

|  |  |  |
| --- | --- | --- |
| |  | | --- | |  | | **PES University, Bengaluru** |
|  | (Established under Karnataka Act 16 of 2013) |
| **Department of Computer Science & Engineering** | |
| **Session: Jan - May 2022** | |

**UE19CS353 – Object Oriented Analysis and Design with Java**

**Theory ISA (Mini Project)**

Report on

**Seat reservation system**

**By:**

**Gourav Aravinda -----------------------------------PES2UG19CS130**

**Dev Darshan J ---------------------------------------PES2UG19CS108**

**Bhargav Narayanan P------------------------------PES2UG19CS088**

**6th Semester Section B**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Sl. no.** | **Chapters** | **pg. no.** |
| **1.** | **Project Description** | **3** |
| **2.** | **Analysis and design models**  2.1 Use case diagram  2.2 Class Diagram  2.3 User state diagram  2.4 Database chart diagram  2.5 Activity Diagram  2.6 Deployment diagram | **3-6** |
| **3.** | **Tools and Frameworks** | **7** |
| **4.** | **Design Principles and Design Patterns Applied**  4.1 MVC (Model View Controller)  4.2 Builder Pattern  4.3 Façade Pattern | **7-9** |
| **5.** | **Application Screenshots** | **9-12** |
| **6.** | **Team Contributions** | **12** |

**List of Figures**

|  |  |
| --- | --- |
| **chapter** | **Fig no** |
| **2.** | 2.1.1 Use case diagram  2.2.1 Class Diagram  2.3.1 User state diagram  2.4.1 Database chart diagram  2.5.1 Activity Diagram  2.6.1 Deployment diagram |
| **4.** | 4.1.1 MVC (Model View Controller)  4.2.1 Builder Pattern  4.3.1 Façade Pattern |
| **5.** | 5.1 Application Screenshot 1  5.2 Application Screenshot 2  5.3 Application Screenshot 3  5.4 Application Screenshot 4  5.5 Application Screenshot 5  5.6 Application Screenshot 6 |

1. **Project Description**

(GitHub-link: -

<https://github.com/GouravAravinda/Seat-Reservation-System>)

The main objective of our project is to provide an easy and user-friendly interface that enables our users, clients to reserve seats and purchase tickets to trains travelling from a certain station to the user’s desired destination.

The project insists on providing a hassle-free experience to our clients while they reserve seats in the trains they desire.

Java is one of the most used programming languages for developing dynamic web applications. A web application is computer software that utilizes the web browser and technologies to perform tasks over the internet. A web application is deployed on a web server.

[Java](https://www.javatpoint.com/java-tutorial) provides some technologies like [Servlet](https://www.javatpoint.com/servlet-tutorial) and [JSP](https://www.javatpoint.com/jsp-tutorial) that allow us to develop and deploy a web application on a server easily. It also provides some frameworks such as Spring, Spring Boot that simplify the work and provide an efficient way to develop a web application. They reduce the effort of the developer.

We will be using the above-mentioned technologies and methods to build a java web application to reserve seats and purchase tickets, described above.

1. **Analysis and Design Models**

2.1 Use Case Diagram: -

This diagram depicts the interaction of user and the database administrator with the system.



