A Project report

On

**‘Railway Ticket Reservation System’**

Submitted in partial fulfillment of the requirement of

Project- (BCA544CO)

Of

Bachelor in Computer Application (BCA)

**Submitted To:**



**Purbanchal University**

Biratnagar, Nepal

**Submitted By:**

Sujit Manandhar (381958)

Subash Rana Magar (382598)

**KANTIPUR CITY COLLEGE**

Putalisadak, Kathmandu

27th August, 2022

A Project report

On

**‘Railway Ticket Reservation System’**

Submitted in partial fulfillment of the requirement of

Project (BCA544CO)

Of

Bachelor of Computer Application

**Submitted to:**

**Purbanchal University**

Biratnagar, Nepal

**Submitted By:**

Sujit Manandhar (372060)

Subash Rana Magar (382598)

**Under the Guidance of**

Ashim KC

Asst. Professor

**KANTIPUR CITY COLLEGE**

Putalisadak, Kathmandu

27th August, 2022

# ACKNOWLEDGEMENTS

The project Railway Ticket Reservation report has been prepared on the basis of the academic requirement of the Internship (BCA415CO) program of Bachelor of Computer Application (BCA) under faculty of Science and Technology affiliated Purbanchal University.

We would like to express our deepest appreciation to our lecturer, **Mr. Ashim KC** for providing proper guidance during the completion of our project. We are hearty thankful for his supervision towards us and for his invaluable advice and guidance in the process of preparing this report. We would like to express our warm gratefulness to our college friends for their support to provide the reference to complete this report in the time frame.

We would also like to share special thanks to all the collagenous teachers for all the guidance that we needed for this report. They have always pointed out our mistakes and provided us with insightful feedbacks to further improve our report.

And last, we would like to thank all our friends and others who helped us indirectly throughout this project.

Sujit Manandhar (381958)

Subash Rana Magar (382598)

# SUPERVISOR’S RECOMMMENDATION

I hereby recommend that this project report prepared under my supervision by Mr. Sujit Manandhar and Mr. Subash Rana Magar in partial fulfillment of the requirement for the degree of Bachelor of Computer Application (BCA) of Purbanchal University be processed for the evaluation.

**………………………..**

**Mr. Ashim KC**

**Assistant Professor**

**IT Department**

# ABSTRACT

The Railway Ticket Reservation System is complete Railway Reservation Mini Project designed in “Django Framework” as a simple web application. This project contains Introduction to the Railway reservation system. In our country Nepal, where there are very few counters for the reservation of the seats and one can get hard time getting tickets booked this project is for everyone to easily book a ticket. It is a computerized system of reserving the seats of train seats in advanced.

The railway reservation system facilitates the passengers to enquiry about the trains available on the basis of source and destination, booking and cancellation of tickets, enquiry about the status of the booked ticket, etc. The aim of case study is to design and develop a data base maintaining records of different trains, train status and passengers. This project contains introduction to the railways reservation system. It is the computerized system of reserving the seats of train seats in advance. It is mainly used for a long route. Online reservation has made the process for the reservation of seats very much easier than ever before.

Our website has various kinds of information that helps regarding booking of tickets via railways. Users will be able to search the train availability, the exact fare, the arrival and departure time of the train and they can also book the ticket by using the debit, credit or master card and after booking the ticket if the user want to cancel it then they can easily do it also.

The source code is to be compiled in Visual Studio Code. This is designed to run on all browsers.

Table of Contents

[ACKNOWLEDGEMENTS iii](#_Toc112450874)

[SUPERVISOR’S RECOMMMENDATION iv](#_Toc112450875)

[ABSTRACT v](#_Toc112450876)

[List of Abbreviations ix](#_Toc112450877)

[List of Figures x](#_Toc112450878)

[List of Tables xi](#_Toc112450879)

[CHAPTER 1: INTRODUCTION 1](#_Toc112450880)

[1.1 Background 1](#_Toc112450881)

[1.2 Objectives 2](#_Toc112450882)

[1.3 Significance 3](#_Toc112450883)

[1.4 Group Member Roles and Responsibilities 3](#_Toc112450884)

[1.5 Organization of Documents 4](#_Toc112450885)

[CHAPTER 2: EXISTING SYSTEM OVERVIEW 5](#_Toc112450886)

[2.1 Existing System 5](#_Toc112450887)

[2.2 Proposed System 6](#_Toc112450888)

[CHAPTER 3: SYSTEM ANALYSIS 7](#_Toc112450889)

[3.1 System Development Model 7](#_Toc112450890)

[3.1.1 Requirement Analysis 7](#_Toc112450891)

[3.1.2 Software Design 8](#_Toc112450892)

[3.1.3 Coding 8](#_Toc112450893)

[3.1.4 Testing 8](#_Toc112450894)

[3.1.5 Implementation 8](#_Toc112450895)

[3.2 Requirement Specification 9](#_Toc112450896)

[3.2.1 Functional Requirements 9](#_Toc112450897)

[3.2.2 Non-Functional Requirements 9](#_Toc112450898)

[3.3 Feasibility Study 10](#_Toc112450899)

[3.3.1 Economic Feasibility 10](#_Toc112450900)

[3.3.2 Technical Feasibility 11](#_Toc112450901)

[3.3.3 Operational Feasibility 11](#_Toc112450902)

[3.3 Gantt Chart 11](#_Toc112450903)

[Colored Boxes 11](#_Toc112450904)

[Working Days 11](#_Toc112450905)

[Uncolored Boxes 11](#_Toc112450906)

[Non-working days 11](#_Toc112450907)

[Chapter 4: System Design 12](#_Toc112450908)

[4.1 System Design 12](#_Toc112450909)

[4.2 Data Flow Diagram 13](#_Toc112450910)

[4.2.1 Context Diagram 13](#_Toc112450911)

[4.2.2 DFD Level 1 14](#_Toc112450912)

[4.2.3 DFD Level 2 15](#_Toc112450913)

[4.3 Use Case Diagram 16](#_Toc112450914)

[4.4 ER-Diagram 17](#_Toc112450915)

[CHAPTER 5: SYSTEM DEVELOPMENT AND IMPLEMENTATION 18](#_Toc112450916)

[5.1 Programming Platform 18](#_Toc112450917)

[5.2 Operating Environment 18](#_Toc112450918)

[5.2.1 Hardware Requirement 18](#_Toc112450919)

[5.2.2 Software Requirement 18](#_Toc112450920)

[5.3 Test Plan 19](#_Toc112450921)

[4.5.1 TEST CASE: 1 20](#_Toc112450922)

[4.5.2 TEST CASE: 2 20](#_Toc112450923)

[4.5.3 TEST CASE: 3 21](#_Toc112450924)

[4.5.4 TEST CASE: 4 21](#_Toc112450925)

[4.5.5 TEST CASE: 5 22](#_Toc112450926)

[4.5.6 TEST CASE: 6 22](#_Toc112450927)

[4.5.7 TEST CASE: 7 23](#_Toc112450928)

[4.5.8 TEST CASE: 8 23](#_Toc112450929)

[4.5.9 TEST CASE: 9 24](#_Toc112450930)

[4.5.10 TEST CASE: 10 24](#_Toc112450931)

[4.5.11 TEST CASE: 11 25](#_Toc112450932)

[4.5.12 TEST CASE: 12 25](#_Toc112450933)

[CHAPTER 6: CONCLUSION AND FUTURE ENHANCEMENT 26](#_Toc112450934)

[6.1 Summary 26](#_Toc112450935)

[6.2 Limitations 26](#_Toc112450936)

[6.3 Future Enhancement 27](#_Toc112450937)

[REFERENCES 28](#_Toc112450938)

[BIBLIOGRAPHY 29](#_Toc112450939)

[APPENDICES 30](#_Toc112450940)

# List of Abbreviations

|  |  |
| --- | --- |
| CSS | Cascading Style Sheet |
| HTML | Hyper Text Markup Language |
| JSON | JavaScript Script Notation |
| OS | Operating System |
| RAM | Random Access Memory |
| SQL | Structured Query Language |
| URL | Universal Resource Locater |
| UI | User Interface |
| UX | User Experience |
| CRUD | Create, Update and Delete |
| IT | Information Technology |

# 

# List of Figures

|  |  |
| --- | --- |
| Figure No. | Figure Name |
| Figure 1 | Waterfall Model |
| Figure 2 | System Design |
| Figure 3 | Gantt Chart |
| Figure 4 | Context Diagram |
| Figure 5 | DFD level 1 |
| Figure 6 | DFD level 2 |
| Figure 7 | Usecase Diagram |

# List of Tables

|  |  |
| --- | --- |
| Table No. | Table Name |
| Table 1 | Group Members |
| Table 2 | Organization of Document |
| Table 3 | Test Plan |
| Table 4 | Database connection testing |
| Table 5 | Modules migration testing |
| Table 6 | homepage testing |
| Table 7 | Login page testing |
| Table 8 | Registration page testing |
| Table 9 | Adding new rail into the database form the admin site |
| Table 10 | Logging Out Testing |
| Table 11 | Dashboard page test |
| Table 12 | Sending confirmation email to user on booking approval |
| Table 13 | Deleting the existing rail into the database form the admin site |
| Table 14 | Updating the existing rail into the database form the admin site |
| Table 15 | Adding new rail into the database form the admin site |

# CHAPTER 1: INTRODUCTION

## Background

In this world, train travel is at a disadvantage. Rail has been the primary mode of transportation for more over 100 years, but it is no longer competitive with air travel. You only need to think about how expectations for purchasing train tickets were influenced by the way we buy airline tickets if you have any doubts about the validity of this comparison.

In the context to Nepal, being vivid in bio diversity due to imbalance geographical areas, it has more issue over airways. And so far it is believed that Railway can be beneficial means of transportation in Nepal. The existing tall mountains with cooperation to the low sea level Terai regions of Nepal, the rail way travelling can be effective within the area of our country Nepal.

Railway Reservation System facilitates the passenger to enquire about the trains available on the basis of source and destination, Booking and Cancellation of tickets, enquire about the status of the booked ticket, etc. The aim of case study is to design and develop a database maintaining the record of different trains, train status and passenger. Our website has various kinds of information that helps regarding booking of tickets via railways.

Users will be able to search the train availability. The exact fare, the arrival and departure time of the train and they can also book the ticket by using the debit, credit or master card and after booking the ticket if the user want to cancel it then they can easily do it also. Our website has various kinds of information that helps regarding booking of tickets via railways.

