



**Department of Computer Engineering**

**Academic Year 2020-2021**



### Team

TE B4 60004180086 – Rushabh Chheda

TE B4 60004180078 – Ravi Patel

TE B4 60004180123 – Yash Dalvi

TE B4 60004180059 – Neel Shah

# Railway Reservation System



## Department of Computer Engineering

Academic Year 2020-2021

### Abstract

The Railway Reservation System facilitates the passengers 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 Project maintaining the 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 advanced. It is mainly used for long route. On-line reservation has made the process for the reservation of seats very much easier than ever before. In our country India, there are number of counters for the reservation of the seats and one can easily make reservations and get tickets. Then this project contains entity relationship model diagram based on railway reservation system and introduction to relation model. There is also design of the database of the railway reservation system based on relation model. Example of some SQL queries to retrieves data from rail management database.

### Introduction

Database is an organized collection of data. The data is typically organized to model aspects of reality in a way that supports processes requiring information. A DBMS makes it possible for end users to create, read, update and delete data in a database. The DBMS essentially serves as an interface between the database and end users or application programs, ensuring that data is consistently organized and remains easily accessible. The DBMS manages three important things: the data, the database engine that allows data to be accessed, locked and modified and the database schema, which defines the database's logical structure. These three foundational elements help provide concurrency, security, data integrity and uniform administration procedures. The DBMS can offer both logical and physical data independence. That means it can protect users and applications from needing to know where data is stored or having to be concerned about changes to the physical structure of data.

#### A. Problem Statement

To design a database system with a Website frontend to facilitate automatic, validated and error-free purchase, reservation and transaction of train tickets.

#### B. Problem Description

This project is about creating the database about Railway Reservation System. The railway reservation system facilitates the passengers to enquire about the trains available based on 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 records of different trains, train status, and passengers. 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. Passengers can book their tickets for the train in which seats are available. For this, passenger must provide the desired train number and the date for which ticket is to be booked. Before booking a



## Department of Computer Engineering

Academic Year 2020-2021

ticket for a passenger, the validity of train number and booking date is checked. Once the train number and booking date are validated, it is checked whether the seat is available. If yes, the ticket is booked with confirm status and corresponding ticket ID is generated which is stored along with other details of the passenger. The ticket once booked can be cancelled at any time. For this, the passenger must provide the ticket ID (the unique key). The ticket ID is searched, and the corresponding record is deleted. With this, the first ticket with waiting status also gets confirmed. List of Assumption Since the reservation system is very large in reality, it is not feasible to develop the case study to that extent and prepare documentation at that level. Therefore, a small sample case study has been created to demonstrate the workings of the proposed system.

### C. Motivation Scope

The main purpose of maintaining database for Railway Reservation System is to reduce the manual errors involved in the booking and cancelling of tickets and make it convenient for the customers and providers to maintain the data about their customers and also about the seats available at them. Due to automation many loopholes that exist in the manual maintenance of the records can be removed. The speed of obtaining and processing the data will be fast. For future expansion the proposed system can be web enabled so that clients can make various enquiries about trains between stations. Due to this, sometimes a lot of problems occur, and they are facing many disputes with customers. To solve the above problem, we design a database which includes customer details, availability of seats in trains, no of trains and their details.

### D. Proposed Solution

The solution proposed is to use a 3-tier DBMS architecture consisting of a static and dynamic database, a website to serve as a frontend presentation layer for users to interact with and a server side backend layer that connects the two layers in a secure and validated manner while providing authentication services.

## Technologies Used

### Django





## Department of Computer Engineering

Academic Year 2020-2021

Django is a Python-based free and open-source web framework that follows the model-template-views (MTV) architectural pattern and has an extensive built in ORM which supports multiple different databases. It is maintained by the Django Software Foundation (DSF), an American independent organization established as a 501(c)(3) non-profit.

Django's primary goal is to ease the creation of complex, database-driven websites. The framework emphasizes reusability and "pluggability" of components, less code, low coupling, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings files and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.

Some well-known sites that use Django include PBS, Instagram, Mozilla, The Washington Times, Disqus, Bitbucket, and Nextdoor.