Fig 2.1.1

2.2 Class Diagram: -

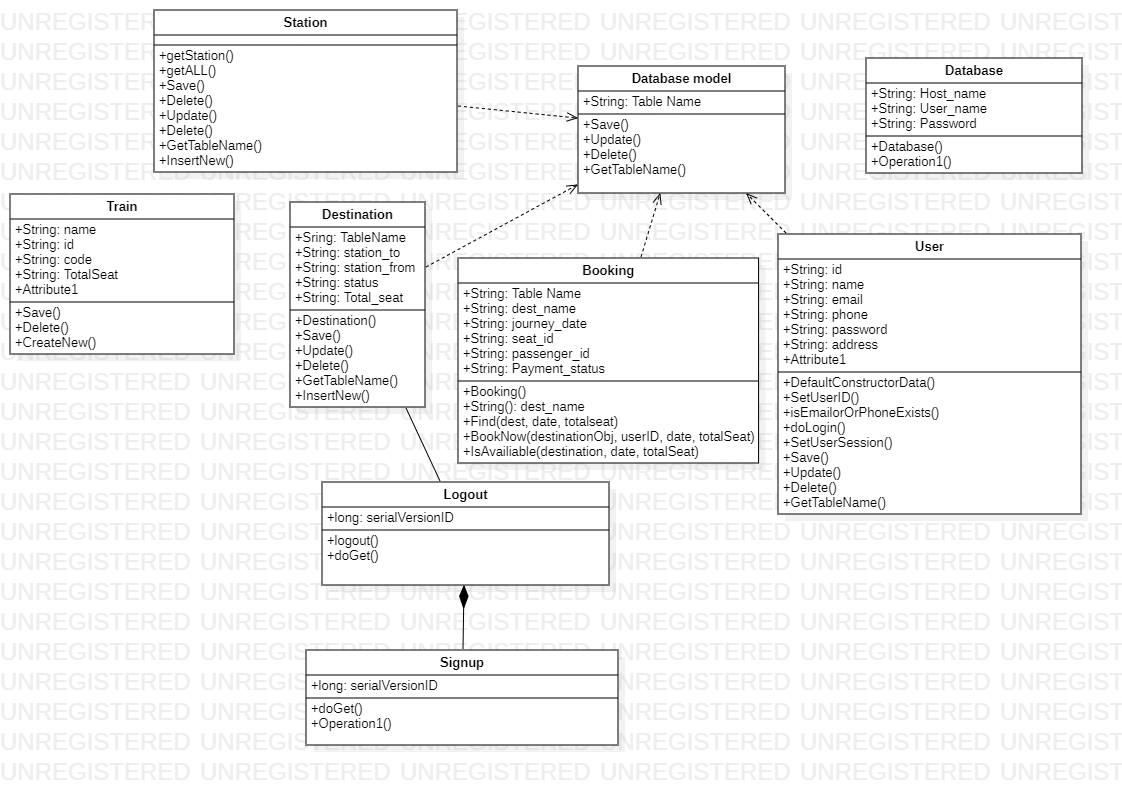


Fig 2.2.1

2.3 User State Chart Diagram: -



Fig 2.3.1

2.4 Database State Chart Diagram: -

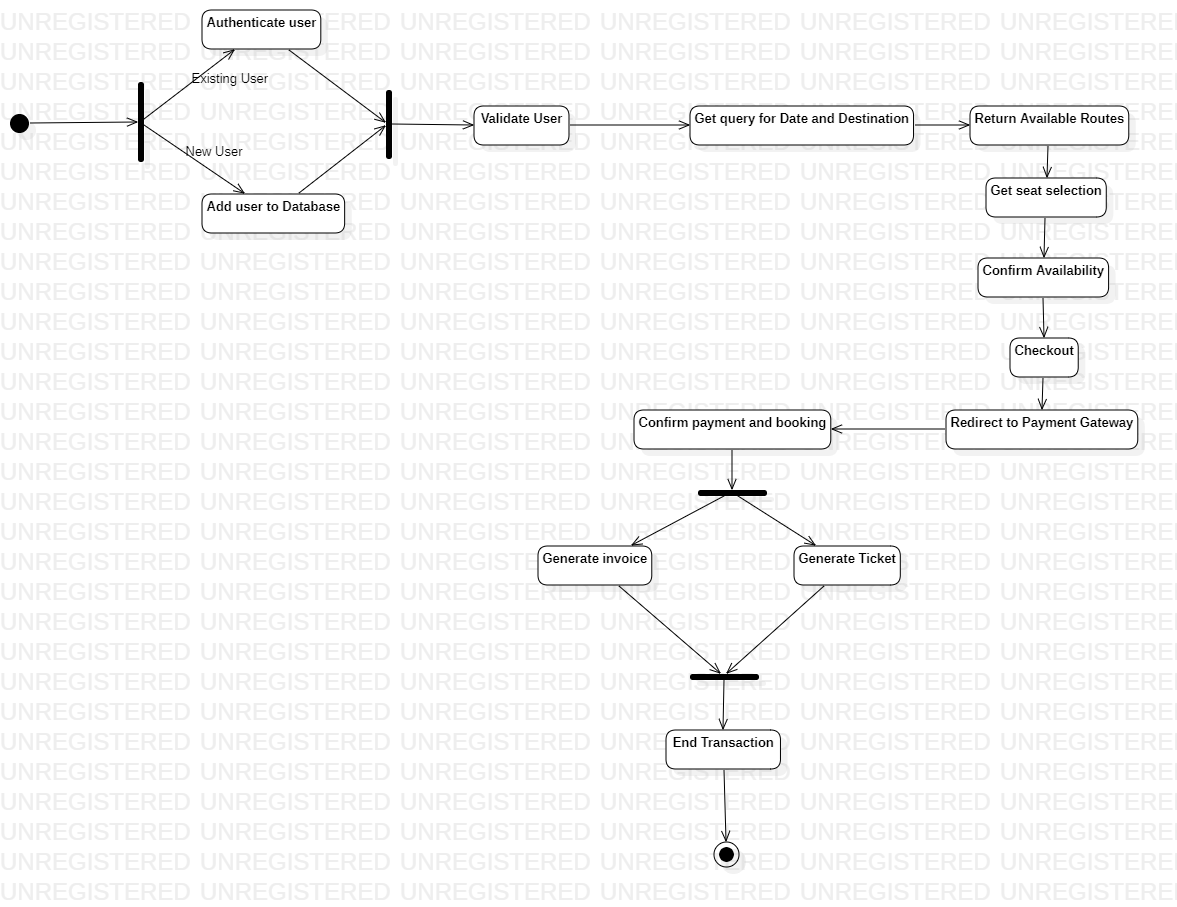


Fig 2.4.1

2.5 Activity Diagram: -

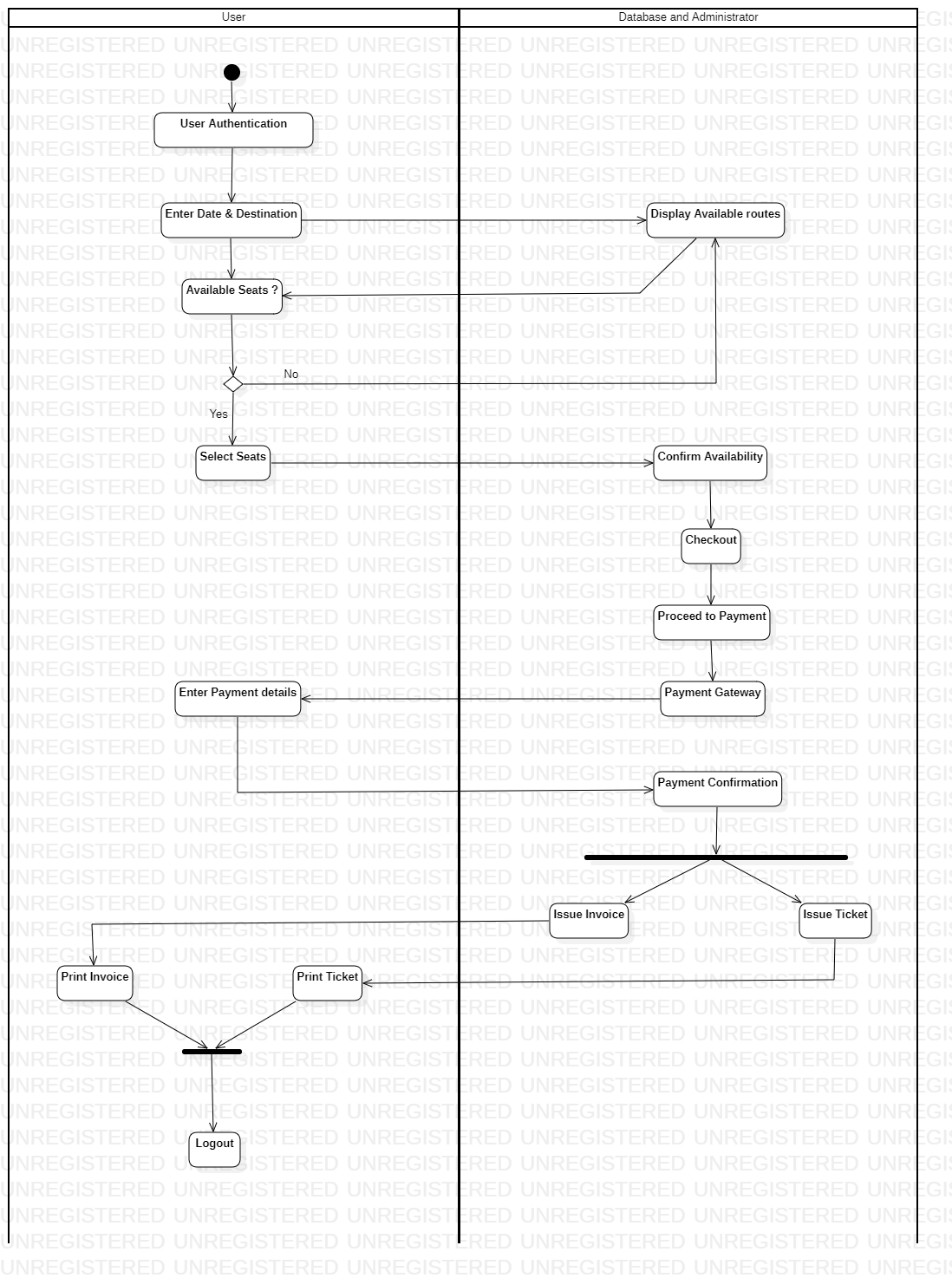


Fig 2.5.1

2.6 Deployment diagram: -

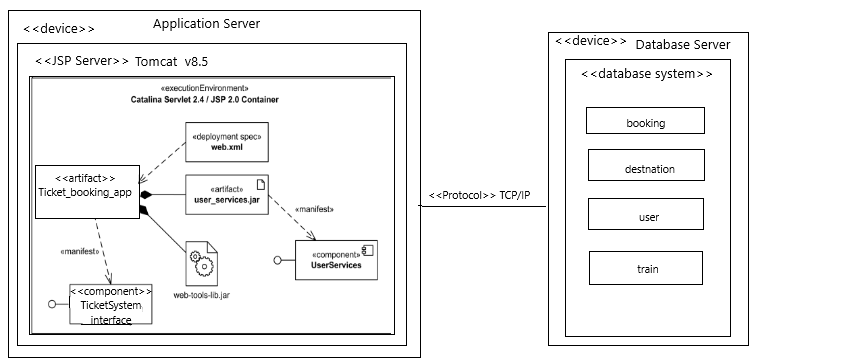


Fig 2.6.1

1. **Tools and Frameworks Used**

3.1 Tools used: -

* Eclipse IDE for J2EE projects
* Apache Tomcat Server v8.5
* MySQL Workbench

3.2 Frameworks: -

* JSP Servlet
* JRE
* J2EE services
* JDBC driver
* SQL for Databases

1. **Design Principles and Design Patterns Applied**

4.1. MVC :- (Model View Controller)

Generally, all implementations are based on MVC architectural pattern.