Railway passengers frequently need to know about their ticket reservation status, ticket availability on a particular train or for a place, train arrival or departure details, special trains etc. Customer information centers at the railway stations are unable to serve such queries at peak periods. The number of the reservation counters available to the passengers and customers are very less. On most of the reservation systems there are long queues, so it takes a long time for any individual to book the ticket. As now there are no call centers facilities available to solve the queries of the passengers.

The online railway ticket reservation system aims to develop a web application which aims at providing trains details, trains availability, as well as the facility to book ticket in online for customers. So, we thought of developing a web-based application which would provide the users all these facilities from his terminal only as well as help them in booking their tickets. The Application was to be divided into two parts namely the user part, and the administrator part. And each of these has their corresponding features.

## 1.2 Objectives

Our project introduces a train reservation system with the goal of making it more efficient, simple, and quick. The goal of this project is to see how computer technology can be used to help users solve their problems. We can inquire about the trains that are available. We can change the information about a) Trains, 1) Timetable, 2) Train Name, 3) Train Number, and b) Ticket Fare, as well as reserve and cancel their seats. This project aims to develop a Railway Reservation System that allows railway customers to manage their reservations and the railway administrator to modify the backend database in a user-friendly manner by modeling existing railway reservation systems.

The objective of the online railway ticket reservation system Project is to design software to fully automate the process of issuing a railway ticket. Some of the major objectives are:

* To create a database of the trains
* To search the trains, it’s arrival and departure time, distance between source and destination and to estimate the time of journey
* To check the availability of the ticket
* To calculate fare
* To cancel the ticket if necessary

## 1.3 Significance

This website is mainly created to fulfil the following requirements, it comprises of the following properties:

* A central database that will store all information.
* An online website that will provide real- time information about the availability of tickets their prices.
* Every registered user is able to view his booking id that has been made in his/her name.
* Every registered user will receive an email after booking the ticket.
* Every guest user can search train availability, price of the ticket, arrival and departure time, distance between source and destination etc.
* The booking window contains all the facilities at one place, the user can simply login to his account and can book his ticket.
* Administration login:
  + In admin mode the administrator can make changes in train details.
  + They can also view all booking that has been made by different users.

## 1.4 Group Member Roles and Responsibilities

Table 1: Group Members and roles

|  |  |
| --- | --- |
| Member’s Names | Tasks Performed |
| Subash Rana Magar | Research, Coding |
| Documentation |
| Sujit Manandhar | Coding and Testing |
| Documentation |

## 1.5 Organization of Documents

We have organized this documentation in different parts, we have indexed all the topics in page “Table of contents”. We have explained the references in the last page. All the pages in our documentation were indexed with page number.

Table 2: Document organization

|  |  |
| --- | --- |
| Chapters | Headings |
| Chapter 1 | Introduction |
| Chapter 2 | Existing System Overview |
| Chapter 3 | System Analysis |
| Chapter 4 | System Design |
| Chapter 5 | System Development and Implementation |
| Chapter 6 | Conclusion |

# CHAPTER 2: EXISTING SYSTEM OVERVIEW

## 2.1 Existing System

Travelling has been one of the most important part for any living creature. With a glance over the world we can find several of means of travelling. Besides other ways of travelling Train travel has disadvantages in the modern world. Despite having been the main form of transportation for more than a century, rail is no longer as cost-effective as air travel. The accuracy of this comparison, is merely needed to consider how expectations for buying train tickets were influenced by the way we buy airline tickets.

*Lack of railway cooperation*. Buying a ticket for a connecting flight is easy. Airlines have interlining agreements with each other allowing passengers to seamlessly travel several flights, checking in once and getting rebooked in case of disruption. The lack of similar alliances among railways leads to less competition, lower passenger volumes, and minimal cross-border traveling.

*Lack of customer data*. Rail operators don’t know their customers. Because of the sporadic nature of train ticket shopping (people can purchase tickets online, in a kiosk at the station, or even on-board), you can’t create a holistic overview of your customers. This means that companies target a much generalized impression of a user and can’t accurately segment users and price their offers.

*Legacy technology.* Legacy systems are the bane of their existence for many traditional industries, railway included. They’re typically developed with outdated technology that has many limitations in today’s world; are expensive to manage and maintain, inclining developers to use workarounds and hacks to keep it alive; not developed from scratch, for example by customizing an airline reservation system; and bespoke, which creates inconsistency and stalls system-to-system connectivity.

Three of these problems (standardization, customer data, and technology) can be approached by implementing a modern, advanced rail reservation system.

## 2.2 Proposed System

 The railway reservation system is software that handles distribution, pricing, scheduling, and other railway operations. Most providers have some of these functions digitized, but as we mentioned, this is done sporadically. We want to talk about main modules of reservation systems and cover extra integrations to be on the lookout for Reservation and ticketing.

Railway transport considers the processes of ticketing and reservation to be one and the same since there’s no delay or distinction between booking and getting a train ticket, unlike in air transport. Ticketing functionality covers customer-facing features that make it convenient for customers to choose, pay for, use, and print tickets. Some of the most crucial ticketing implementations in a modern reservation platform include the following.

Advanced search capabilities. One of the biggest trends in travel sales today is providing capabilities not only for searching for a relevant option but helping customers plan their whole journey on your platform. Such a system collects more information about the customer to tailor results to their individual needs. This can be achieved using rules (for example, showing the fastest routes for customers who checked the “traveling on business” box) or smart algorithms that collect explicit and implicit data to make best-fit offers.

# CHAPTER 3: SYSTEM ANALYSIS

## 3.1 System Development Model

This project was developed following advance version of classical waterfall model i.e. “Iterative Waterfall Model”. The iterative waterfall model provides feedback paths from every phase to its preceding phases, which is the main difference from the classical waterfall model. Phases of iterative waterfall model are:

Fig 1: Waterfall Model

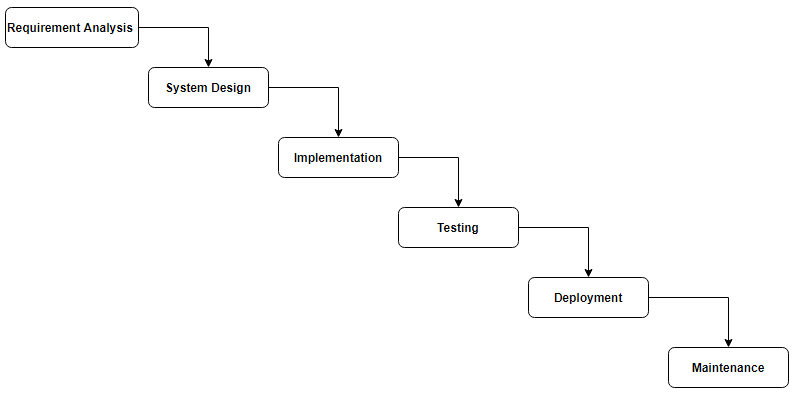


Fig: Waterfall Model

We chose iterative waterfall model as our project was small and simple. And since our project was easy to understand and we were able to go back on the previous phase and change the requirements and some modification could be done at any phases if necessary. Hence, we came to the conclusion that an iterative waterfall model would be the most optimum pick.

### 3.1.1 Requirement Analysis

The tools and resources required for developing this project were researched and gathered over internet. Also it was necessary to analyze time division for different development phases so the development would finish in time. Then features to be implemented in the system was decided and understand the structure for problem the software system is to solve.

### 3.1.2 Software Design

In designing Phase, we wanted to keep this project simple and easy to use since there was time limit and limited resources. So, the system interface was kept simple. Also, Design phase create a major impact on the later phases, particularly testing and maintenance. The problems occurs in other phases starting with what is needed, design takes us toward how to satisfy the needs. Resources for game design were gathered from internet.

### 3.1.3 Coding

This is the phase where software design is brought into reality for practical use. For this system programming, we had choose Python Django programming Language.

### 3.1.4 Testing

After coding part is done, comes testing. After running the code multiple times many errors and bugs were found during the system implementation. Finally when the system was successful and all functions were working properly, it was time to finalize project.

3.1.5 Implementation

Now the system is almost ready and we can simply reserve the ticket of the railway station without having worries.

## 3.2 Requirement Specification

### 3.2.1 Functional Requirements

Features of Railway Reservation System in Django in admin side:

* Login – To gain access to the system, the administrator must first log in.
* Manage Reservations – The admin has the ability to manage and view customer reservations.
* Train – The admin has the ability to add, amend, and delete information about the train.
* Station – The admin has the ability to add, amend, and delete information about the station.
* Routes – The admin has the ability to add, update, and delete information about routes.

Features of Railway Reservation System in Django in frontend side:

* Register – The customer must first register in order to create an account before registering.
* Login – After creating an account, the consumer must log in to gain access to the entire system.
* Search Trains – The user has the option of searching all of the available trains.
* Reservations – The consumer can make reservations for available trains.
* Cancellation of Reservations – The customer has the option to cancel his or her reservation.

### 3.2.2 Non-Functional Requirements

#### Performance

After login through the valid identity, individuals can easily get access over the available train routes and can book the tickets. There are no errors during the operation of the system. System gives good performance in a normal environment but may show some error if an undefined variable is introduced.

#### Availability and Compatibility

The system will be only available to the authorized users. Every user needs the authorized email address to login into the system. This system is compatible in most of the hardware but 2GB of Ram and 4GB of hard disk as well as Windows 10 and google chrome is preferred.

#### Security

Only the authorized user can access their account and use the application. The data is highly secured. PostgreSQL database is used to store the information of the system. Only the authorized user and admin can see and modify the data.

#### Usability

The purpose of this system is to be able to reserve the ticket for the railway travelling. As this system is very easy to understand and simple to use even for non-technical people. It consists of a simple interface hence, users can use this system without any prior knowledge.

## 3.3 Feasibility Study

Before even designing and developing the application there are many things to be considered. Therefore, a thorough study is done to ensure if the said project is possible or not. Hence, with respect to the goal of our project, we performed the feasibility study before starting the beginning phase of the project.

Depending on the result of the initial investigation the survey is expanded to a more detailed feasibility study. A feasibility study is a test of a proposal according to its workability, impact on the organization, ability to meet user needs, and effective use of resources. Various types of feasibility to be determined are:

### 3.3.1 Economic Feasibility

Development of this application is highly economically feasible. It is cost effective as creating new identification in the system to access their respective account doesn’t cost money. Sending messages and using the system doesn’t require any charges as it is free of cost.

### 3.3.2 Technical Feasibility

The technical feasibility for this system is economic and it does not require any other additional hardware and software. This application relies on the internet service by which the services of this system can run. Hence, it is easily technically feasible.