### Django REST Framework



Django Rest Framework (DRF) is a powerful and flexible toolkit for building Web APIs. It is an extension library for Django and provides

- A Web Browsable REST API
- Authentication Policies and different forms of Authentication such as Basic, Session and Token Auth
- Serialization of ORM and non-ORM data sources
- Function and Class bases API Views



Department of Computer Engineering

Academic Year 2020-2021

## MySQL



MySQL is a free and open-source relational database management system (RDBMS) written in C and C++. It is available under the terms of the GNU General Public License and is also available under a variety of proprietary licenses.

MySQL has stand-alone clients that allow users to interact directly with a MySQL database using SQL, but more often MySQL is used with other programs to implement applications that need relational database capability. MySQL is a component of the LAMP web application software stack (and others), which is an acronym for Linux, Apache, MySQL, Perl/PHP/Python. MySQL is used by many database-driven web applications, including Drupal, Joomla and WordPress. MySQL is also used by many popular websites, including Facebook, Flickr, Twitter, and YouTube.

## jQuery



jQuery is a lightweight, "write less, do more", JavaScript library. The purpose of jQuery is to make it much easier to use JavaScript on your website. jQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish and wraps them into methods that you can call with a



## Department of Computer Engineering

Academic Year 2020-2021

single line of code. jQuery also simplifies a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation. The jQuery library contains the following features:

- HTML/DOM manipulation
- CSS manipulation
- HTML event methods
- Effects and animations
- AJAX
- Utilities

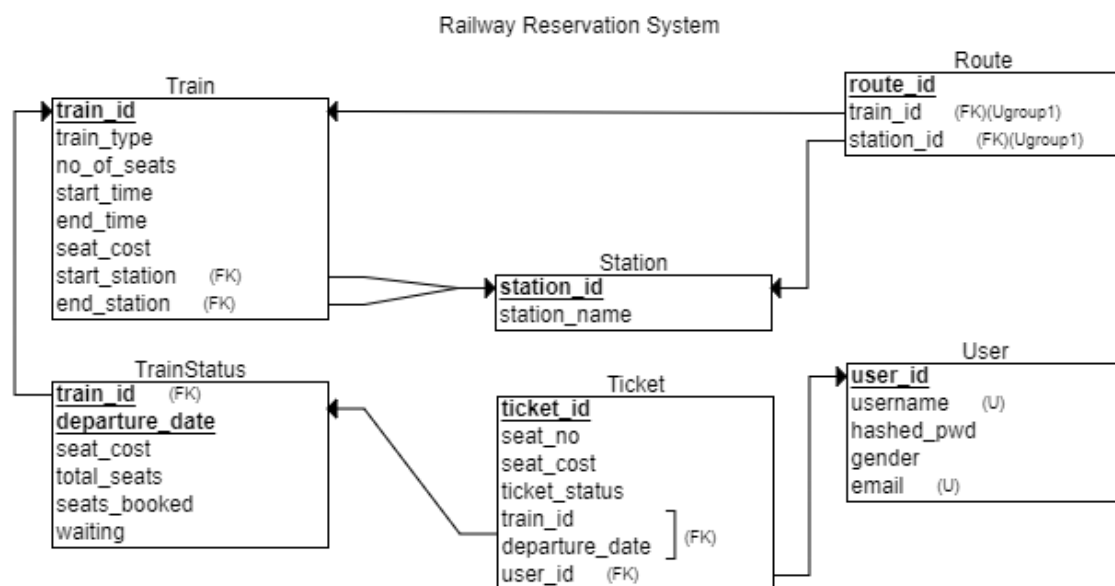
Axios



Axios is a promise based HTTP Client for the browser and node.js useful for making XMLHttpRequests, html requests and implementing Token Authorization.

