Final Project Report
CS3120 - Database Management Systems Laboratory

Railway Reservation System
(BOOK MY TICKET)

Team members
Jagadeesh - 112001045
Anish - 112001020
Rakesh - 112001012
Veeraj - 111801021

Overview

Here we briefly describe about the features which we have implemented in our database system. Our reservation system will allow the user to make an account so that he/she can book or cancel tickets and can see what trains are available between two stations on a particular date. A user can see all bookings and cancellations made by him . A user has to be registered in order to book tickets. An unregistered user or Passenger can still check the PNR status whether the ticket is confirmed or in the waiting list and he can also check the train status. The model of our train follows that of a real train in India. A train has multiple classes like AC, NON-AC. Since a train moves between several railway stations, we give the flexibility of choosing the start and end station to the user.

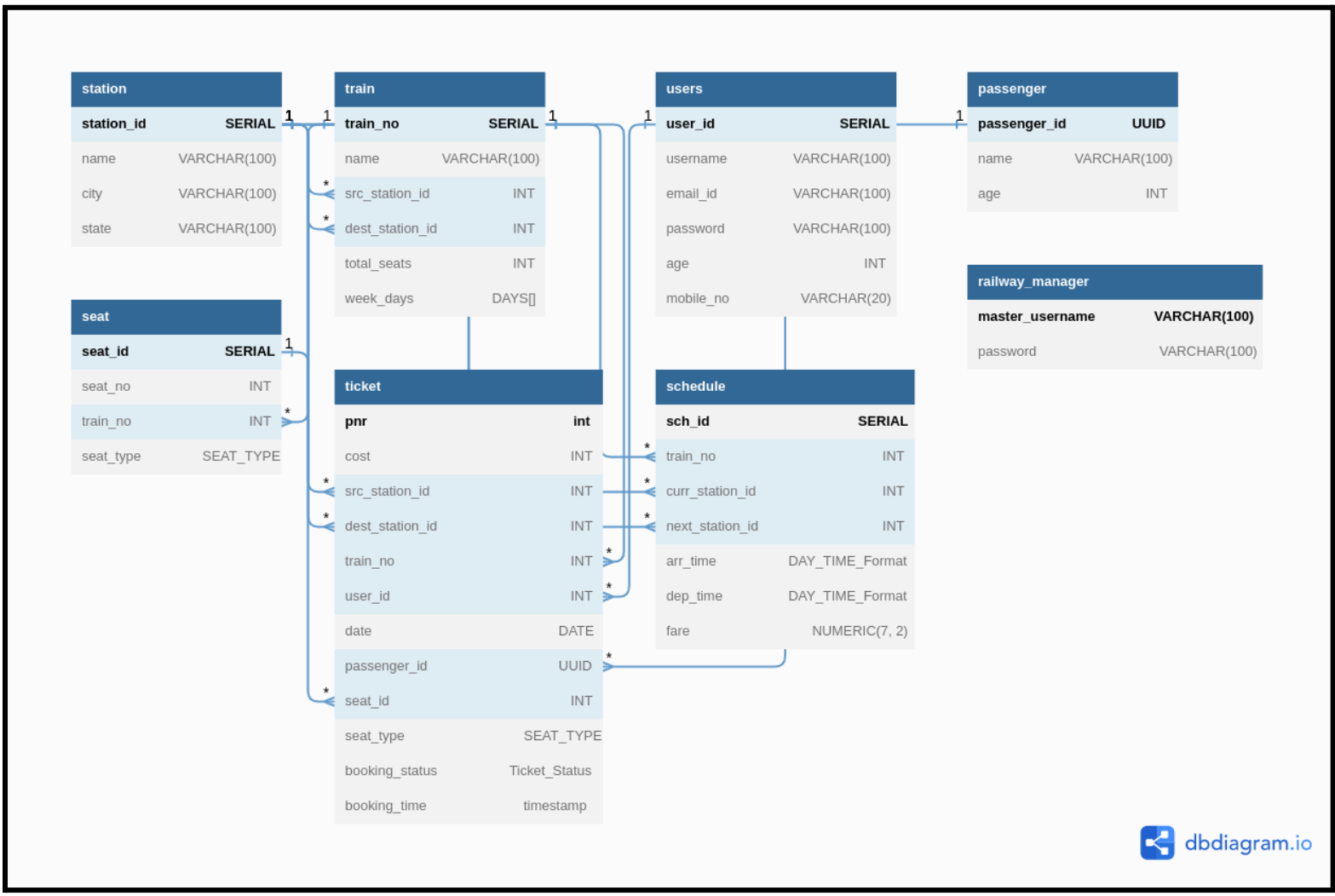
Implementation for running the application

We run our entire system using python. Thus all the update, insert and delete queries are called by python scripts. This is done using the sqlalchemy package. We also implemented an interface to interact with our database system by using python streamlit library. Our final system consisting of the entire database along with the web interface is shown in the end.

