DBMS PROJECT
Batch- B3
Group No: - 3

Team Members

1) **Surya Kartikeya**
   Roll No: - 210001041
2) **Shreyas Mahesh Honrao**
   Roll No: - 210001024
   GitHub Repo Link: -
   https://github.com/surya1176/TRAIN-DATABASE.git

TRAVEL DATABASE MANAGEMENT SYSTEM

What is Travel Database Management System?

A travel database management system helps to keep a record of booking, tracking, and analysing business travel. Typically, a travel management system refers to a platform, where travellers can book tickets and travel across various places.

So, our Website (**Happy Easy Go**) showcases Train Ticket Booking System. Basically, the process of booking the ticket on our website is easy and simple.

For Building the website, we have used several Tech Stacks such as: -

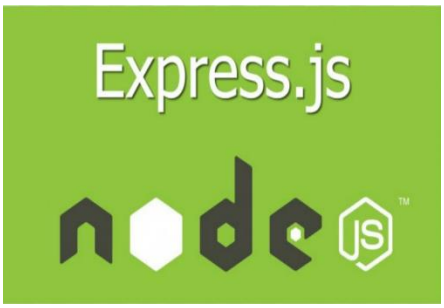
1) **HTML &CSS**

With CSS, you can control the color, font, the size of text, the spacing between elements, how elements are positioned and laid out, what background images or background colors are to be used, different displays for different devices and screen sizes, and much more!

2) **JAVASCRIPT**

Javascript is used by programmers across the world **to create dynamic and interactive web content like applications and browsers**. JavaScript is so popular that it's the most used programming language in the world, used as a client-side programming language by 97.0% of all websites

3) NODE JS & EXPRESS JS



Express. js is a framework based on Node. js which is used for building web-application using approaches and principles of Node. js that provides a robust set of features for web and mobile applications.

4) MYSQL

MySQL is used as a relational database management system (RDBMS). It is free and open-source. Moreover, it is also useful for both small and large applications, ours being a small one currently.

5) EJS

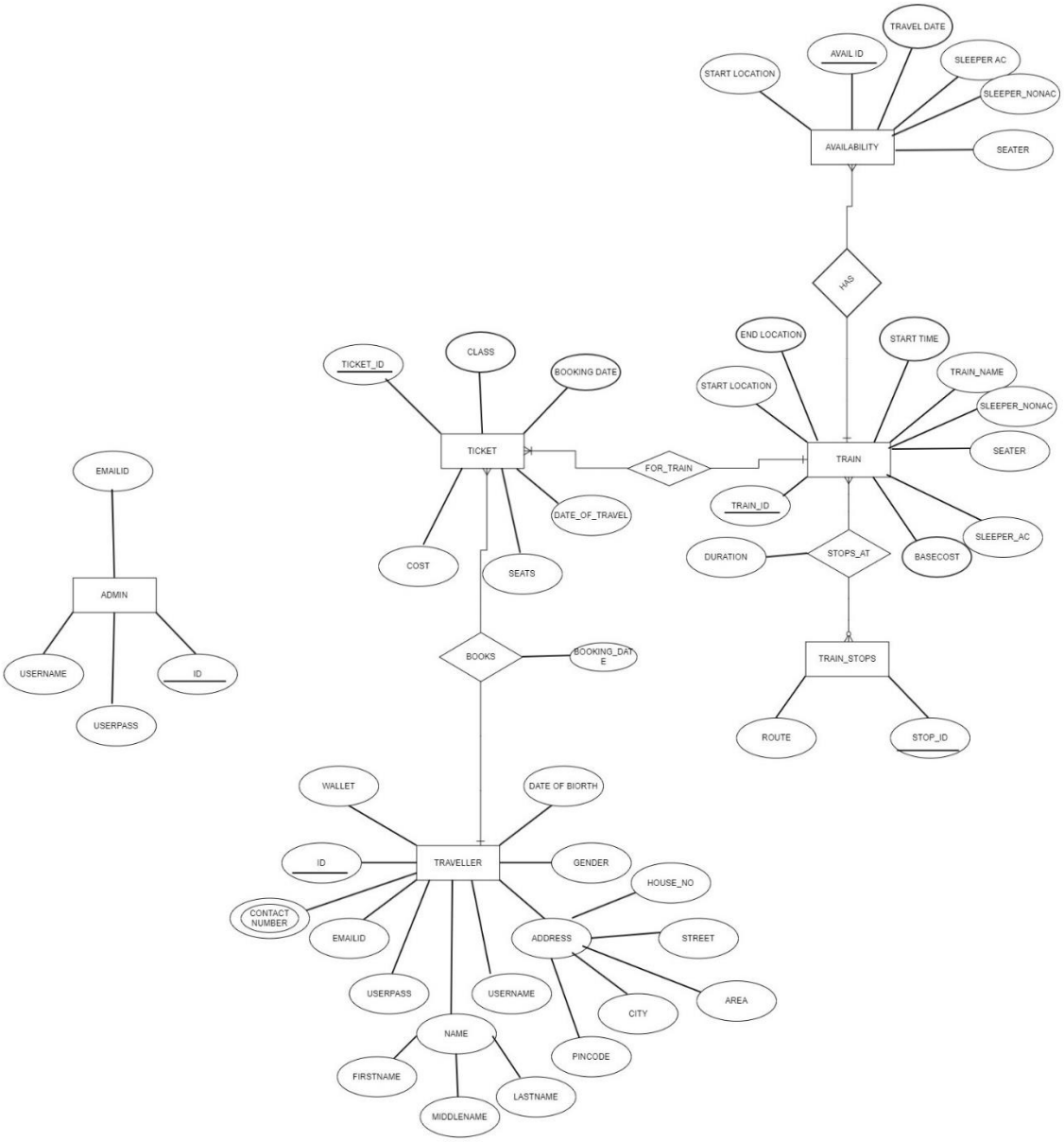
EJS (Embedded JavaScript Templating) is one of the most popular template engines for JavaScript. As the name suggests, it lets us embed JavaScript code in a template language that is then used to generate HTML. It retains the syntax of HTML while allowing data interpolation.