### 3.3.3 Operational Feasibility

This system is quite easy to use and learn due to its simple and easy to understand interface. Users require no special training for operating the system. They can easily login to their account and use the system. Hence, it is operationally feasible.

### 3.3 Gantt Chart

Fig 2: Gantt Chart

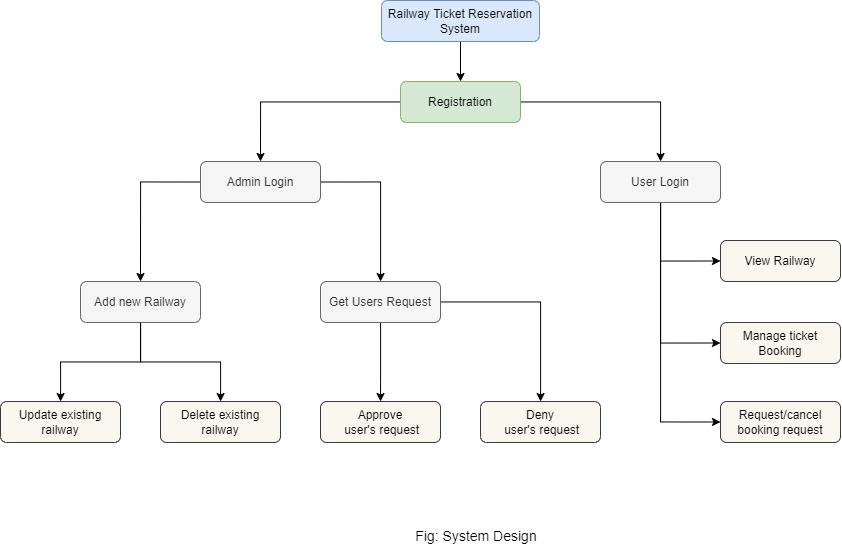
|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Month | | April | | | March | | | Jun/Jul | | | August |
| S.N | Work | 1 | 10 | 20 | 1 | 10 | 20 | 1 | 20-30 | 1-20 | 1-20 |
| 1 | Research & Analysis |  |  |  |  |  |  |  |  |  |  |
| 2 | Requirement Gathering |  |  |  |  |  |  |  |  |  |  |
| 3 | System Design |  |  |  |  |  |  |  |  |  |  |
| 4 | Coding |  |  |  |  |  |  |  |  |  |  |
| 5 | Testing |  |  |  |  |  |  |  |  |  |  |
| 6 | Documentation |  |  |  |  |  |  |  |  |  |  |

|  |  |
| --- | --- |
| Colored Boxes | Working Days |
| Uncolored Boxes | Non-working days |

# Chapter 4: System Design

## 4.1 System Design

Fig 3: System design



In the Railway ticket Reservation system, admin can access to the admin dashboard thought creating superuser through python command. On the other hand users are required to sign up through registration form. Through login page both with respective admin and user can get logged in into their respective dashboard. Admin should be able to access updating, adding and deleting the railways. Moreover admin should be able to handle users reservation requests during the system. On the other side, users/customers should be able to see available train ways and manage their bookings. The user should get message about the success on the ticket booked through the email.

## 4.2 Data Flow Diagram

This Railway Reservation System DFD Data Flow Diagram represents the flow of data and the transformations in the form of diagrams Level 0, 1 and 2. These transformations occurs as data enters and exits in a system. In the DFD, input, processing, and output are used to represent and define the overall system.

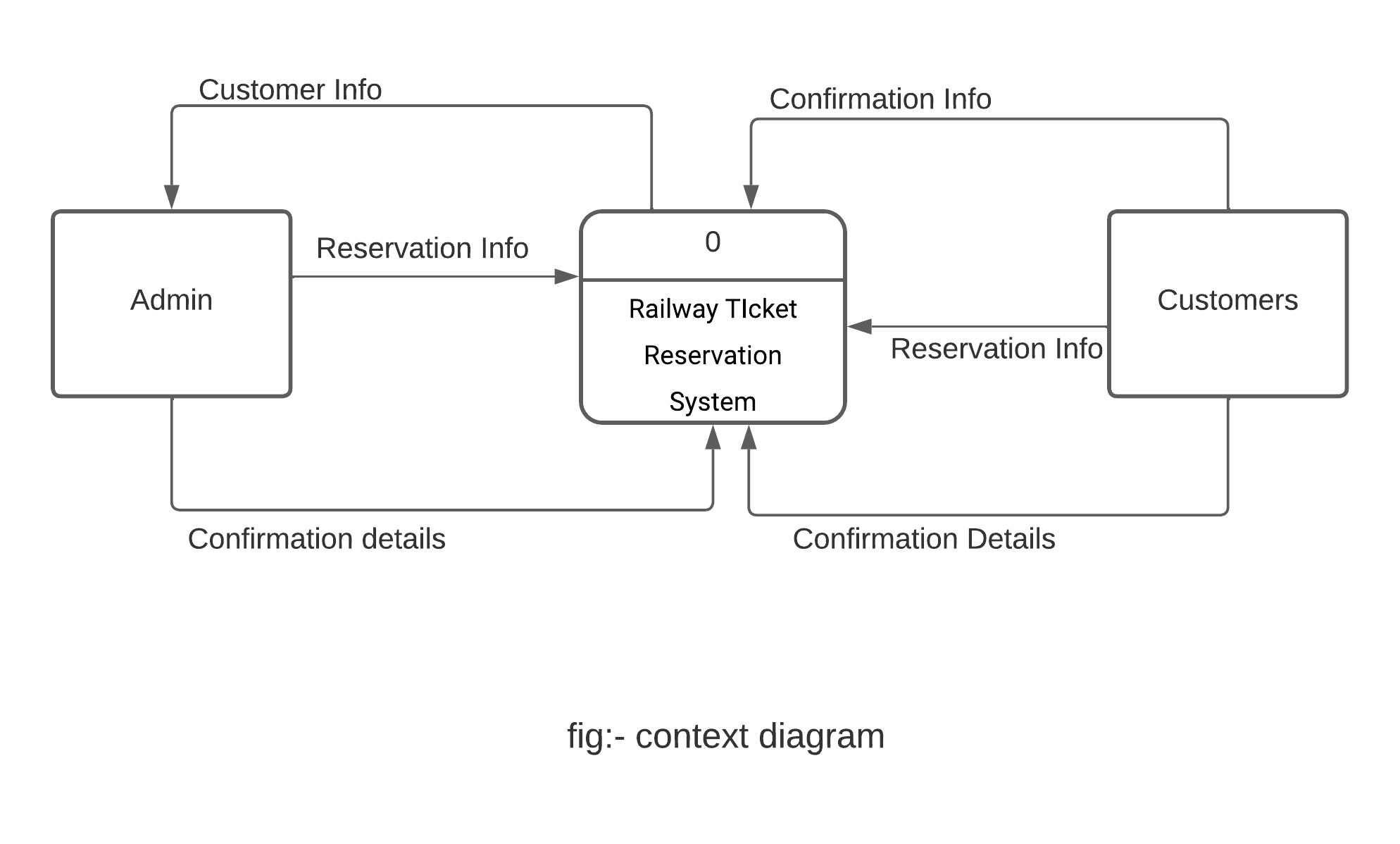
A [data flow diagram](https://meeraacademy.com/dfd-diagram-for-online-shopping-website/) is a graphical view of how data is processed in a system in terms of input and output. The Data flow diagram (DFD) can be represented with the following symbol for drawing the data flow diagram.

### 4.2.1 Context Diagram

The DFD Level 0 is also known as the context diagram of the system. The context diagram presents the main idea as the basis for the subsequent levels. The basic idea is represented by a single process consisting of the main process, users, and data.

Railway Reservation System Context diagram (Level 0)

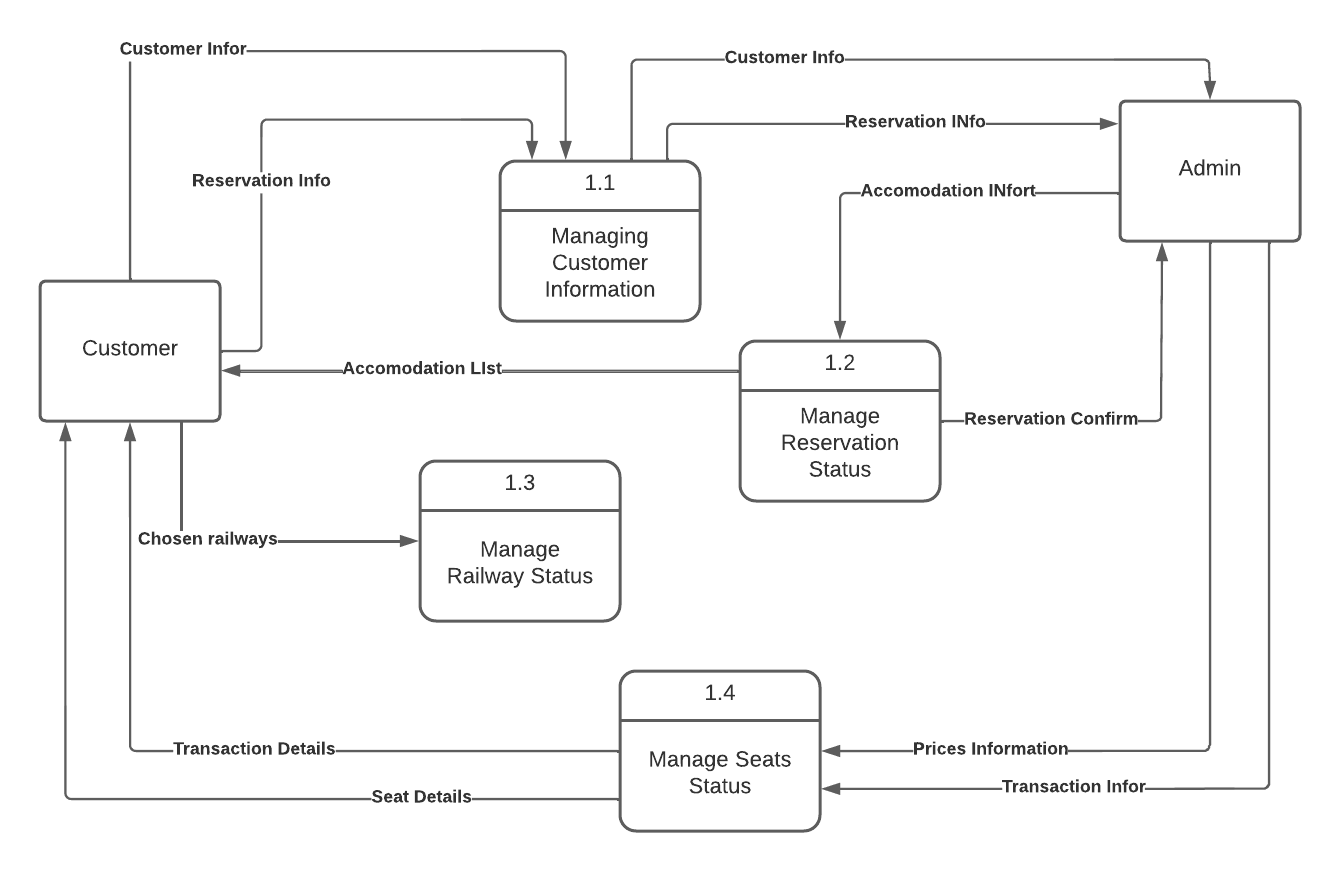
Fig 4: context diagram



### 4.2.2 DFD Level 1

These level 1 diagram shows how data flows through the system. In addition, its content comes from the previous level, which helps you understand the main process of the Railway Reservation System even more. In DFD level 1, the diagram also shows more information about how processing works.