INTRODUCTION

An application created specifically to automate and manage the process of ordering train tickets for passengers is known as a railway reservation system. The traditional manual booking system's problems are being addressed by this system, which will also increase the overall effectiveness of the railway booking procedure. The technology intends to offer users a simple way to search for and reserve train tickets to their selected locations. The workload of the railway workers who conduct the booking process is also made simpler.

Relational diagram



Integrity constraints and General constraints

Tables Schema

USERS:
This table represents the user details like user_id, username, email_id, password, age, mobile. Here user_id is the primary key. Email_id should be unique as it is used for login and All the attributes have NOT NULL constraints. There are check constraints on age, email attributes to check age is positive integer and email id should be in proper format.

RAILWAY_MANAGER:
This table represents the RAILWAY MANAGER login credentials like username, password. Here the username is the primary key and all the attributes are NOT NULL.

RAILWAY_STATION:
This table represents the railway station details like station_id, name, city, state. Here station_id is the primary key. All the attributes have NOT NULL constraints.

TRAIN:
This table represents the train details like train no, train name, starting station id , terminating station id, on which days it will run in the week. Here train_no is the primary key. src_station_id , dest_station_id are the foreign keys references from the railway_station table. All the attributes have NOT NULL constraints. Check constraints on total_seats to make sure that the total seats are positive.

SEAT:
This table represents the seat details like seat_no, train_no, seat type, seat_id.Here seat_id is the primary key and train_no is the foreign key references from train table.In this table seat_no and train_no are unique and check constraint on seat_no to make sure that it is positive. All the attributes have NOT NULL constraints.

TICKET:

This table represents the ticket details like pnr, cost, source station , destination station, train no. ect.. Here pnr is the primary key constraint, src_station_id,dest_station_id are foreign key constraints referenced from the station table. Train_no is foreign key constraints referenced from the train table. Seat_id is foreign key constraints referenced from the seat table. Passenger_id is foreign key constraints referenced from the passenger table. Check constraints on the date. All attributes have NOT NULL constraints.

SCHEDULE:

This table represents the schedule details like train no, current station, next station, arrival time , departure time etc.. . Here sch_id is the primary key constraint and train_no, curr_station_id, next_station_id are the foreign key constraints. Check constraint is used on departure time to make sure that departure time is more than the arrival time.All the attributes have no null constraint.

PASSENGER:

This table consists of details of the passengers like passenger_id, name, age . Here passenger_id is the primary key. Check constraints for age to make sure that age is a positive integer.

Roles

We defined 3 roles of database:

- Admin: The admin has all the access to modify the entire database.
- User: All privileges on passengers table. Select privilege on ticket, train, station tables, Booking details.
- Passenger: He can only view pnr status.
- Railway Manager:The railway manager is responsible for updating the schedules for different trains. He has all privileges on stations, trains,and schedule tables.

views

- **Booking_details** :
  It shows the list of passengers whose tickets are confirmed and their details like mobile number, email, train number, train name, boarding station, departure station, boarding date, user name.This view has access to the railway manager role. Where he can view all the booking details of the passengers.
  This view also has access to the user but the user can only view his booking details rather than everyone's details.

- **Trains_availability -**
  It shows all the routes of each train in a database, used to check train availability between any two stations. It displays train number, stations along which it travels , arrival day and time at that station , halt time , fare of that route. This is used to check the available trains between stations. This view has access by railway manager and user role.
- **User_info -**
  It shows all the users who signed up , details shown are username, email, age. This view has access to the railway manager role.

Functionalities:

- **Available trains between two stations:**

By providing source station,destination station and date user can get a list of available trains between two stations on that date

- **Book Tickets:**
  A user needs to be a registered user to book tickets. To book tickets users need to select the source, destination stations and train. Number of passengers and their details and seat type.
- **SIGN UP**:
  To book a ticket he needs to be a registered user. So he needs to sign up so that he will be a registered user.
- **Cancel Ticket**:
  A user can cancel ticket by providing PNR no. of the ticket
- **ADD STATION:**
  Railway manager can add new station in the station table
- **ADD TRAIN :**
  Railway manager can add new train in the station table
- **ADD Schedule:**

Railway manager can add a new schedule to the schedule table.