Why These Tech Stacks only?
. HTML, CSS & JAVASCRIPT for the front end. We also made sure that our website is responsive.
. MYSQL is the database we used.
. EXPRESS JS and NODE JS are used to connect the front end with the back end.
. Since in the modern-day world most of the websites are made using java script and it is quite easy to use, we have used these tech stacks for our project.

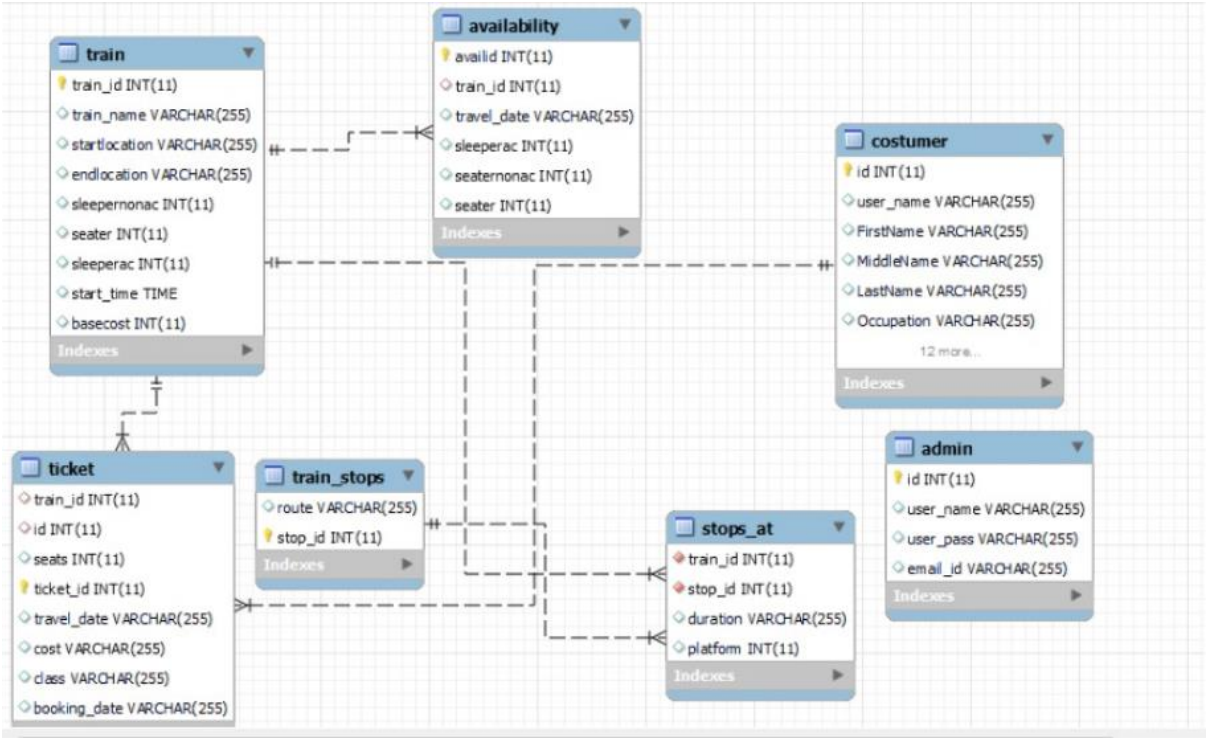
FEATURES AND DESCRIPTION OF OUR ER-DIAGRAM

Few of the features we implemented in our project were: -
. Before logging into the website, the customers must register and fill all their details.
. First the customers will have to login to the website to book ticket, know their ticket status, cancellation of the ticket etc.
. He will have to enter the date of travel, arrival and departure location.
. He will then be showed with different train options which are available.
. He will then enter all his personal details and then he will be confirmed with the ticket.
. Depending on the date he booked his ticket he will get a discount on the ticket price.
. After he books the ticket, he can view his ticket status and further can also cancel his ticket.
. The amount of the refund he will be getting depends on how early or late he cancels the ticket.
. He can also view the information about the trains by entering the train id.
. In the case of emergency if a train gets cancelled, we will be informing all the customers who have booked that train through mail so that they can book another train.

ER MODEL



Relational Schema



Firstly, we have implemented the costumer table which will contain all the details about the customer. Every customer is associated with a unique id. While registering in our website the customer will have to enter his email-id so that we can inform him about the travel updates. The customer will also choose a username while registering so that he can login to the website using these credentials later. All the details he fills in the register page will be stored in our database. We have ensured that username is unique, and no two customers can have the same username. The table will also contain details like date of birth, gender and other details which he will be filling before confirming the ticket. Since we could not implement payment, we have included a wallet attribute in the customer table so that we can keep updating the attribute after each transaction. The primary key of the customer table is referenced in the ticket table.

After a customer books a ticket, we will allot him/her a ticket id which is also the primary key of the ticket table. The table will be containing information about the class which he chooses, cost of the ticket, number of seats booked etc. All these will be displayed on the ticket after confirmation. These details will also display during cancellation. Ticket table has attributes train id, which is referenced from the train table and id which is referenced from the customer table.