## Implementation

### Schema Diagram

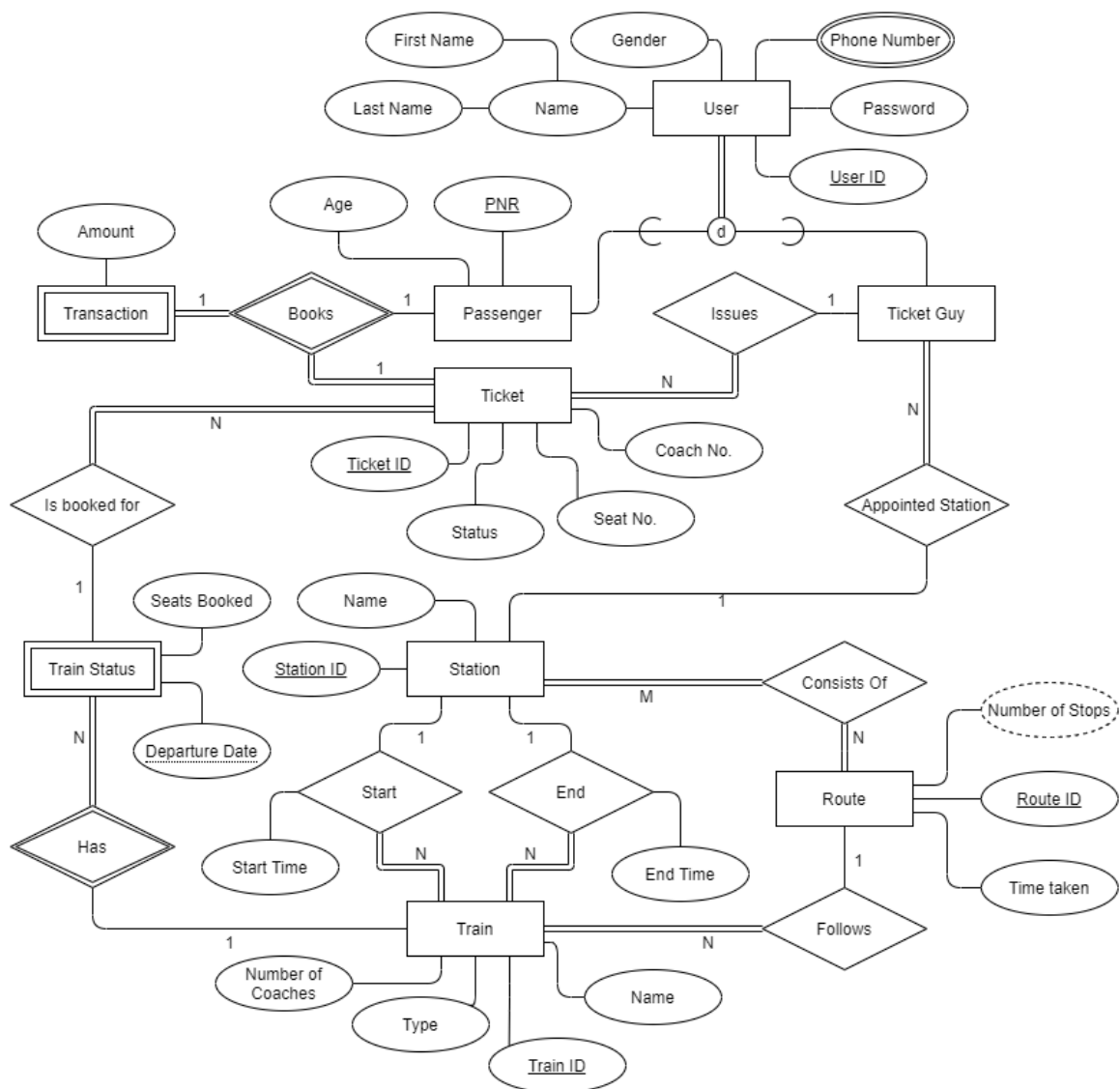




**Department of Computer Engineering**

**Academic Year 2020-2021**

**ER Diagram**







## Department of Computer Engineering

Academic Year 2020-2021

### Website Implementation

#### Admin Page

Django administration

WELCOME, DS VIEW SITE / CHANGE PASSWORD / LOG OUT

Site administration

AUTH TOKEN

Tokens [+ Add](#) [Change](#)

AUTHENTICATION AND AUTHORIZATION

Groups [+ Add](#) [Change](#)

Users [+ Add](#) [Change](#)

RAILWAYAPP

Profiles [+ Add](#) [Change](#)