TRIGGERS

- Schedule_log - after insert/update/delete on schedule table. Whenever any new route or any change in route of train or any train route gets cancelled on the schedule table , this triggers inserts data into a log table (updated_stations_schedule). This table contains new data and modified data.
- Assign_seat - after insert or update on the ticket table. It executes the book_seat function.
  Whenever a user books a ticket, this trigger allocates a seat and updates ticket status to confirm else the ticket remains in the waiting status. When a user cancels a ticket , this trigger cancels the ticket and updates seats and assigns seats to waiting list tickets (takes the first ticket among the waiting list).
- Check_email - before insert on the user table, it checks whether the user email is in correct format or not and also unique.
- Check_pnr - when a user cancels the ticket , it checks whether the pnr provided by the user is valid or not and makes sure it is cancelling the given pnr ticket.

Functions

- GET_TRAINS - Given source station ,destination station and date, it returns list of available trains ,seat availability in that trains and week days in which it runs
- BOOK_TICKETS - Given the details of the passenger, boarding and departure stations and date. Book a ticket by initially keeping ticket status as waiting.
- CANCEL_BOOKING - Given pnr number it cancels the ticket.
- ADD_RAILWAY_STATION - inserts new station and its details like city, state into railway station table.
- ADD_SCHEDULE - inserts new routes ,trains, delay time, price to schedule table
- ADD_USER - when user signups in frontend , his details are inserted into user table

We also wrote some of the helper functions like get_train_name, get_train_station_name, validate_pnr,get_journey_at_station which are used in the most used queries.

Indexes

S.No	Table name	Index name	Index Type and column
1	seat	seat_train_no	HASH(train_no)
2	ticket	ticket_src_station_id	HASH(src_station_id)
3	ticket	ticket_dest_station_id	HASH(dest_station_id)
4	ticket	ticket_train_no	HASH(train_no)
5	ticket	ticket_pid	HASH(passenger_id)
6	ticket	ticket_user_id	HASH(user_id)
7	ticket	ticket_seat_id	HASH(seat_id)
8	ticket	ticket_date	BTREE(date)
9	reservation	temp_res_sch	HASH(sch_id)
10	reservation	temp_res_seat_id	HASH(seat_id)
11	schedule	schedule_curr_station_id	HASH(curr_station_id)
12	schedule	schedule_next_station_id	HASH(next_station_id)

In almost all the queries we used equality based where condition. After the creation of the index we got less execution time compared to the before creation of the index.

- Since the book_seat function, the most frequent query in our database, is executed inside the trigger for each new booking or cancellation of a booking, we are creating a hash index on the train_no column of the seat table.

- Next, we are creating hash indexes on the train_no, curr_station_id, and next_station_id columns of the schedule table and src_station_id and dest_station_id columns of the train table as these columns are also the most accessed fields inside many different query functions like get_trains, get_train_schedule, get_trains_schedule_at_stationb, get_fare, and our view

stations_trains. Next, since they are frequently accessed in queries, we have hash indexes on the ticket table's foreign keys and BTree indexes on its date column.

Some Useful Queries

Query 1:

For a Given pnr shows booking status

```
SELECT booking_status FROM ticket WHERE pnr =1001;
```

Query 2:

It shows schedule of all trains (train name, current station, next station , arrival time, halt time, running days, price)

```
SELECT sch.train_no, tt.name AS train_name, curr.name AS
current_station, nxt.name AS next_station, (sch.arr_time).time
AS arrival_time, (sch.dep_time).time - (sch.arr_time).time AS
halt_time,
ARRAY_TO_STRING(get_updated_days((sch.arr_time).day_of_journey
,tt.week_days), '/', '')AS running_days,
get_fare(tt.name,curr.name,nxt.name) as price
FROM schedule AS sch JOIN train AS tt ON sch.train_no =
tt.train_no JOIN railway_station AS curr ON
sch.curr_station_id = curr.station_id LEFT JOIN
railway_station AS nxt ON sch.next_station_id = nxt.station_id
WHERE nxt.name IS NOT NULL ORDER BY sch.train_no ASC,
sch.arr_time ASC;
```