We have then implemented the train table which will have information about various trains. Each train is given a train id which is the primary key of the train

table. Each train will have attributes start location, end location, start time and the base cost which we will be the minimum cost of the ticket in that train. It is to be noted that the final cost of a ticket is dependent on how fast a customer books a ticket and class which the costumer books. Train id is referenced in the ticket table and availability table.
We have train stops table which will have a stop id which is also the primary key of the table. Stop id is assigned for a place which is given the name route in train stops table.
Connecting the train and the train stops table is the stops at table. This table has attributes train id which is referenced from the train table. We have stop id which is referenced from stops table. This table also contains other details like platform which the train will be arriving, the time taken to reach a station. This table is helpful to get information about the route which the train is following. When a user searches for train updates by entering a train id we will be displaying information from this table and train table to inform him about all the routes which the train takes.
To keep track of the availability of the trains and the number of seats in a train on a certain date we have implemented the availability table. It also has attributes sleeper ac, sleeper non ac, seater which indicates the number of seats in each class and its count will be updated after a user books a ticket. The primary key of this table is avail id. This table will have foreign key as train id. This table is mainly implemented as the same train can run on multiple dates and each train will have different availability of seats. It has attributes sleeper ac, sleeper non ac, seater which will give information about the number of seats left in each class of the train running on that date. When a user searches for trains he will be shown information about various trains based on his travel date. Availability along with train table display information about the trains available after a user search for trains.
Finally, we have the admin table which will help us keep track of the admins and the admin will the option of releasing the tickets. The admin can cancel a particular train if it fails suddenly on that day. After cancelling the train all the customers will be informed via mail about the train failure.

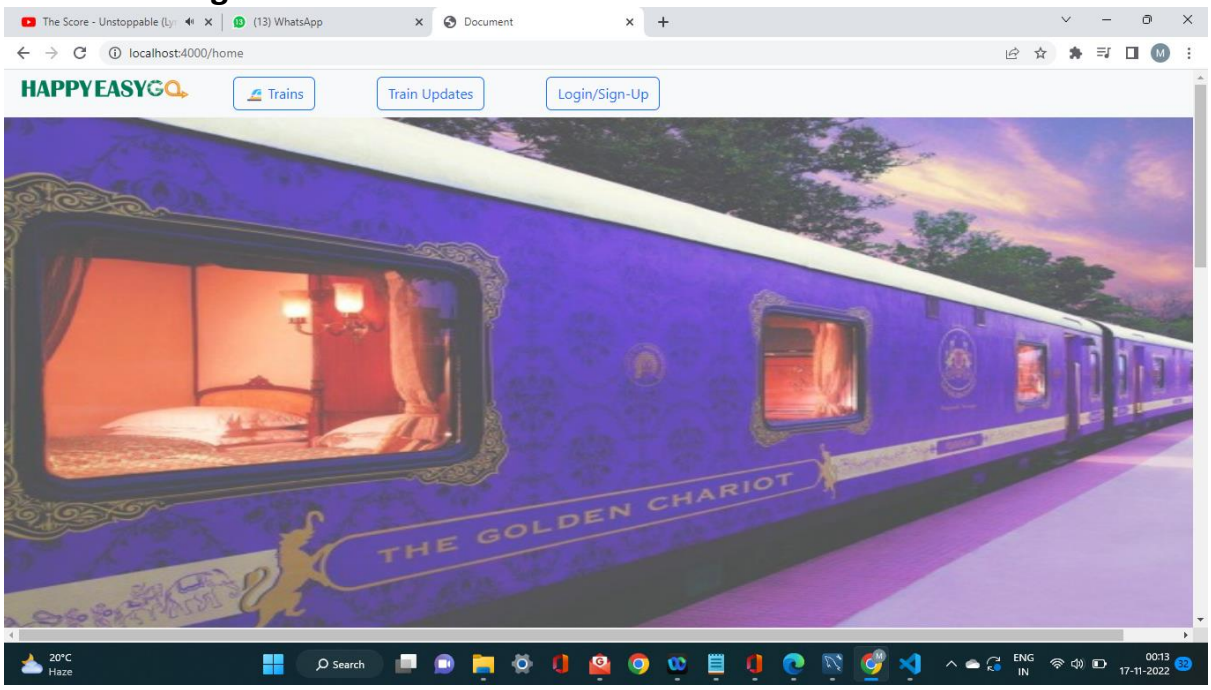
Relationships involved in our ER model: -

. The relationship between the costumer and the ticket table is one to many as one costumer can book multiple tickets and each ticket id is corresponding to only one customer.
. The relationship between the train and the ticket table is one to many as one train can correspond to multiple ticket ids and each ticket id corresponding to only one train.