Routes [+ Add](#) [Change](#)

Stations [+ Add](#) [Change](#)

Ticket2s [+ Add](#) [Change](#)

Trains [+ Add](#) [Change](#)

Trainstatus2s [+ Add](#) [Change](#)

Recent actions

My actions

+ Dakshin Express R | Bangalore > Delhi | Delhi

Route

+ Dakshin Express R | Bangalore > Delhi | Agra

Route

+ Dakshin Express R | Bangalore > Delhi | Pune

Route

+ Dakshin Express R | Bangalore > Delhi | Bangalore

Route

+ Dakshin Express | Delhi > Bangalore | Bangalore

Route

+ Dakshin Express | Delhi > Bangalore | Pune

Route

+ Dakshin Express | Delhi > Bangalore

#### Main Page

RRS Home Schedule About

Sign-Up Login

Book your Ticket

From

From\*

To

To\*

Date

dd-mm-yyyy

Train Type

Sleeper

Submit





## Department of Computer Engineering

Academic Year 2020-2021

### Signup Page

[RRS](#) [Home](#) [Schedule](#) [About](#) [Sign-Up](#) [Login](#)

### Register

**Username**  
Enter Username

**Password**  
Enter Password

**First name**  
Enter First name

**Last name**  
Enter Last name

**Email**  
Enter Email

**Gender**  
Male

Login

### Login Page

[RRS](#) [Home](#) [Schedule](#) [About](#) [Sign-Up](#) [Login](#)