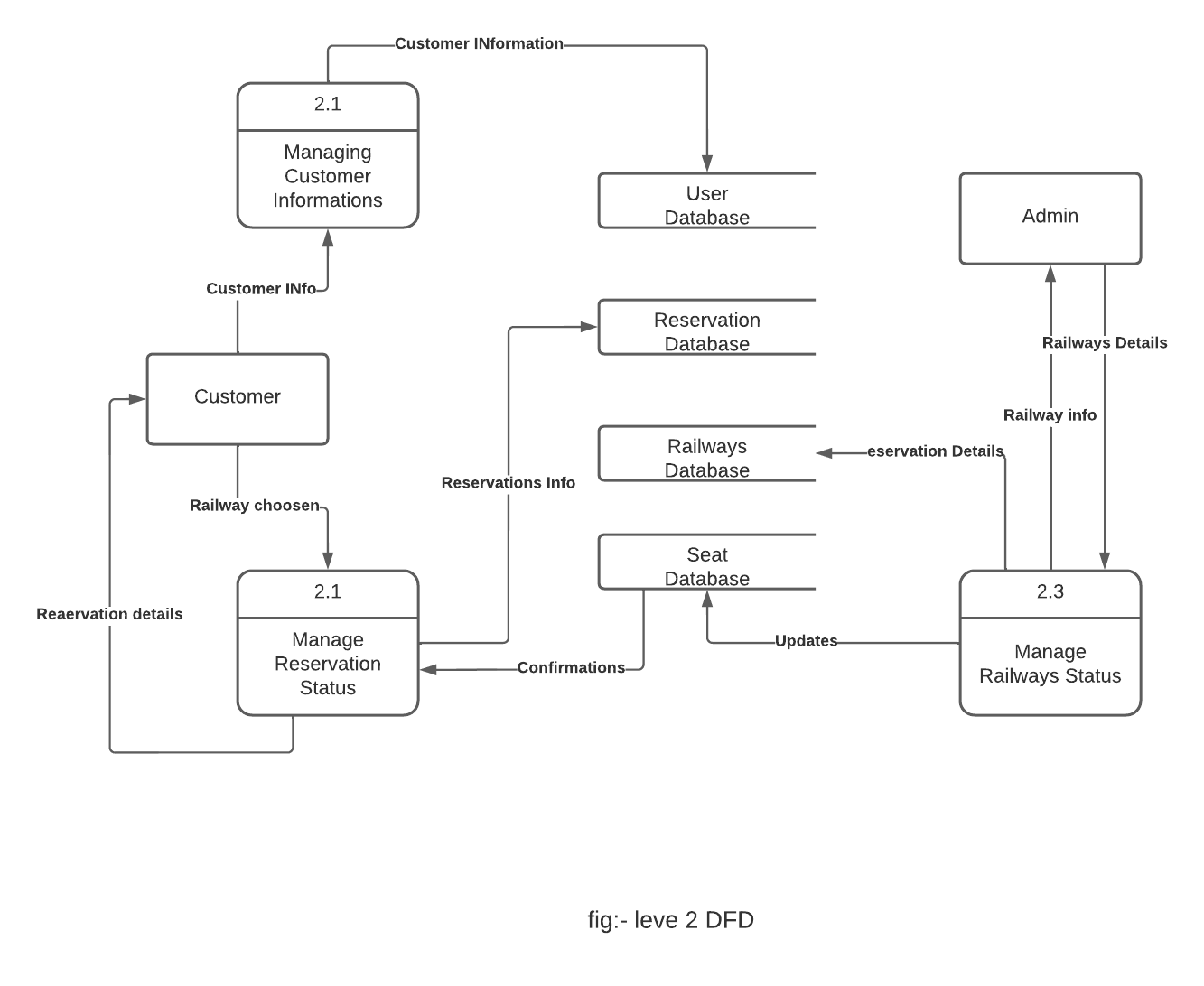
Fig 5: DFD level 1



## 4.2.3 DFD Level 2

DFD level 2 comes next after level 0 and 1. But creating the DFD Level 2 has two things to think about. DFD Level 2 should show all the processes and data in a more detailed way than level 1. As a result, it should also show the databases that are used.  
  
This diagram shows the database that stores all the information you need to finish the whole Railway Reservation System process. Furthermore, the main goal of DFD Level 2 is to show where all the data flows through the railway reservation system comes from and go to. Lastly, it is to show programmers on the project how the data flows inside it.

Fig 6: DFD level 2

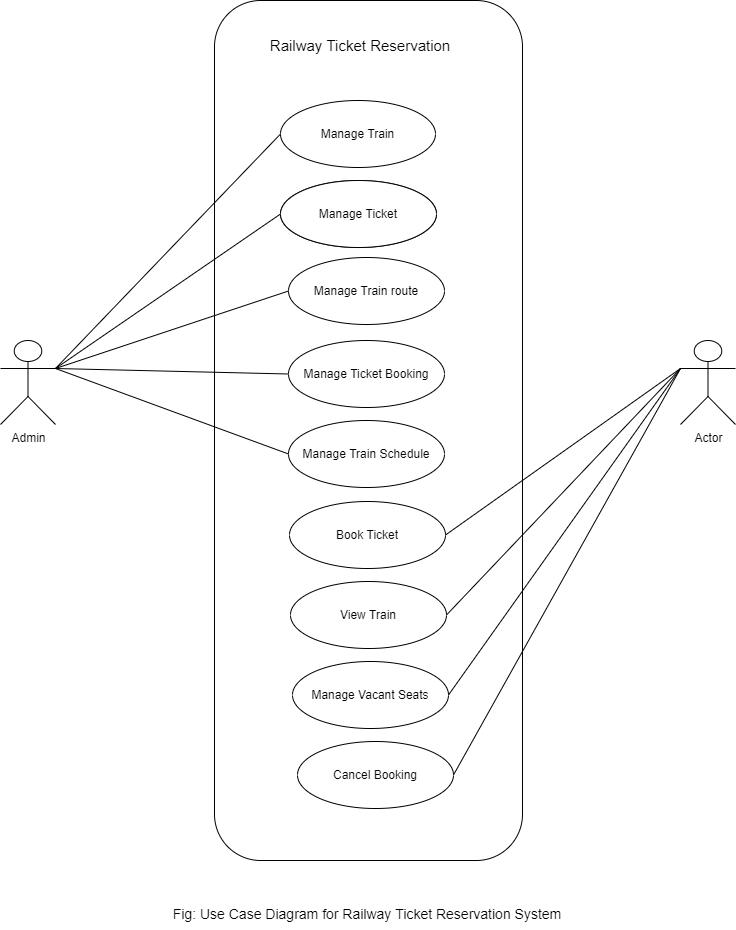


## 4.3 Use Case Diagram

Use case diagram are usually referred to as behavior diagram used to describe the actions of all user in a system.

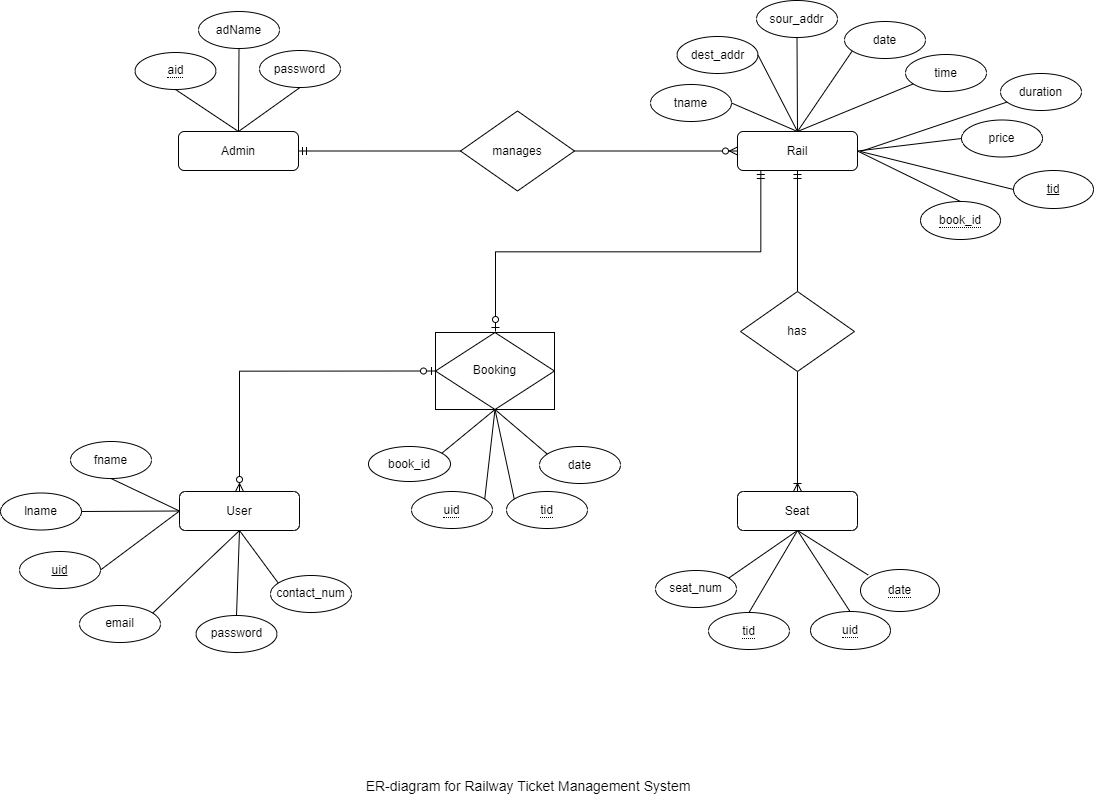
All user shown in use case are actors and the functionality as action of system. A use case is also often accompanied by other diagrams, as well. Circles or ellipses are used to depict the use cases.