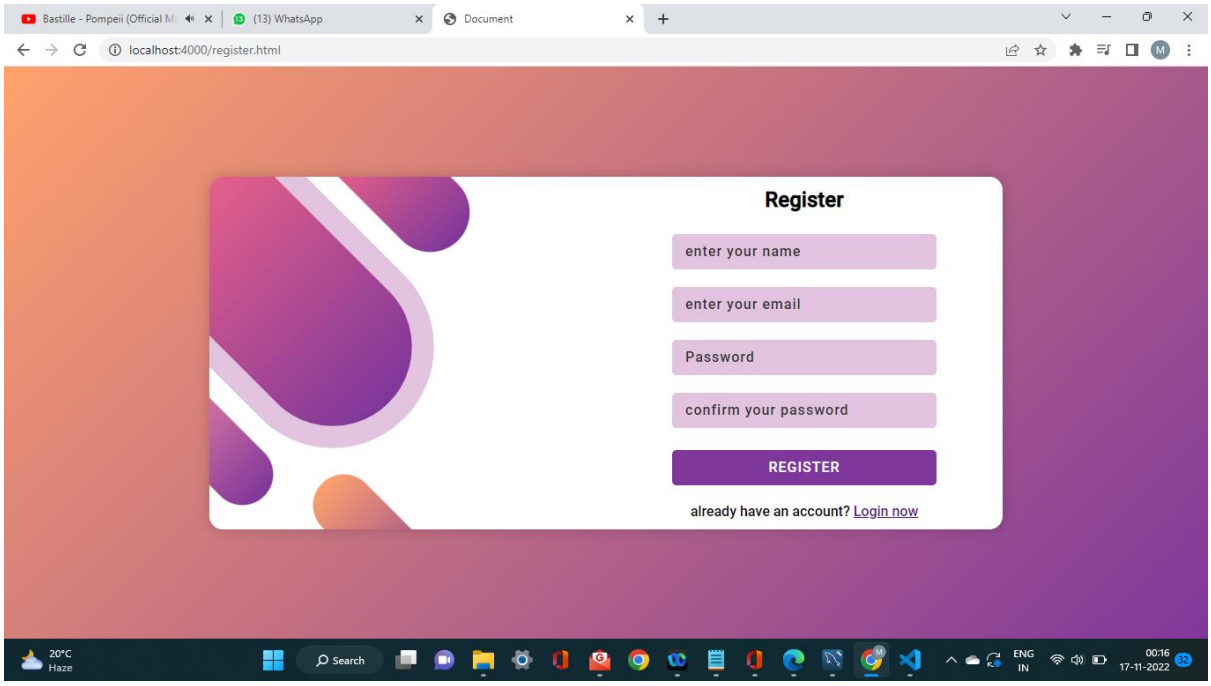
. The relationship between the train and availability table is one to many. This is because a train can correspond to multiple avail ids and each avail id corresponds to one train.
. The relationship between train and train stops table is many to many since one train Id can have many stops and each stop id can correspond to many train ids. Hence, we need 3 tables to represent this relationship which is done by stops at table.