* + Model - represents the data & Business logic layers
  + View - represents the UI, every UI interaction is sent to the controller
  + Controller - represents the connection between the View & Model

The main purpose is to separate the functioning between the UI implementation and the application logic.

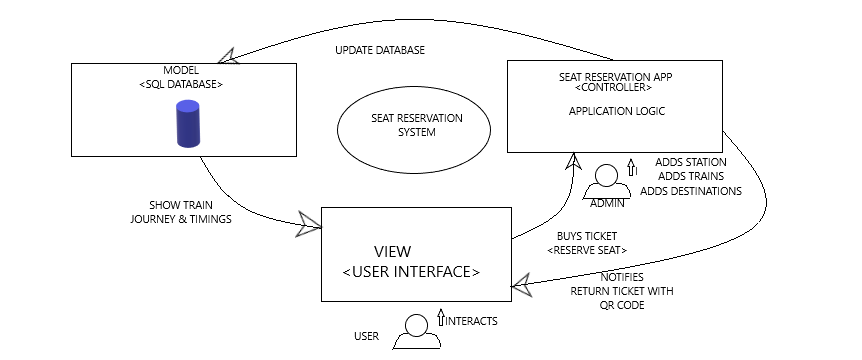


Fig 4.1.1

4.2. Builder Pattern

The Builder pattern, which is one of the 23 Gang of Four Design (GoF) patterns, is a creational design pattern that lets you construct complex objects step by step. It allows you to produce different types and representations of a product using the same construction code.

This Pattern is used in our Project to help generate tickets to different train journeys for the users. Every ticket generated will compose of the same constituents but for each ticket information varies.