Query-3:
It gives booking details of passenger (pnr , passenger name, mobile number, age, train number, seat number, seat type, booking status, bording_station, destination_station)

```
   SELECT pnr, u.mobile_no as user_mobile, p.pid as
passenger_id, p.name as passenger_name, p.age as
passenger_age, t.train_no as train_no, t.seat_id, t.seat_type,
t.booking_status,
   t.src_station_id as bording_station, t.dest_station_id as
destination_station, t.date as bording_date
   from ticket as t natural join users as u inner join
passenger as p on p.pid = t.pid;
```

Query-4:
Given train number gives its route

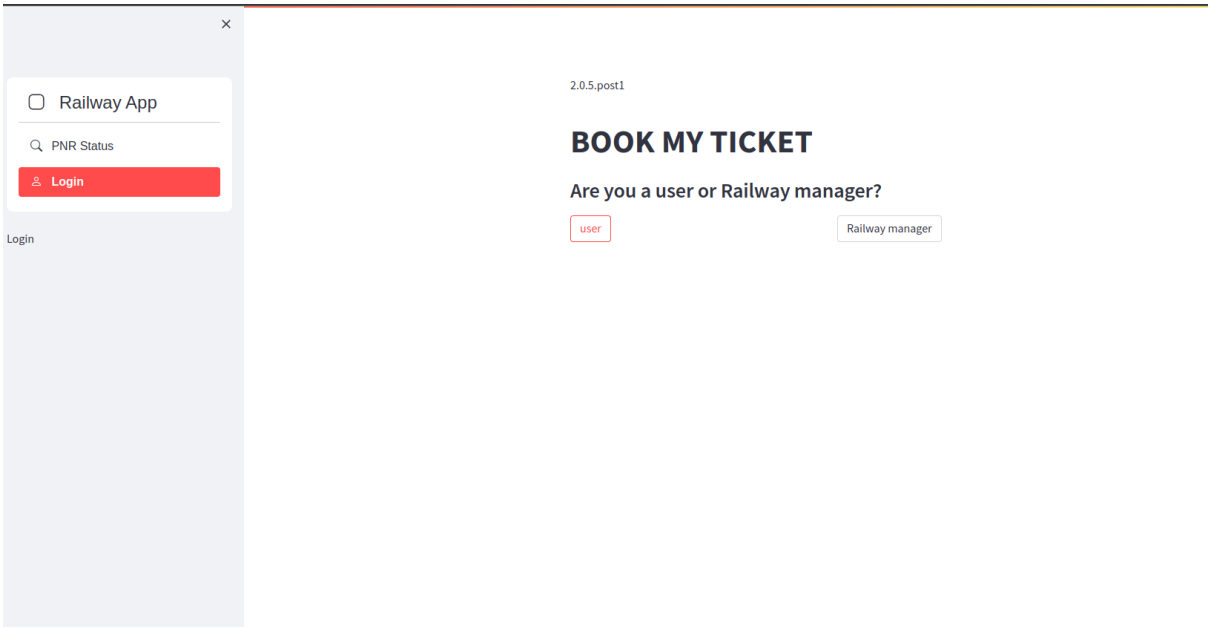
```
SELECT st.current_station,
          st.next_station,
          st.arrival_time,
          st.departure_time,
          st.arrival_days,
          st.running_days,
      FROM trains_availability AS st
      WHERE st.train_no = 4;
```