Fig 7: Usecase Diagram



## 4.4 ER-Diagram

Fig 7: ER diagram



# CHAPTER 5: SYSTEM DEVELOPMENT AND IMPLEMENTATION

## 5.1 Programming Platform

Railway Ticket Reservation System is the name of the project, which is written in Python Django. The Python version used is 3.10.5. The database is PostgreSQL, the application is a web app.

Table 3: Programming Platform

|  |  |
| --- | --- |
| Project Name: | Railway Ticket Reservation System |
| Languages Used: | Python Django |
| Python version(Recommended): | 3.10.5 |
| Database: | PostgreSQL |
| Type: | Web Application |

## 5.2 Operating Environment

### 5.2.1 Hardware Requirement

* RAM : 2GB
* Hard Disk : 4GB of Available Hard Disk
* Processor : 1.6GHz or Faster Processor

### 5.2.2 Software Requirement

* Operating system: Windows 10
* Web browser : Google chrome (preferred)

## Test Plan

Table 4: Test Plan

|  |  |
| --- | --- |
| Test Cases | Objectives |
| 1 | To check whether the database was connected to the PostgreSQL server or not? |
| 2 | To check whether the modules was migrated to the PostgreSQL or not? |
| 3 | To check whether homepage/welcome page loads or not? |
| 4 | To check whether Login page loads or not? |
| 5 | To check whether Registration page loads or not? |
| 6 | To check whether new railway can be added or not? |
| 7 | To check whether user can log out or not? |
| 8 | To check whether dashboards loads or not? |
| 9 | To check whether user gets email or not on approving their reservation request? |
| 10 | To check whether the existing railway can be deleted or not? |
| 11 | To check whether the existing railway can be updated or not? |
| 12 | To check whether the existing railway can be updated or not? |

### 4.5.1 TEST CASE: 1

Table 5: Database connection testing

|  |  |
| --- | --- |
| Test Case | Database connection |
| Test objective | To check whether the database was connected to the PostgreSQL server or not? |
| Test data | Running pgAdmin4 |
| Excepted result | Database should connect successfully |
| Test result | Database connects properly |
| Conclusion | Expected result matches actual expectation |

### 4.5.2 TEST CASE: 2

Table 6: Modules migration testing

|  |  |
| --- | --- |
| Test case | Modules Migration |
| Test objective | To check whether the modules was migrated to the PostgreSQL or not? |
| Test data | Running the program |
| Expected result | All the modules should migrate |
| Test Result | All modules were migrated |
| Conclusion | Expected result matches actual result |

### 4.5.3 TEST CASE: 3

Table 7: homepage testing

|  |  |
| --- | --- |
| Test case | Homepage/welcome page |
| Test objectives | To check whether homepage/welcome page loads or not? |
| Test data | Running the program |
| Expected result | Homepage/welcome page should display properly |
| Test result | Homepage/welcome page appears properly |
| Conclusion | Expected result matches actual result |

### 4.5.4 TEST CASE: 4

Table 8: Login page testing

|  |  |
| --- | --- |
| Test case | Login page |
| Test objectives | To check whether Login page loads or not? |
| Test data | Running the program and giving credentials |
| Expected result | Login page should display properly |
| Test result | Login page appears properly |
| Conclusion | Expected result matches actual result |

### 4.5.5 TEST CASE: 5

Table 9: Registration page testing

|  |  |
| --- | --- |
| Test case | Registration page |
| Test objectives | To check whether Registration page loads or not? |
| Test data | Running the program and giving the new data |
| Expected result | Registration should be done properly |
| Test result | Registration was done properly |
| Conclusion | Expected result matches actual result |

### 4.5.6 TEST CASE: 6

Table 10: Adding new rail into the database form the admin site

|  |  |
| --- | --- |
| Test case | Adding a new railway |
| Test objectives | To check whether new railway can be added or not? |
| Test data | Running the program and adding new railway |
| Expected result | The new railway added should be recorded into the database. |
| Test result | The railway can be added successfully |
| Conclusion | Expected result matches actual result |

### 4.5.7 TEST CASE: 7

Table 11: Logging Out Testing

|  |  |
| --- | --- |
| Test case | Logging out test |
| Test objectives | To check whether user can log out or not? |
| Test data | Running the program |
| Expected result | User should log out and redirect to welcome page |
| Test result | Logs out properly |
| Conclusion | Expected result matches actual result |

### 4.5.8 TEST CASE: 8

Table 12: Dashboard page test

|  |  |
| --- | --- |
| Test case | Dashboard page |
| Test objectives | To check whether Dashboard page loads or not? |
| Test data | Running the program |
| Expected result | Dashboard page along with all the nav bar and sidebar should display properly |
| Test result | Dashboard page appears properly |
| Conclusion | Expected result matches actual result |

### 4.5.9 **TEST CASE**: 9

Table 13: Sending confirmation email to user on booking approval

|  |  |
| --- | --- |
| Test case | Approval of user request |
| Test objectives | To check whether user gets email or not on approving their reservation request? |
| Test data | Running the program and try to approve the user reservation request |
| Expected result | User should reserve booking confirmation mail |
| Test result | User gets the confirmation message form the admin on booking confirmed |
| Conclusion | Expected result matches actual result |

### 4.5.10 TEST CASE: 10

Table 14: Deleting the existing rail into the database form the admin site

|  |  |
| --- | --- |
| Test case | Deleting the existing railway |
| Test objectives | To check whether the existing railway can be deleted or not? |
| Test data | Running the program and delete the existing rail |
| Expected result | The existing railway should be deleted from the database. |
| Test result | The deleted railway in the admin site is removed from the database. |
| Conclusion | Expected result matches actual result |

### 4.5.11 TEST CASE: 11

Table 15: Updating the existing rail into the database form the admin site

|  |  |
| --- | --- |
| Test case | Updating the existing railway |
| Test objectives | To check whether the existing railway can be updated or not? |
| Test data | Running the program and update the existing rail |
| Expected result | The existing railway should be updated into the database. |
| Test result | The updated railway in the admin site is also updated into the database. |
| Conclusion | Expected result matches actual result |

### 

### 4.5.12 TEST CASE: 12

Table 16: Adding new rail into the database form the admin site

|  |  |
| --- | --- |
| Test case | Approving user reservation should s |
| Test objectives | To check whether the existing railway can be updated or not? |
| Test data | Running the program and update the existing rail |
| Expected result | The existing railway should be updated into the database. |
| Test result | The updated railway in the admin site is also updated into the database. |
| Conclusion | Expected result matches actual result |

# CHAPTER 6: CONCLUSION AND FUTURE ENHANCEMENT

## 6.1 Summary

In our country Nepal, though there isn’t a railway platform running effectively for the travelling services. Our project aims that in the coming near future we will have effective railway platform that the counters as well as user/customers can use this application for the reservation of the seats and get tickets books.

Railway reservation system, has described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus it will help organization it better utilization of resources. Administrator of the project, with the help of a password, can enter new train record, display all train records, modify train records and delete train records. The record of train includes its number, name, source, destination, and days on which it is available, whereas record of train status includes dates for which tickets can be booked, total number of seats available, and number of seats already booked.

## 6.2 Limitations

There is no way to reserve tickets for more than 6 persons each train using the software. The fee for each reservation is not based on the number of kilometers traveled, but rather on the number of miles traveled. For each train’s mode (AC, Non-AC, or General). The software is designed to allow for a maximum of 15 trains to be reserved at once. The program does not allow for multi-day reservations, which means that reservations cannot be made for more than one day. It is not done in advance, but rather for a single day. The program does not allow for fares to be reduced for children, the elderly, or military personnel. etc., i.e., the fare is the same for everyone. The software ignores the stations that are in between the source and destination stations.

## 6.3 Future Enhancement

The ultimate goal of our system is to help customers find train tickets. And sometimes, rail carriers need to take small steps to make it happen because some of the issues are more prosaic than we’re used to dealing with.

One final distinction between air and train travel is how it deals with all geographical regions of the Nepal. Since our system can be expanded through the enhancement of the railway transportation for that it requires cordial cooperation among all the areas of the country. Because people of Terai region might find other means of transportation better that the railway.

In addition, short train ways can be costlier that other means of transportation. So the railway routes and duration should be suited for long distance coverage and in our country it obviously indicates the requirement of all regions of people to be supportive towards the railway system.

.

# REFERENCES

College, K. C. (2020, January 3). Home: Kantipur City College. Retrieved from Kantipur

City College | Affiliated to PU: <http://kcc.edu.np/>

[Online], (2022, March 3). Introduction. Bootstrap. Retrieved from getbootstrap:

<https://getbootstrap.com/docs/5.0/getting-started/introduction/>

Channel, A.M (2021 June 2). & Common Challenges Faced by Interns – ToughNickel –

Money. Retrieved from toughnickel.com/business/Challenges-faced-by-interns-

Problems-commonly-found-in-internships.

