
□	hacker@gmail.com
□	srujan@gmail.com
□	pavan@gmail.com

3 users

Description:- Total number of registered users.

CONCLUSION

CONCLUSION

Although the overall system will bring some advantages for EZB, some of the facts needed to be considered. As we said earlier the system is not going to available for 24*7 and there will be a limited operation hour, EZB may lose some customers in future due to the booking time restriction. If the internet connection become down then the system will halt and no offline data can be accessed through it. Changing customers mind wont effect the system as there will be no refund after booking.

It can be observed that computer applications are very important in every field of human endeavour. Here all the information about customer that made reservation can be gotten just by clicking a button with this new system, some of the difficulties encountered with the manual system are overcome. It will also reduce the workload of the staff, reduce the time used for making reservation at the bus terminal and also increase efficiency. The application also has the ability to update records in various files automatically thereby relieving the company's staff the stress of working from file security of data.

This project, as a whole, will give a new way in bus reservations and ticketing processes. The automation and management of seats and reservations will be done online. However, this project does not limit the walkin passengers that is passengers who visit the company's counter because it also caters for them. This also lessens the use of papers like in the traditional way of ticketing.

BIBLIOGRAPHY

TEXT BOOKS :

1. “Python Crash Course” by Eric Matthes: For a solid introduction to Python programming.

2. "Fluent Python" by Luciano Ramalho: To deepen your understanding of Python's features and best practices.

WEBSITES :

https://www.w3schools.com
https://www.python.org
https://www.upgrad.com