PRESENTATION OF OUR WEBSITE

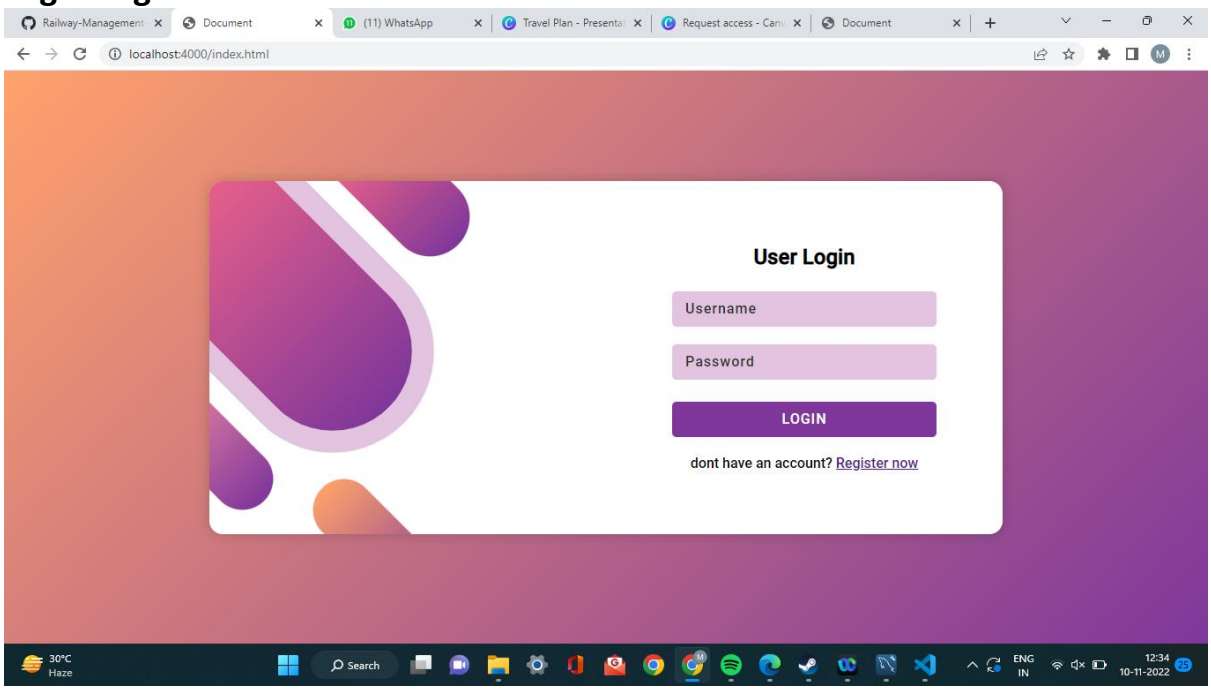
Our Home Page: -



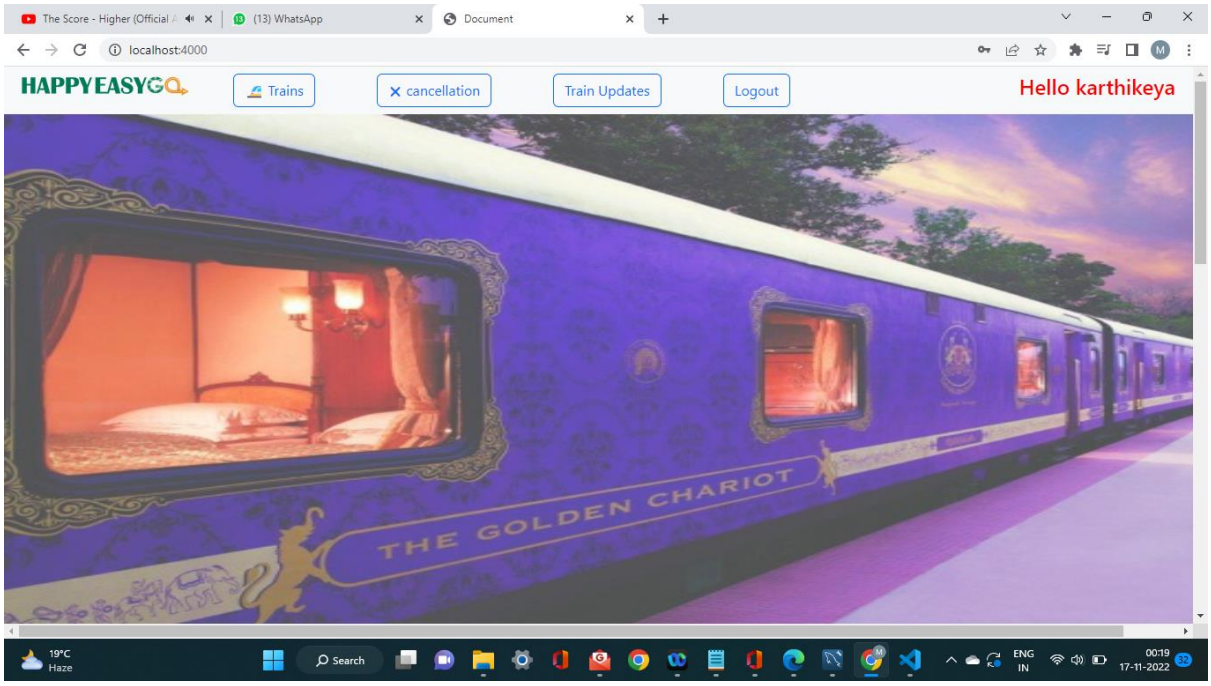
Register Page: -



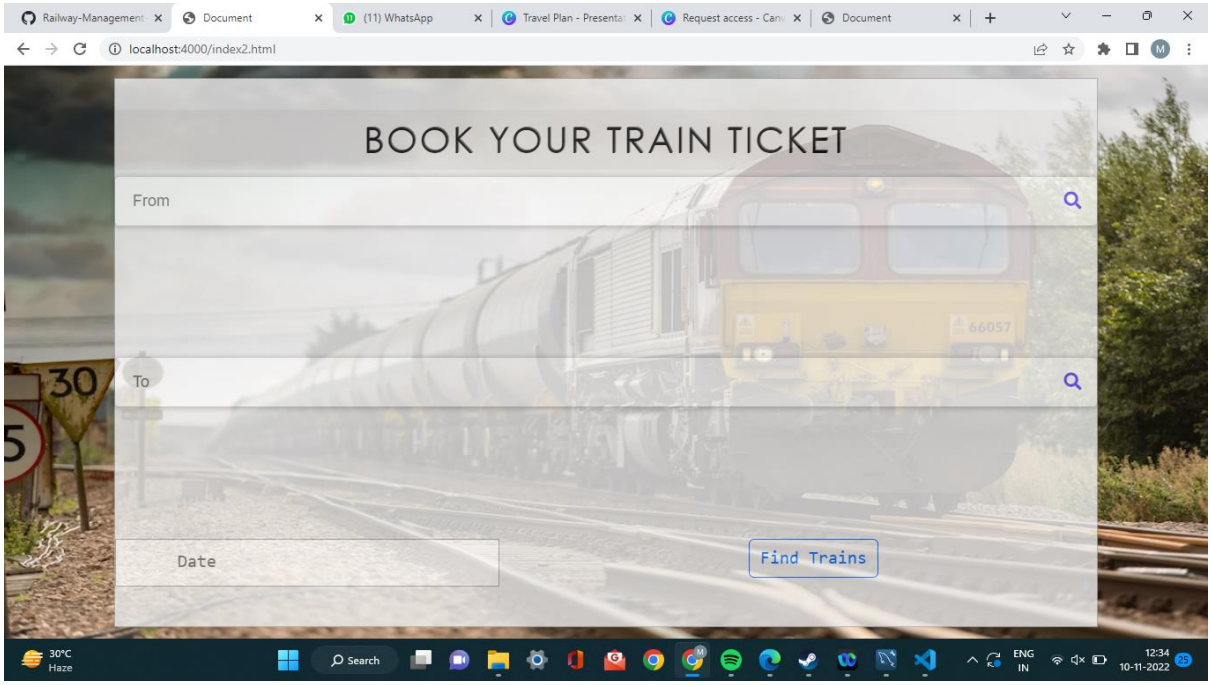
Login Page: -



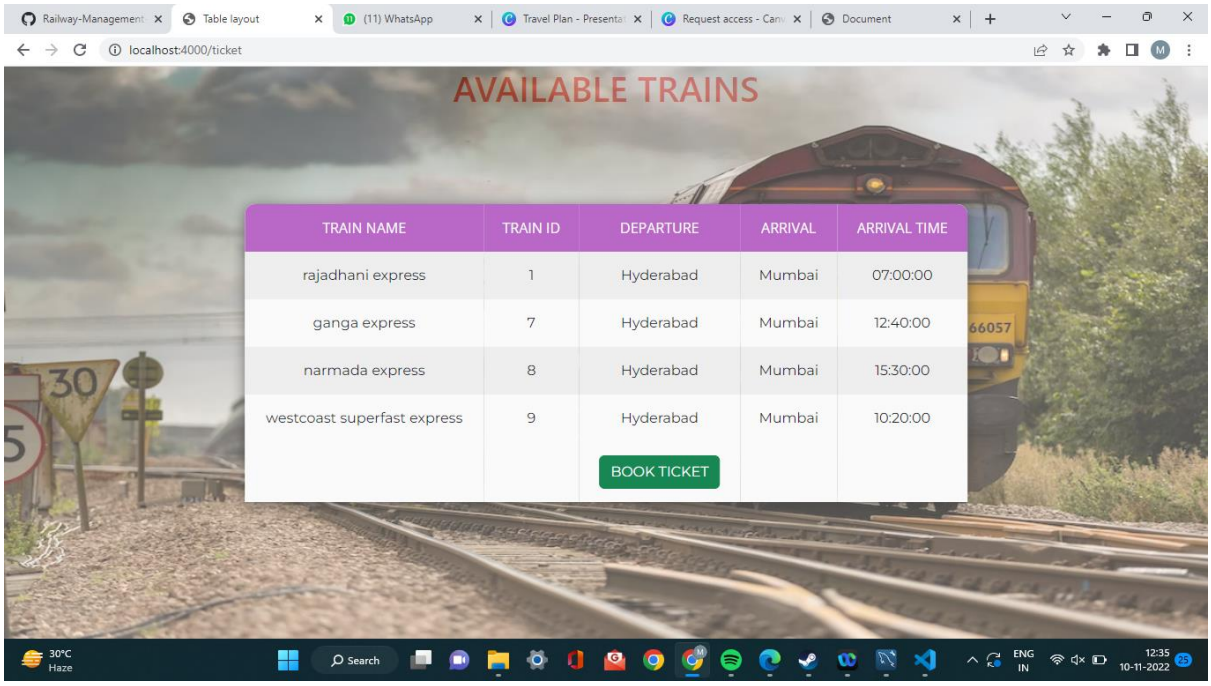
After a user login he will be redirected to the home page again.



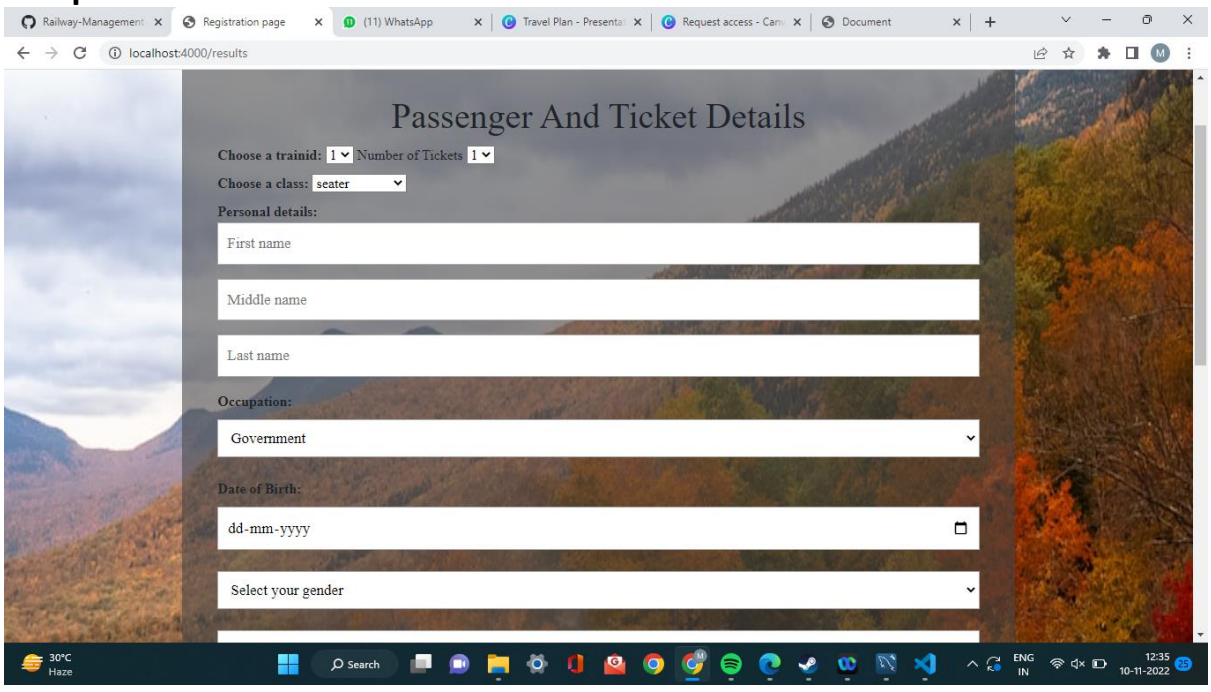
A customer can click on the trains button to find trains.



After entering all the fields train results will be shown.



After clicking on the book ticket button, the user will have to fill about ticket and personal details.



After a customer enters all the details, he will be provided with an option of downloading the ticket.