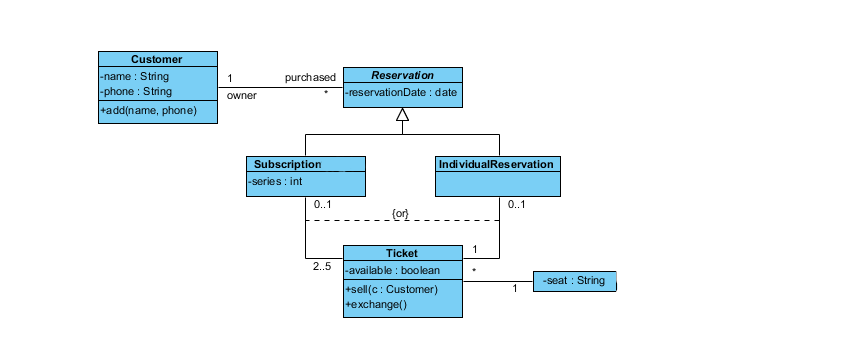


Fig 4.2.1

4.3. Facade

Facade is a part of Gang of Four design pattern and it is categorized under Structural design patterns. In Java, the interface JDBC can be called a facade because, we as users or clients create connection using the “java.sql.Connection” interface, the implementation of which we are not concerned about. The implementation is left to the vendor of driver.

A facade is an object that serves as a front-facing interface masking more complex underlying or structural code.

In our project we have used this design pattern to project to the users, the information about the train routes, without having to disclose the internal complexities within the code to the user.