Kantor, I. (2022, March 30). An Introduction to JavaScript. Retrieved from an Introduction

To JavaScript: <https://javascript.info/intro>

University, P. (2020, January 4). Home: Purbanchal University. Retrieved from Purbanchal

University: Office of the Examination Management, Biratnagar

<http://www.puexam.edu.np/>

[Online] w3Schools. W3Schools Online Web Tutorials. Retrieved from W3Schools:

<https://www.w3schools.com/>

[Online] w3School.Bootstrap 5 Tutorial. Retrieved from Bootstrap 5 Tutorial:

<https://www.w3schools.com/>bootstrap5/

# BIBLIOGRAPHY

Title: Pictures of Class description

Article Title: Class descriptions | STUDIOMAX

Website Title: STUDIOMAX

URL: <https://www.studiomax.com/class-descriptions>

Title: W3school

Article Title: W3School Online Web Tutorials

Website Title: W3schools.com

URL: <https://www.w3schools.com/>

Title: Research on Sites

Website Title: Themeforest.net

URL: https://themeforest.net/search/laravel

Title: Django Download and Details

Website Title: Laravel

URL: https://docs.djangoproject.com/en/4.1/

Title: Database Migration in Django

Website Title: Migration in Django

URL: https://docs.djangoproject.com/en/4.1/ref/databases/

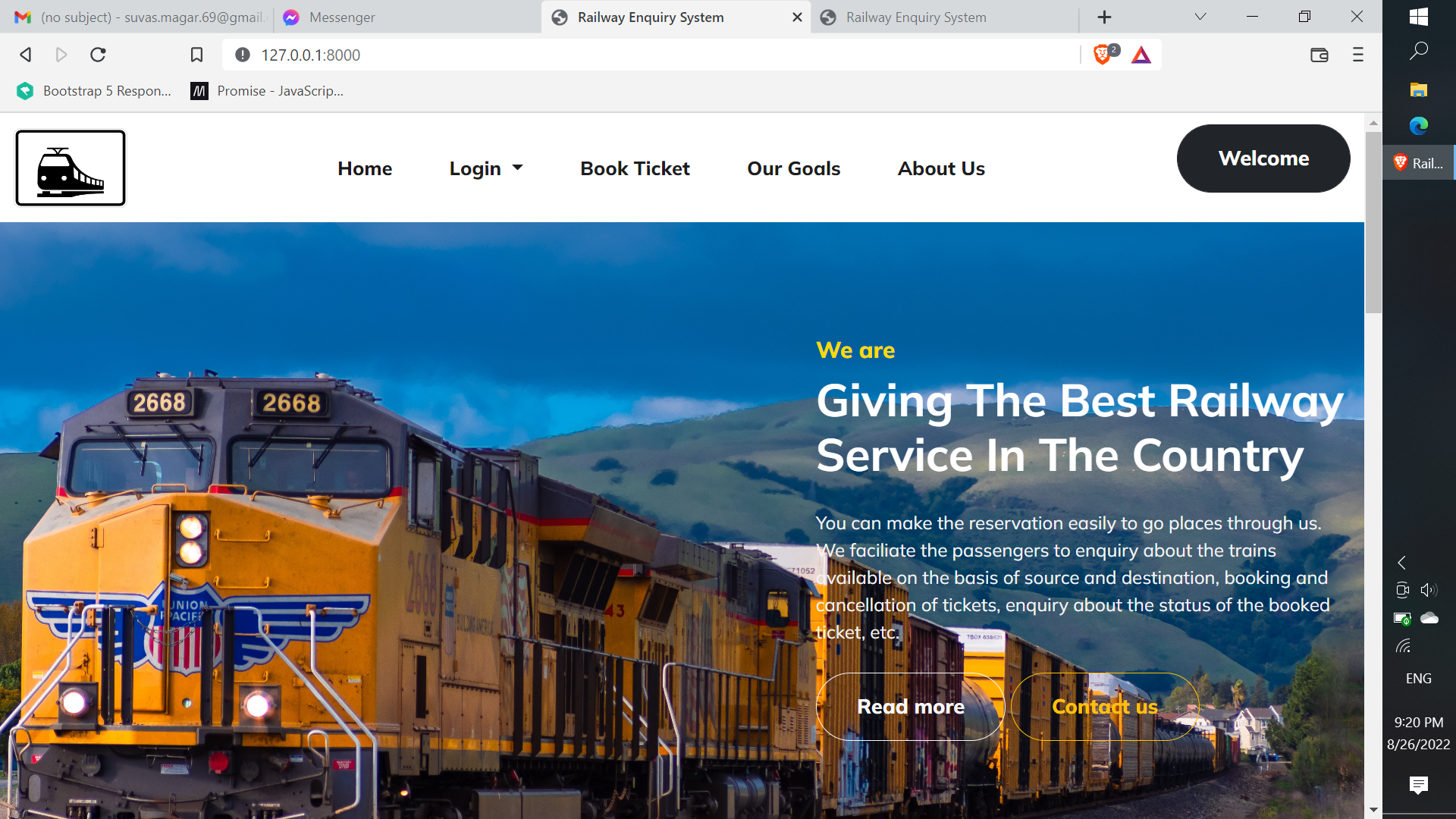
Title: Module View Template

Website Title: Modules in Django

URL: https://docs.djangoproject.com/en/4.1/

# APPENDICES

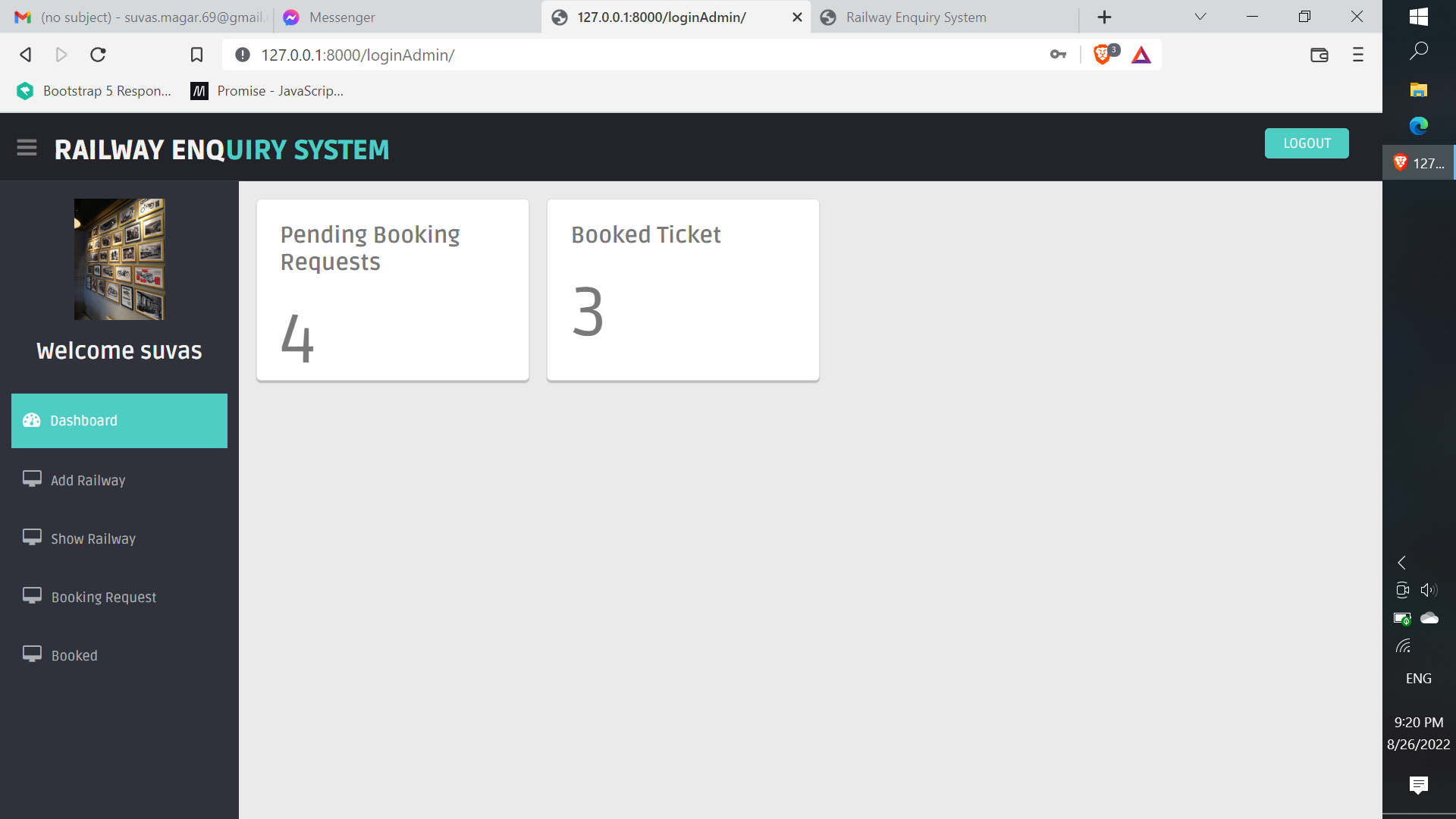