Query-5:
For Given train number shows total number of seats

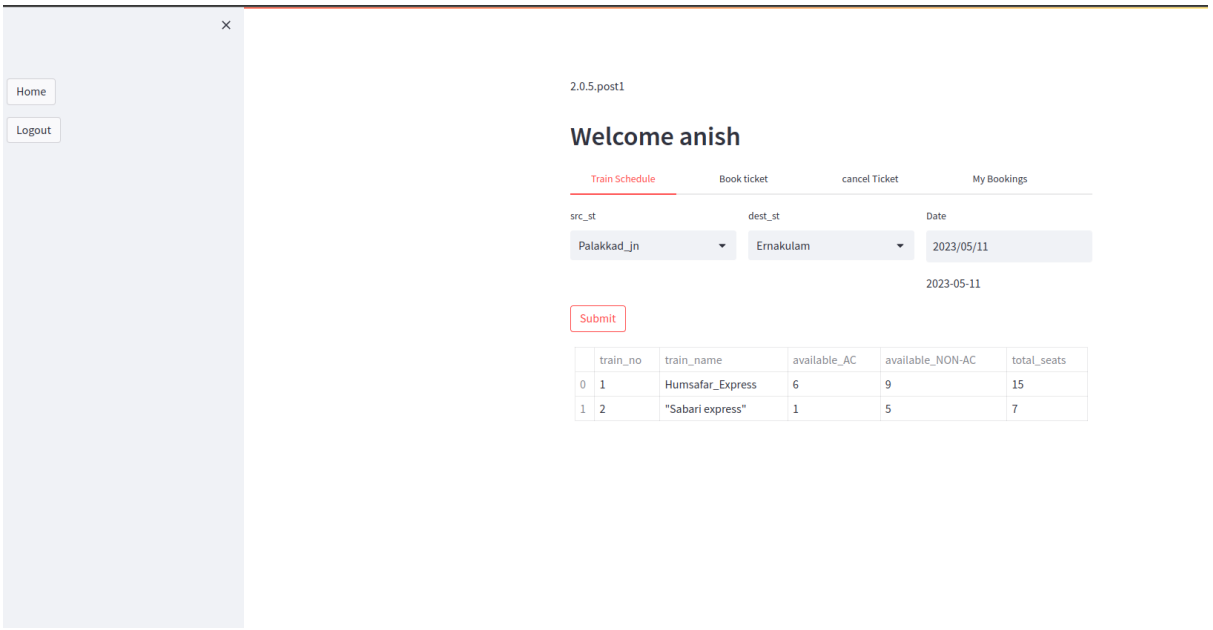
```
SELECT COUNT(seat_id) FROM seat where train_no = 4 GROUP BY
train_no;
```

Interface for database:

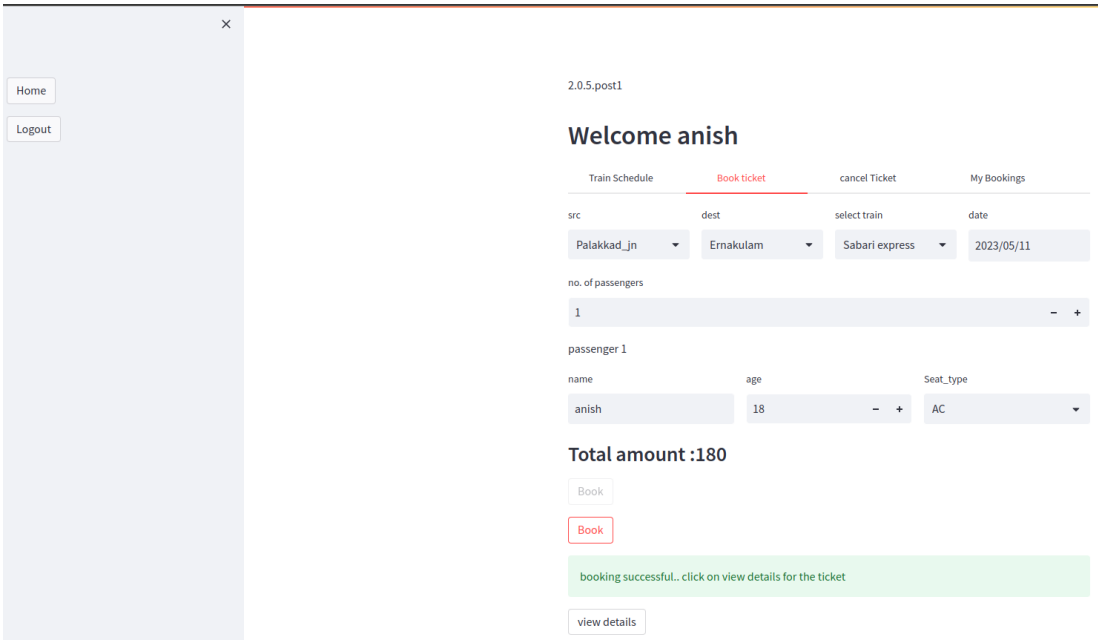
Here are a few screenshots demonstrating our implemented system.
To login you need to select the role like user or railway manager