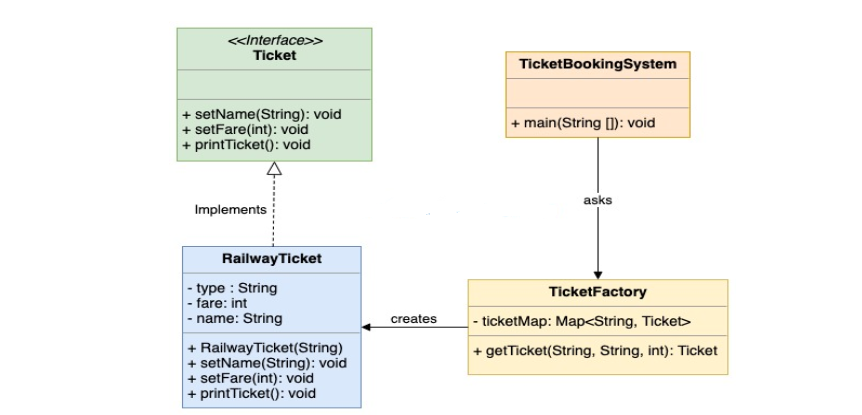


Fig 4.3.1

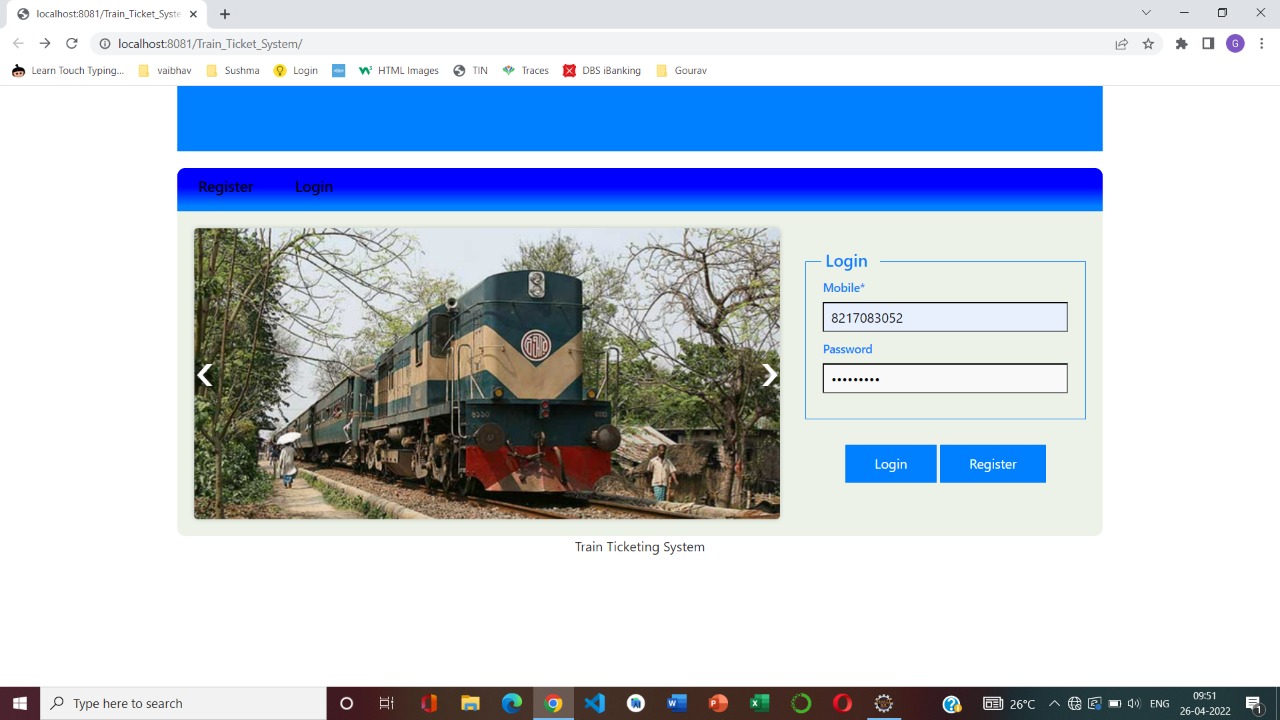
1. **Application Screenshots**

Fig 5.1

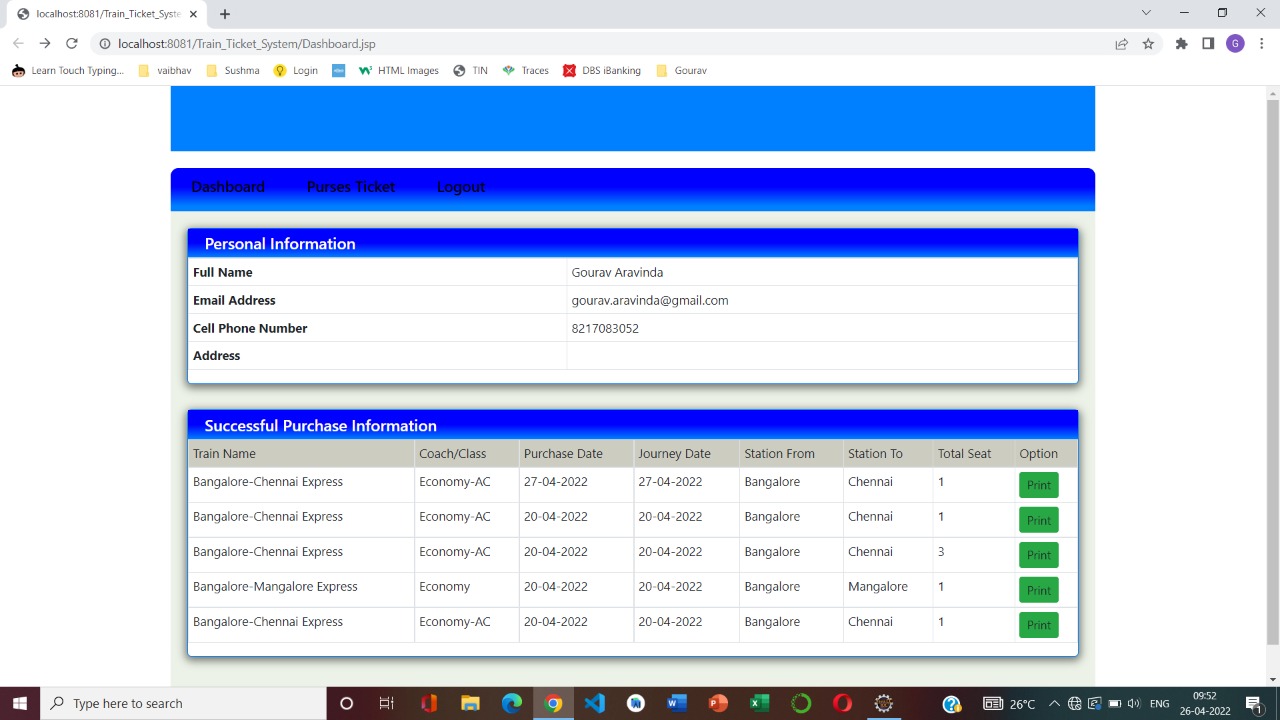


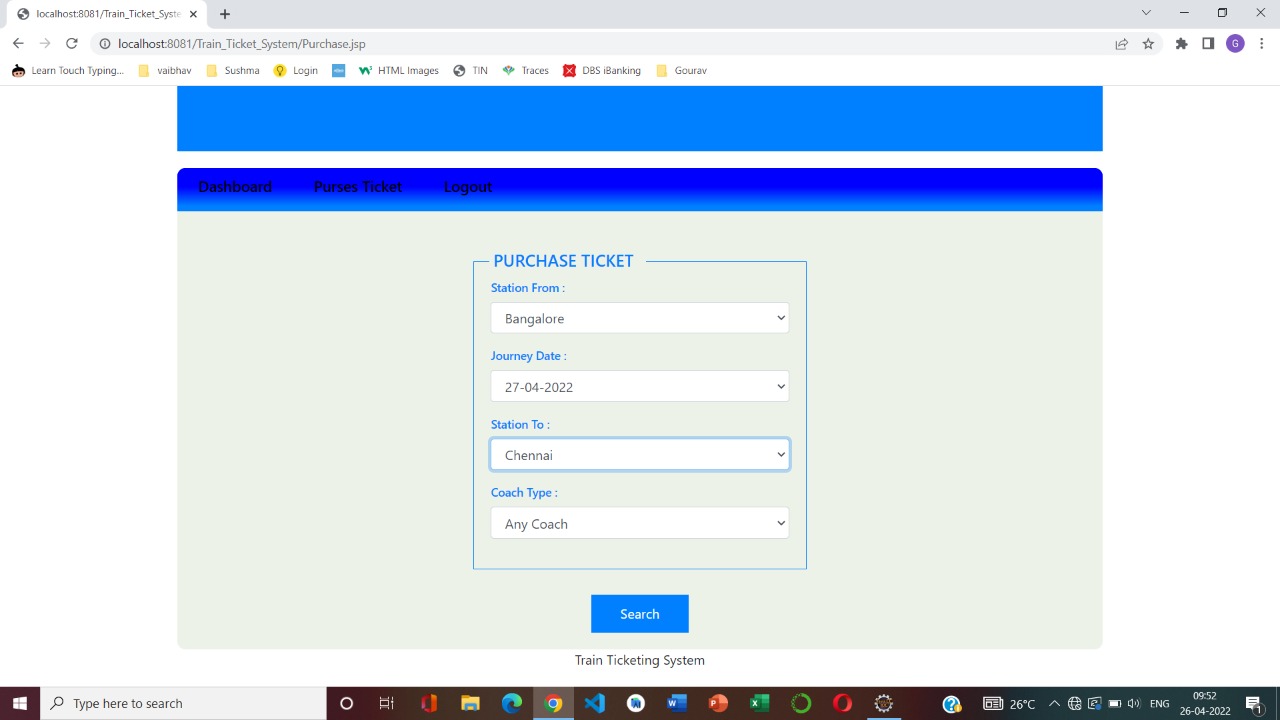