Download Home

The below ticket is shown when the costumer books within 30days of the travel date. (The ticket cost is increased by 50/- due to late booking)

Download Home

Depending on the number of seats booked the cost of the ticket is accordingly shown.

Download Home

Depending on the class selected the cost of the ticket varies.

Download Home

Since we couldn't implement payment, we have added an attribute wallet to the costumer table and update that attribute after every transaction.

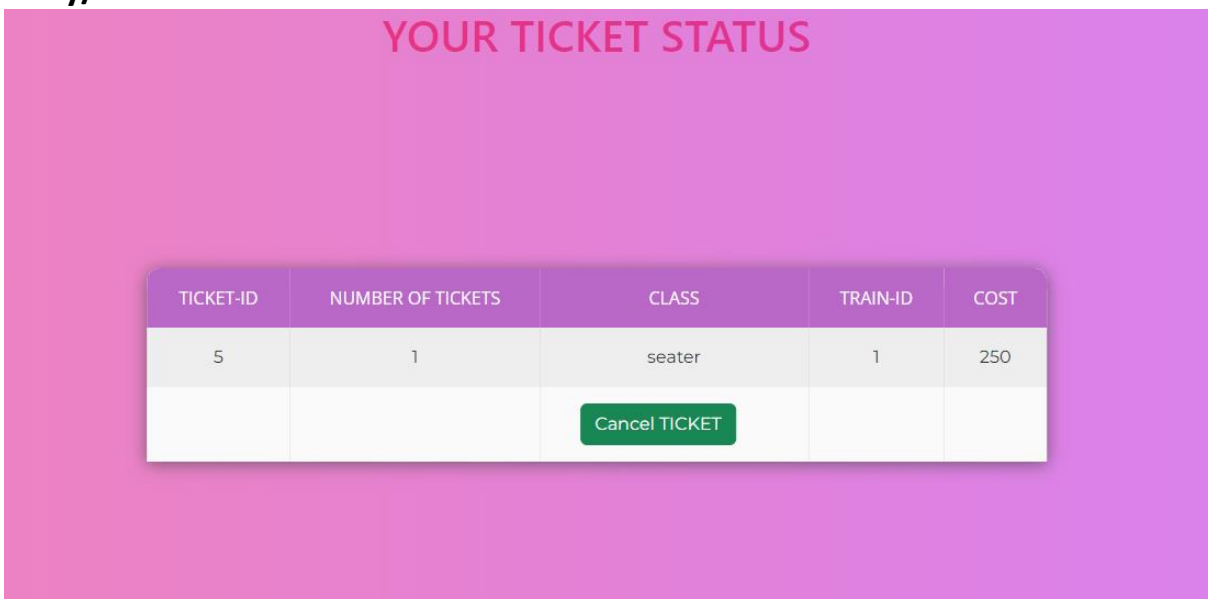
Wallet before the ticket booking I.e., -2000

)001024@iiti.ac.in	pillu	0989	kujbcd	fef	djc	sdsdds	sddvvd	hgj	750
akarthikeya@gmail.com	surya	9898	rf	ghd	feff	jdj	yjf	dsdssf	2000

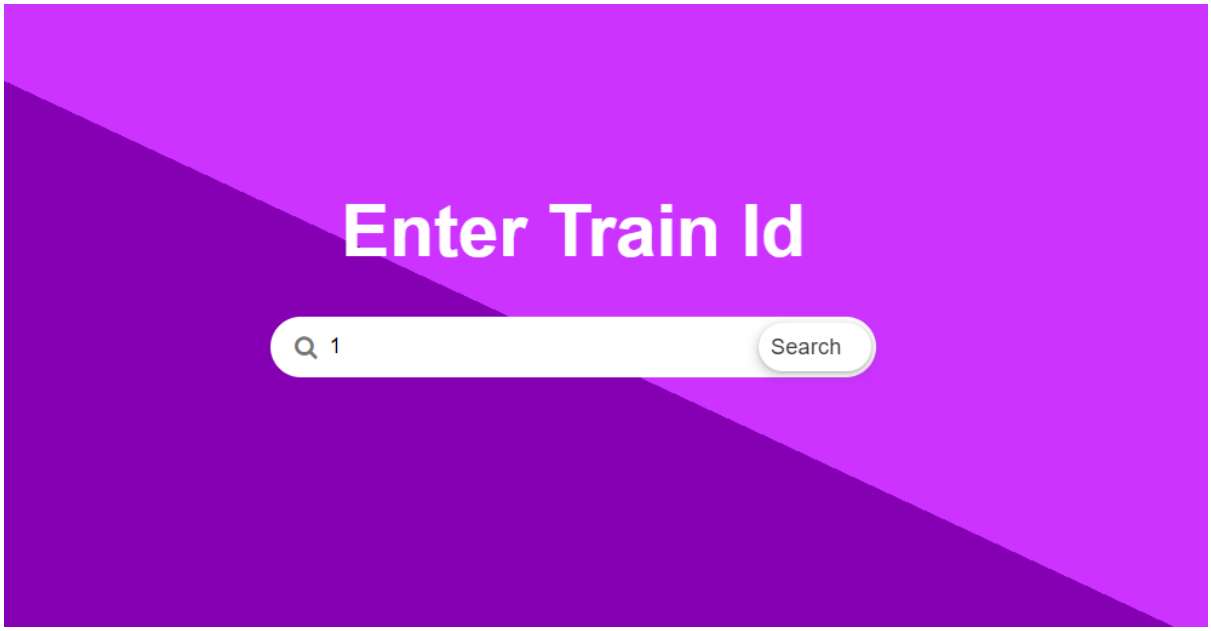
Wallet after the ticket booking I.e. - 1650

4@iiti.ac.in	pillu	0989	kujbcd	fef	djc	sdsdds	sddvvd	hgj	750
eya@gmail.com	surya	9898	rf	ghd	feff	jdj	yjf	dsdssf	1650

Now after a customer click on cancellation button in the home page, he will be directed to the cancellation page where he can cancel his ticket. Like the ticket booking we have also implemented cancellation depending how early/late he cancels the ticket.