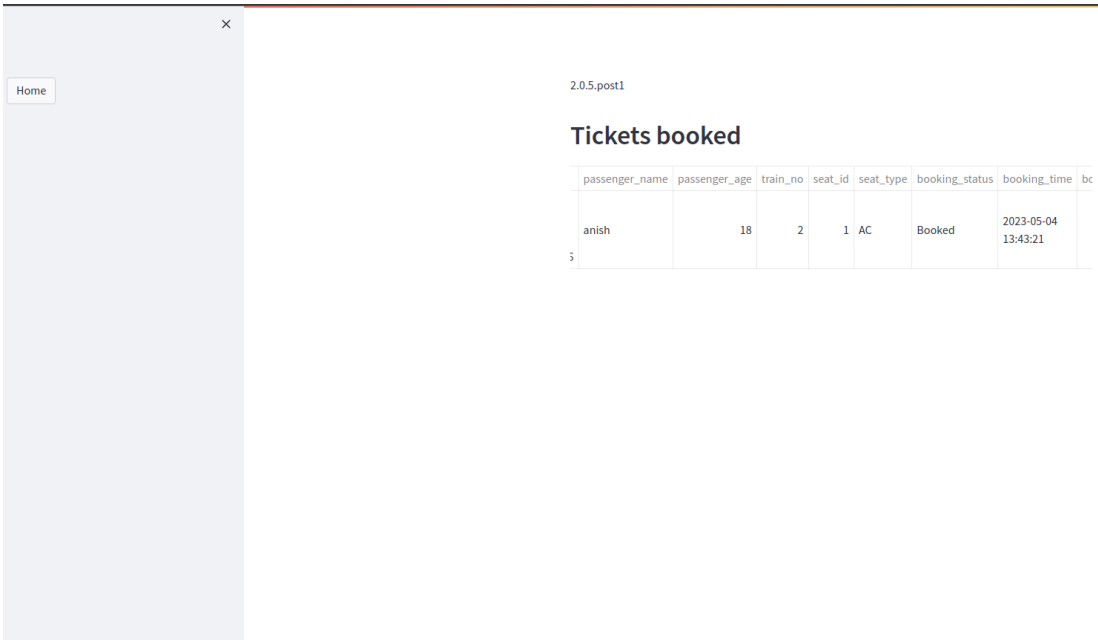


Logged into user account as anish, and search trains available between two stations

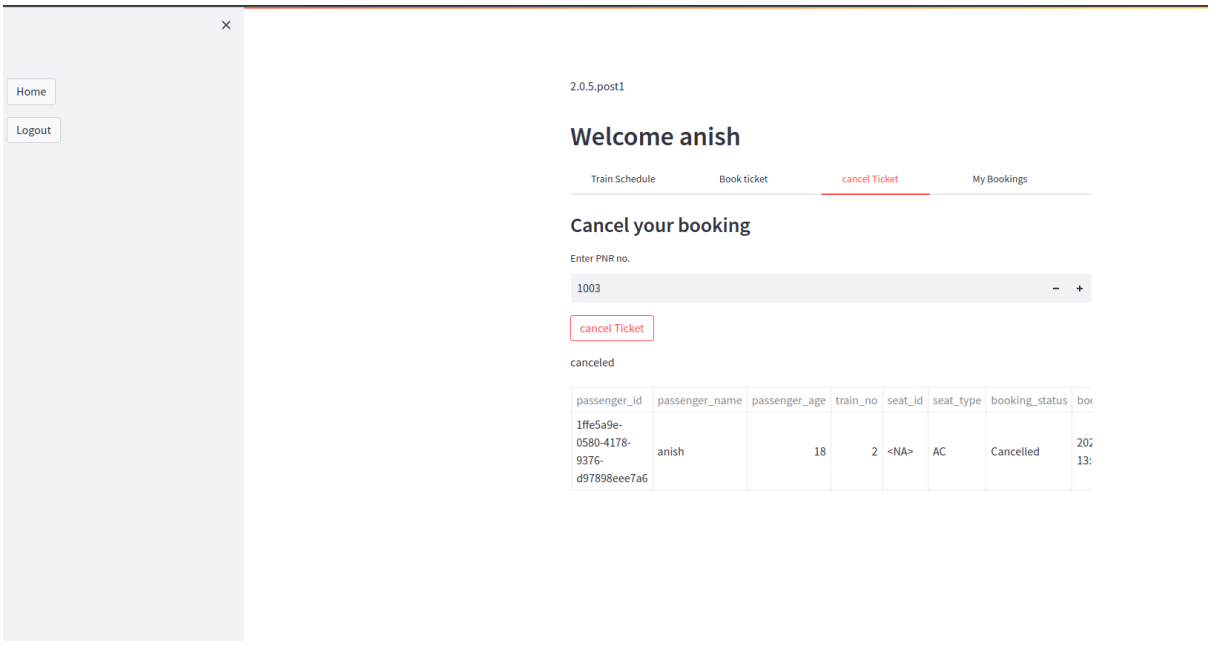


Ticket booking





Cancelling the booked Tickets



Logged in as Railway manager