Fig 5.2

Fig 5.3

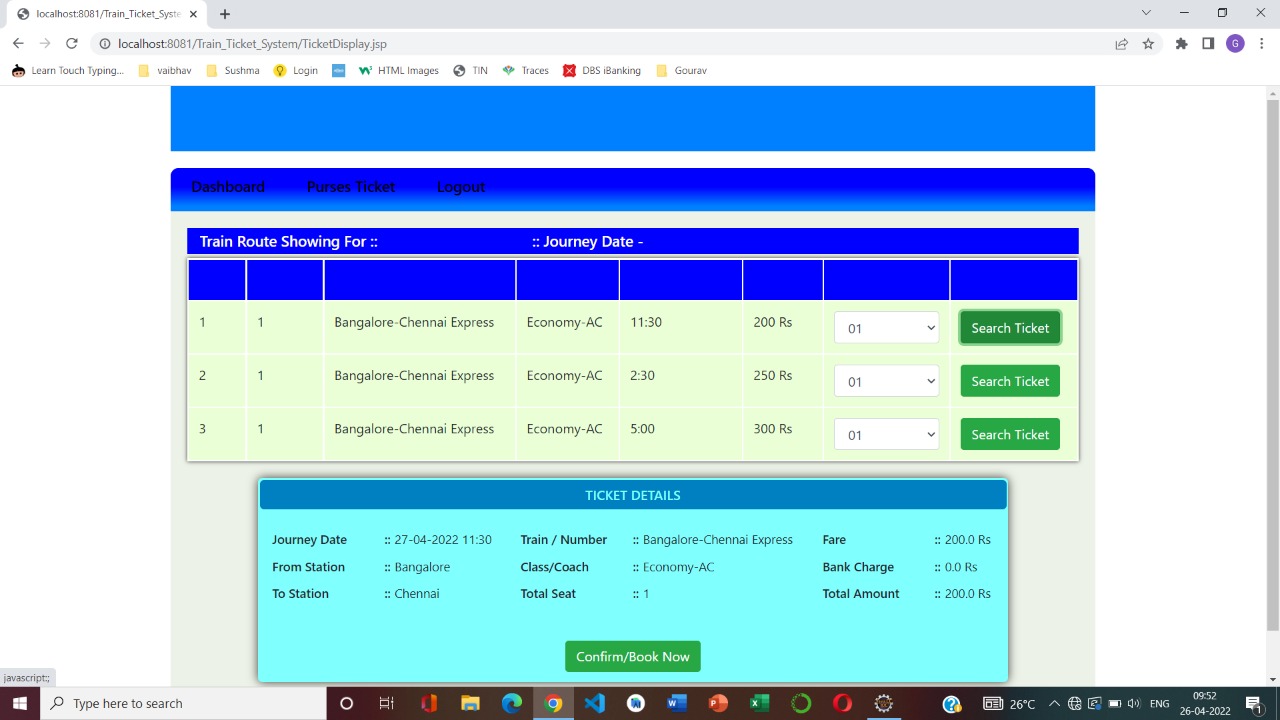


Fig 5.4

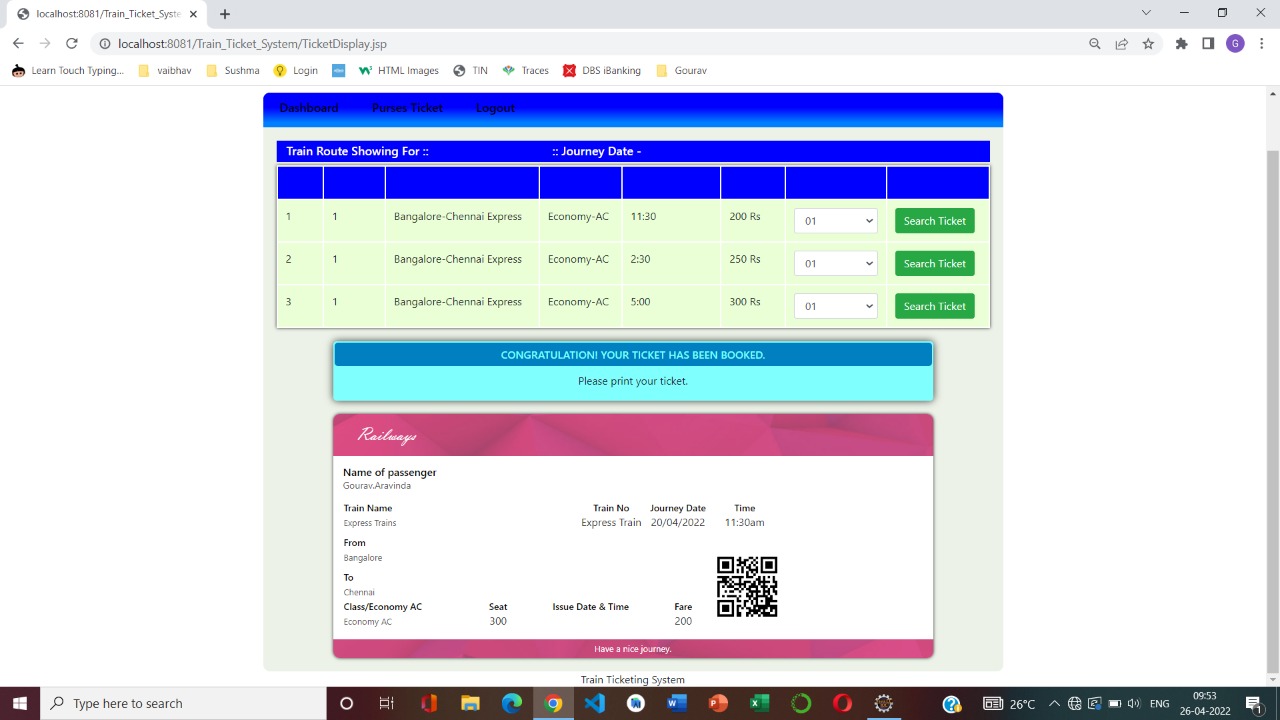


Fig 5.5

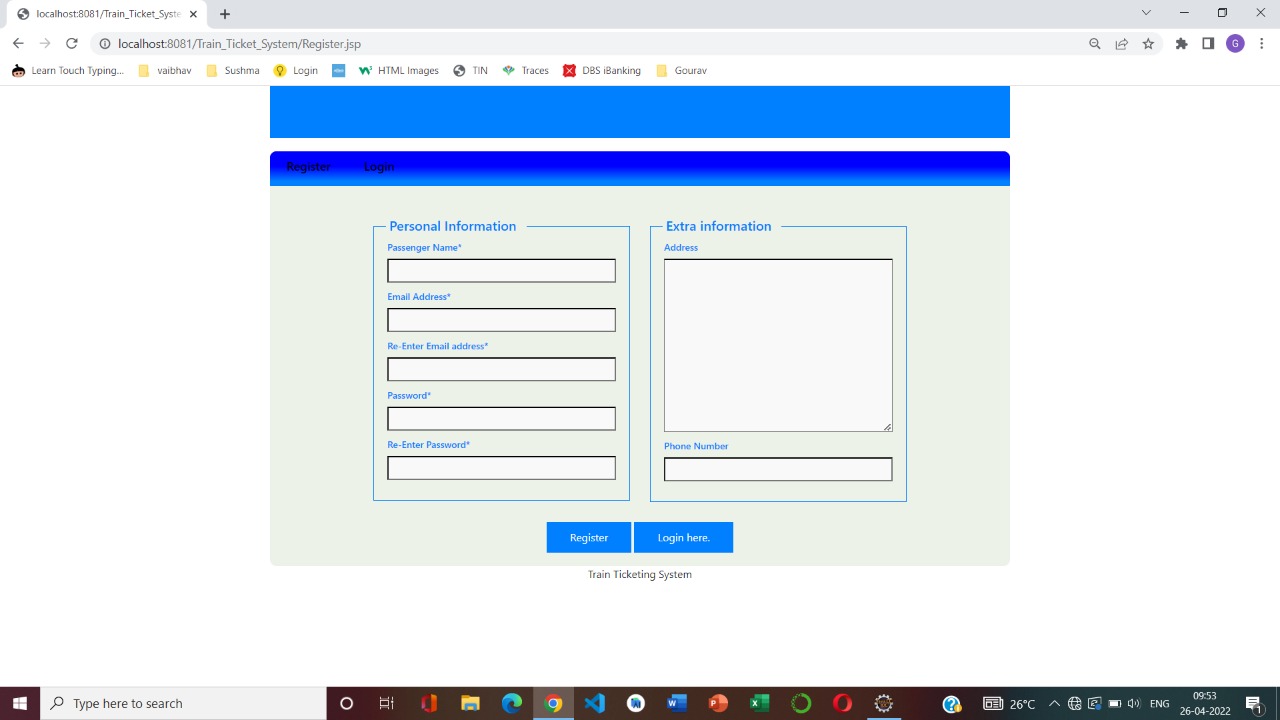


Fig 5.6

1. **Team member contributions**

|  |  |  |  |
| --- | --- | --- | --- |
| **SRN** | **Name** | **Contributions** | **Percentage** |
| **PES2UG19CS108** | **DEV DARSHAN J** | Train list, New destination, Create Stations, Station list | **33.334%** |
| **PES2UG19CS130** | **GOURAV ARAVINDA** | Seat reservation, Purchase ticket, Display ticket, Find ticket | **33.333%** |
| **PES2UG19CS088** | **BHARGHAV NARAYANAN** | User Registration, login , Sign up, User verification | **33.333%** |