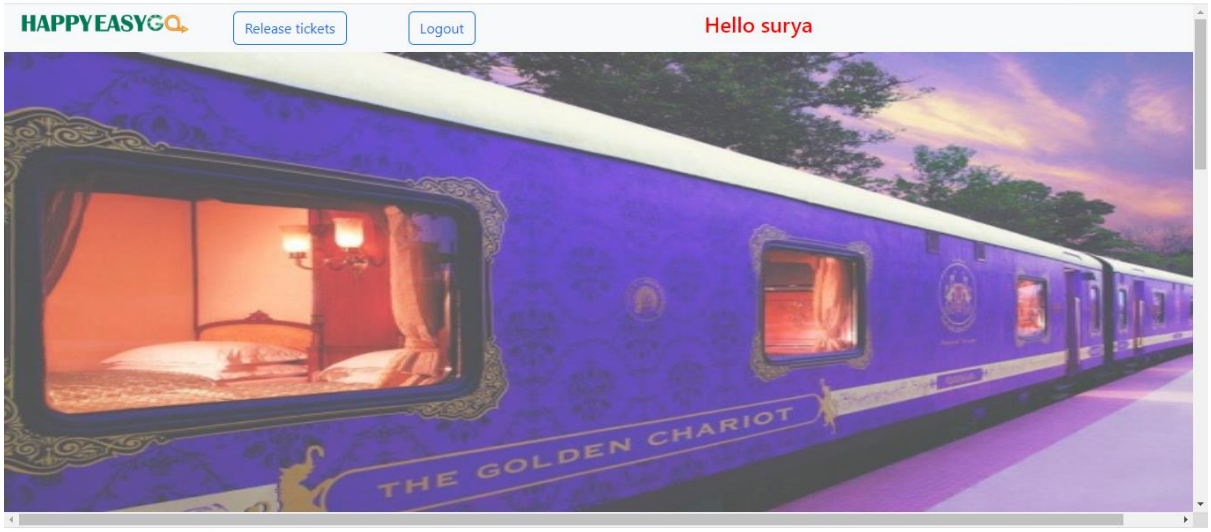


Customer can also check the train updates by entering the train id of that train.

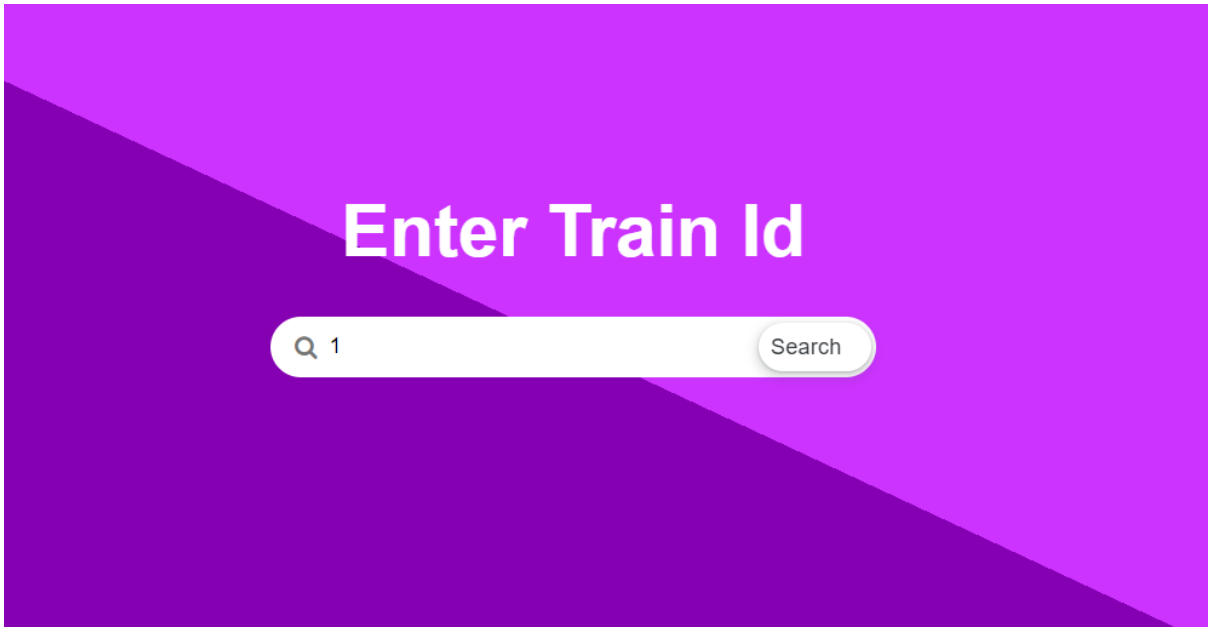


After entering the train id, he will be showed the corresponding train, its route etc.

Now in the case of emergency if a train fails then we will be informing all the customers who booked that train. So, we created an admin page which is shown below. This page appears when we login with a particular username/password.



The admin can enter the corresponding train id and all the corresponding customers will receive an email.



Below is the screenshot of email to all the customers.