Appendix 1: Home page



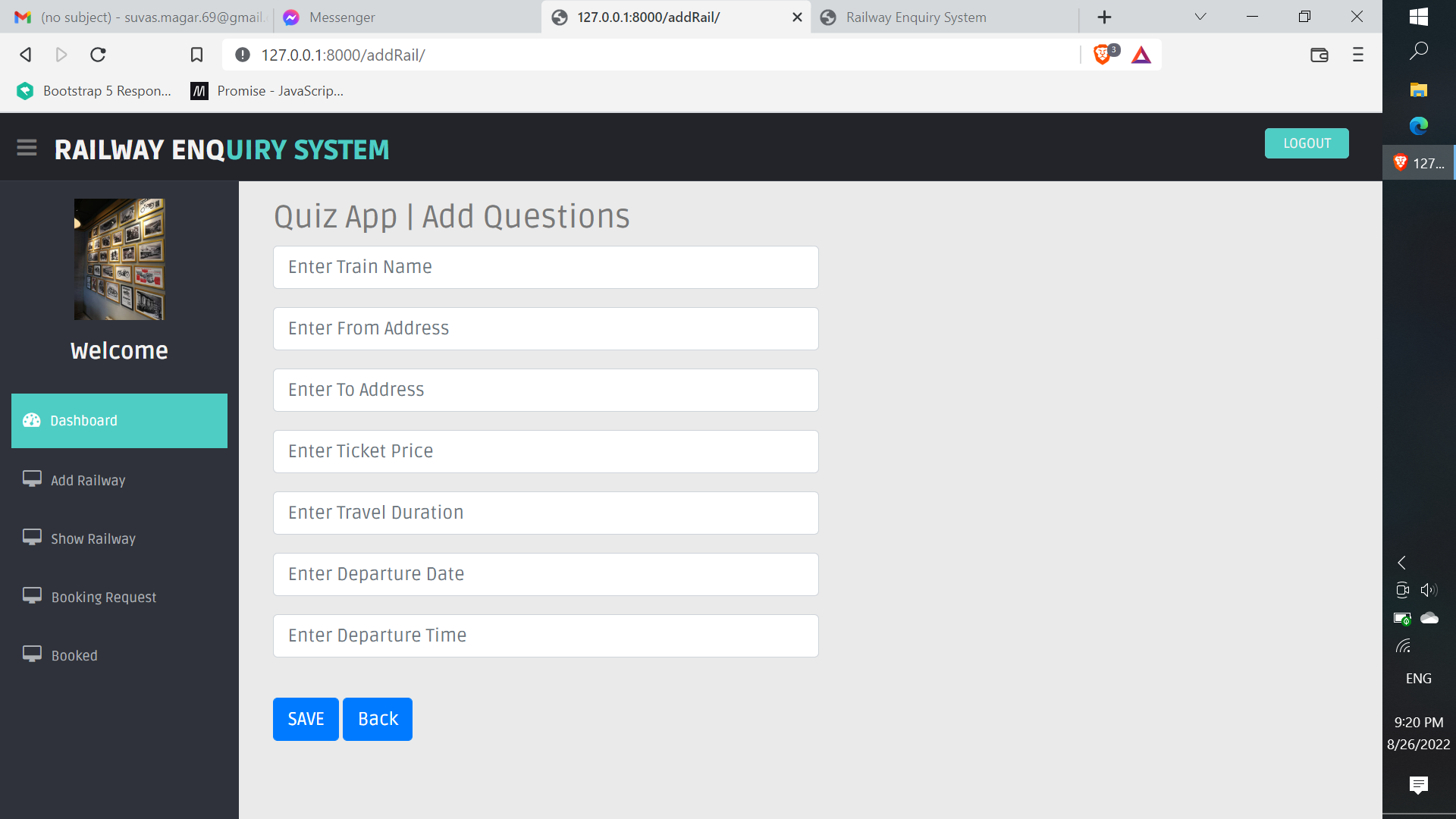
Appendix 2: Admin Login Page



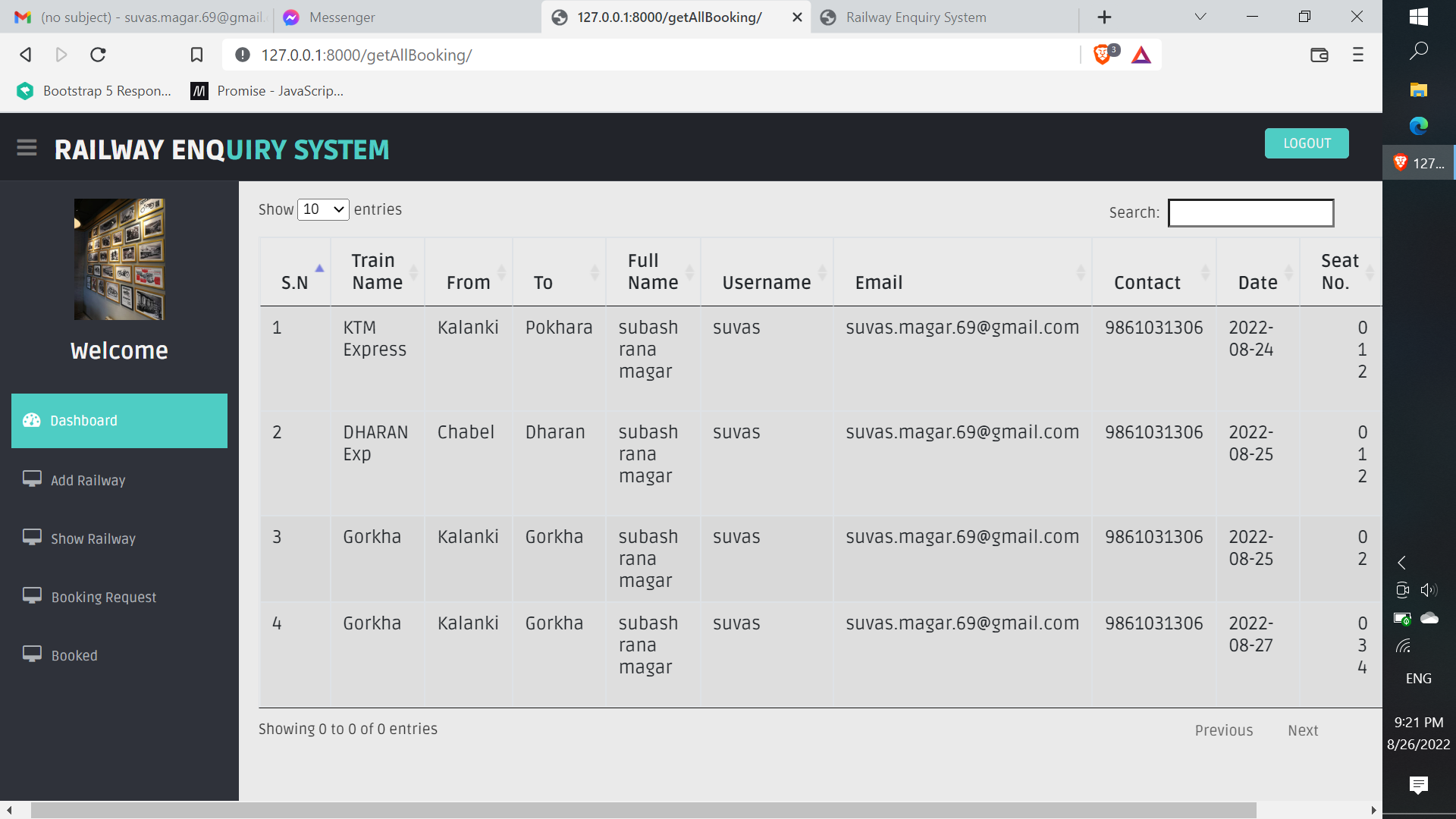
Appendix 3: Admin Dashboard



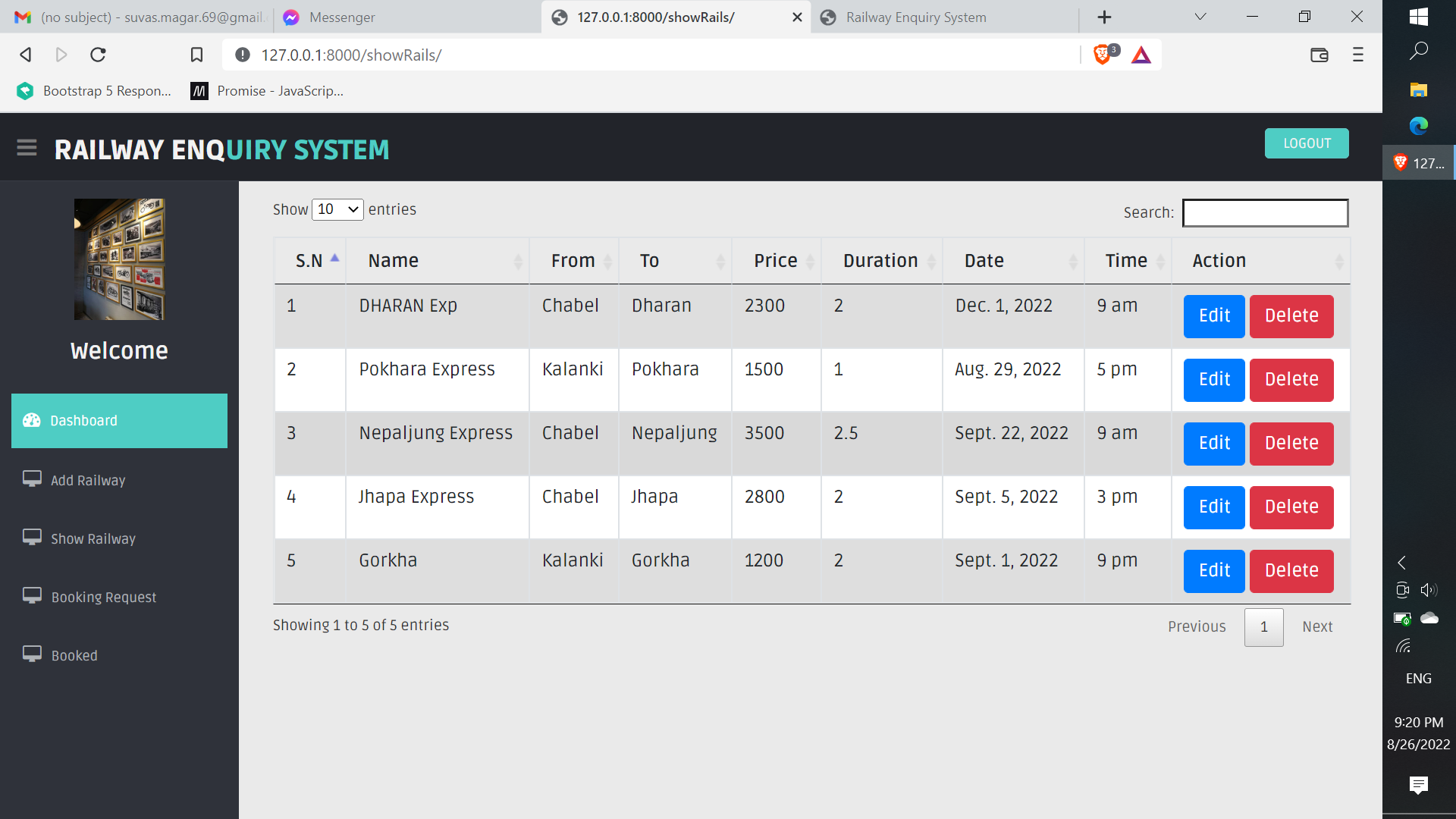
Appendix 4: Add Railway



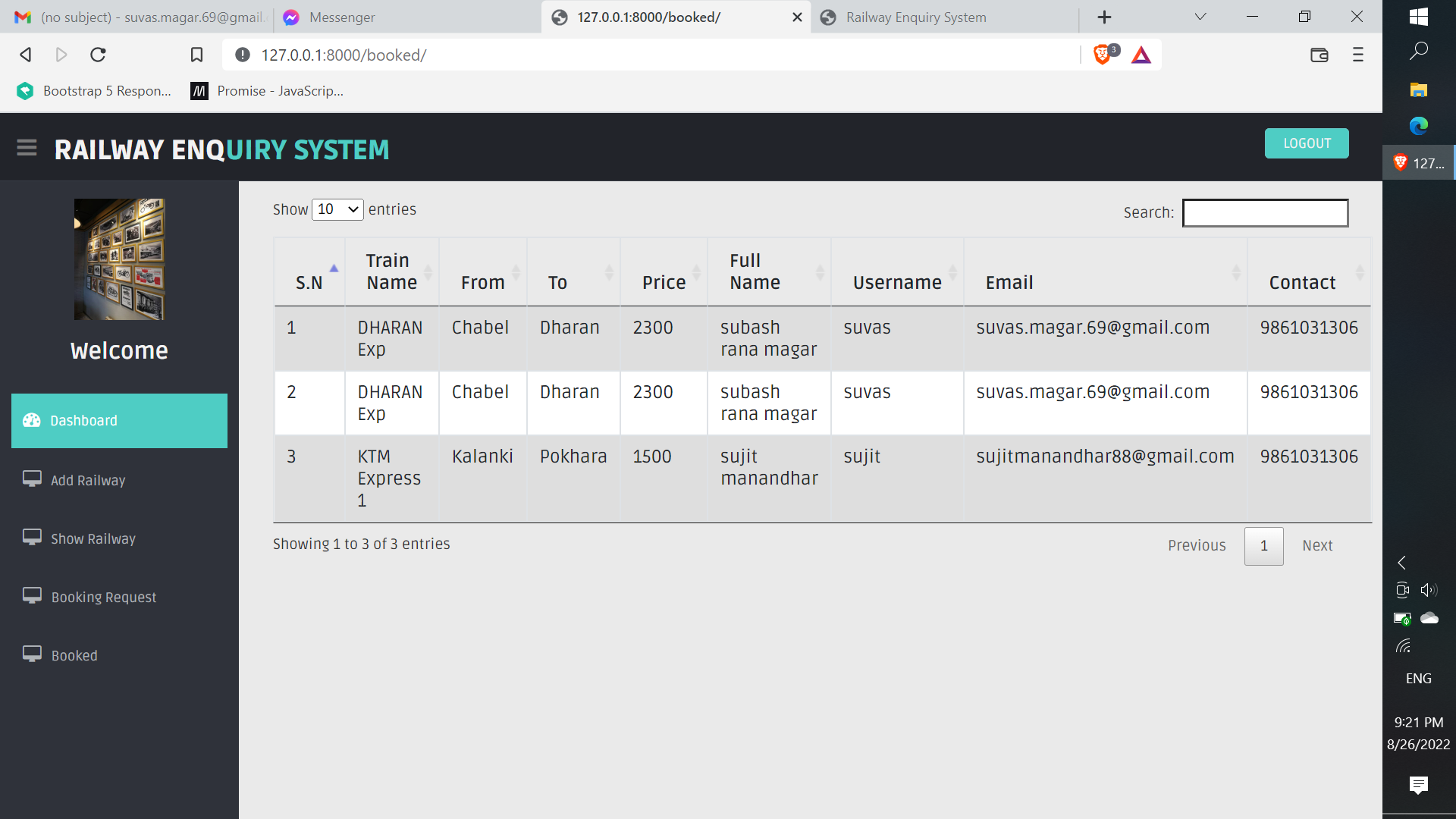
Appendix 5: Booking lists requested by the customers



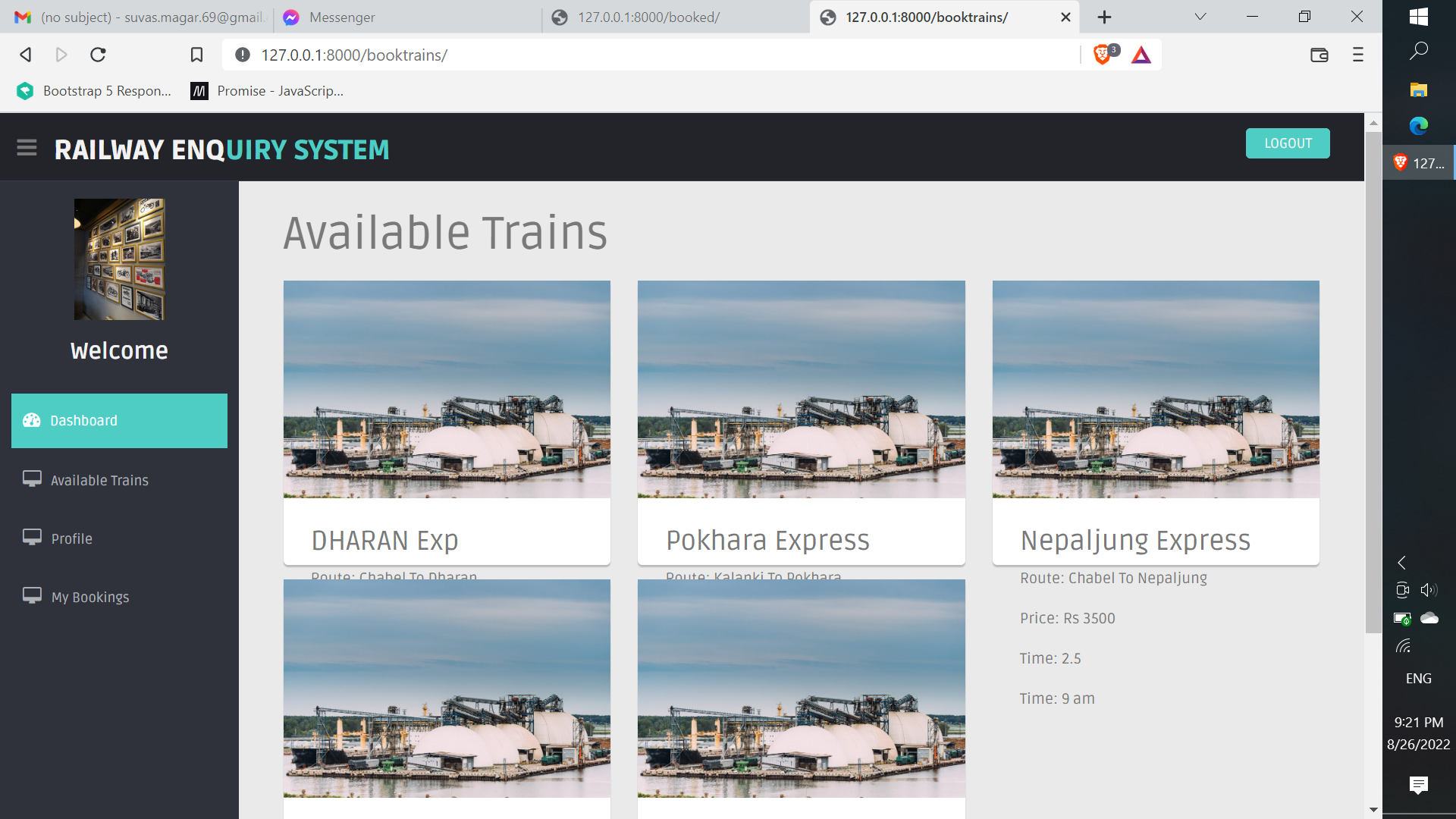
Appendix 6: Railways List in the Admin site



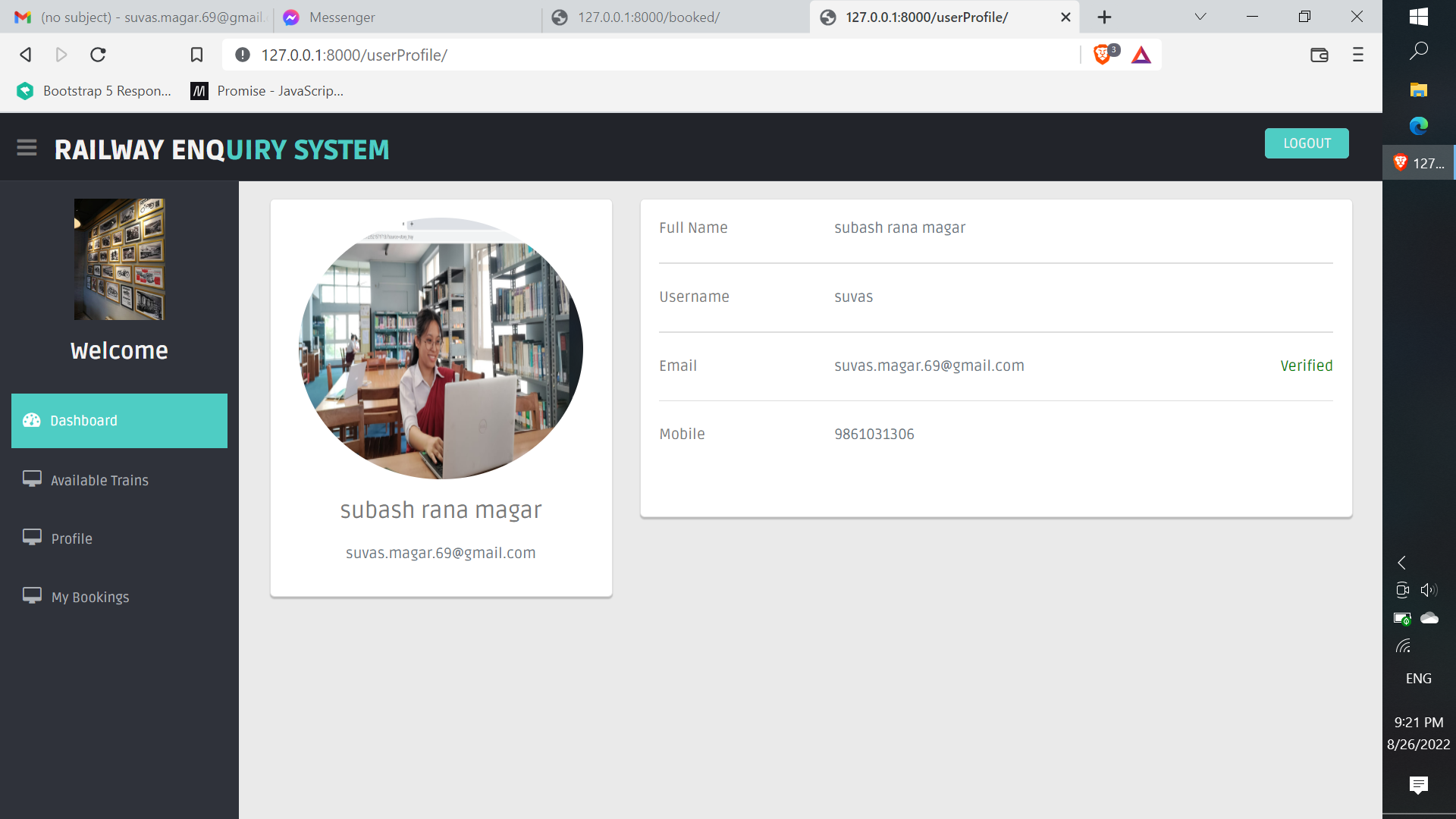
Appendix 7: Approved reservation for the seat requested by the Customers



Appendix 8: Customer Dashboard for the Lists of all the available trains



Appendix 9: Customer Profile Page



Appendix 10: Customer Requests for the seat reservations