### Login

**Username**  
Enter Username

**Password**  
Enter Password

Login





## Department of Computer Engineering

Academic Year 2020-2021

### Schedule Page

RRS

Home

Schedule

About

Sign-Up

Login

Search for Train Route

Train Name

Dakshin Express

Submit

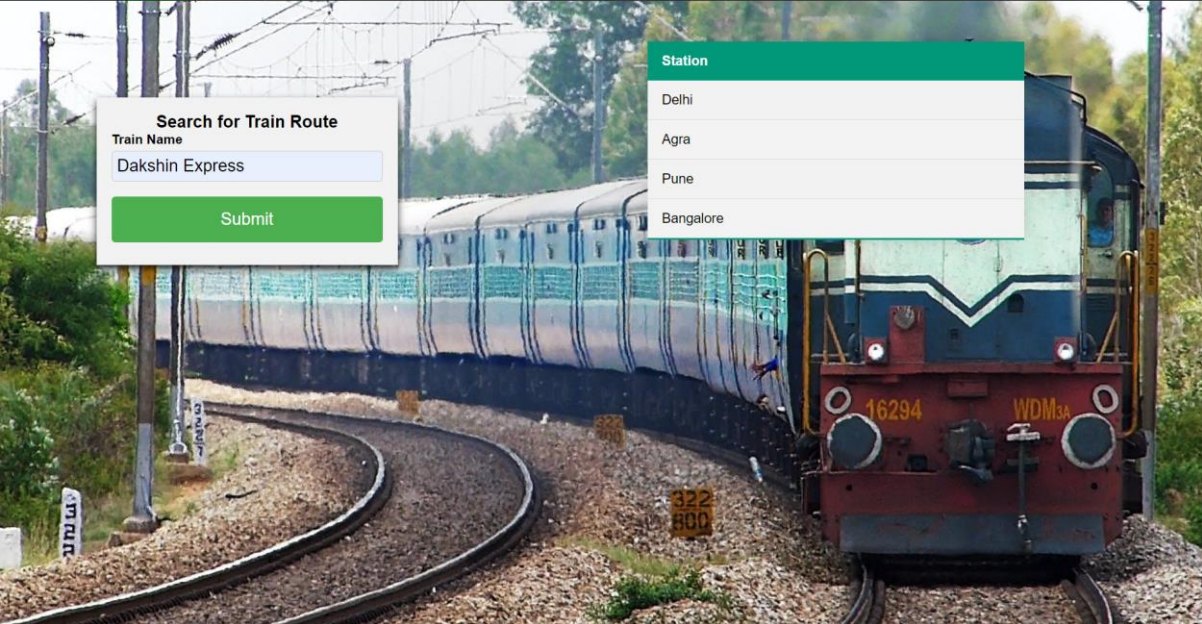
Station

Delhi

Agra

Pune

Bangalore



### User History

RRS

Home

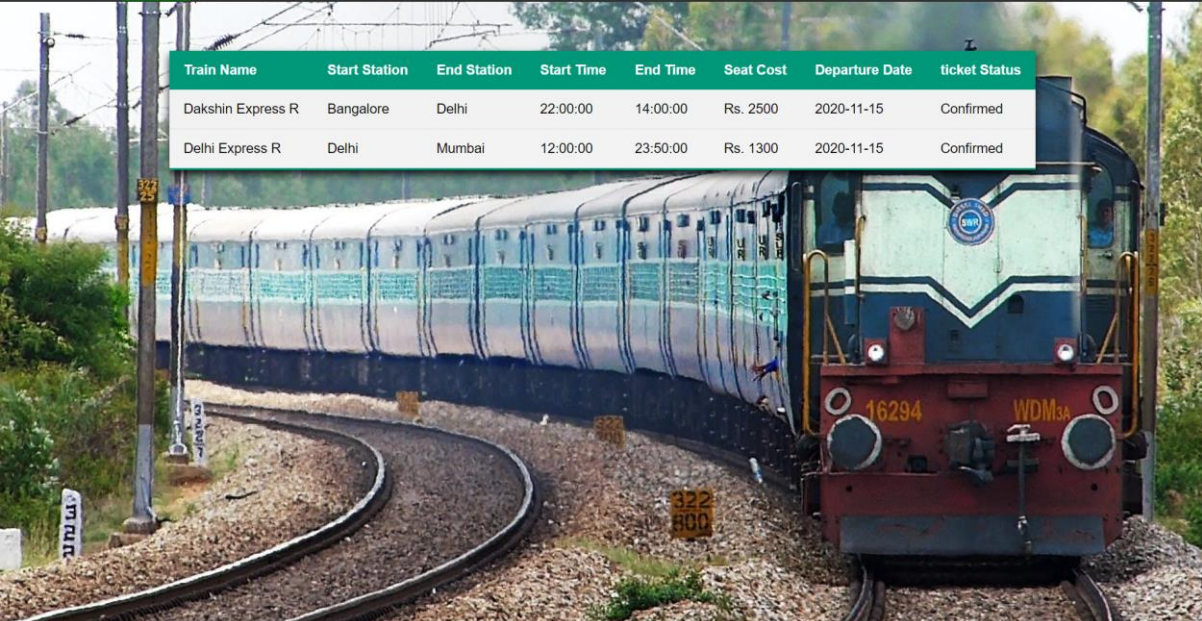
Schedule

About

Hello, Dolly

Logout

| Train Name        | Start Station | End Station | Start Time | End Time | Seat Cost | Departure Date | ticket Status |
|-------------------|---------------|-------------|------------|----------|-----------|----------------|---------------|
| Dakshin Express R | Bangalore     | Delhi       | 22:00:00   | 14:00:00 | Rs. 2500  | 2020-11-15     | Confirmed     |
| Delhi Express R   | Delhi         | Mumbai      | 12:00:00   | 23:50:00 | Rs. 1300  | 2020-11-15     | Confirmed     |





## Department of Computer Engineering

Academic Year 2020-2021

### General Search Page

| Train Name      | Start Station | End Station | Start Time | End Time   | Seat Cost | Purchase            |
|-----------------|---------------|-------------|------------|------------|-----------|---------------------|
| Ajanta Express  | Bangalore     | Surat       | 12:30 p.m. | 11:50 p.m. | Rs. 900   | <a href="#">Buy</a> |
| Delhi Express R | Delhi         | Mumbai      | noon       | 11:50 p.m. | Rs. 1300  | <a href="#">Buy</a> |

### Database Implementation

#### User Table

MySQL Workbench - Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Indexes in Table

| Visible                             | Key     | Type  | Unique | Columns  |
|-------------------------------------|---------|-------|--------|----------|
| <input checked="" type="checkbox"/> | PRIMARY | BTREE | YES    | id       |
| <input checked="" type="checkbox"/> |         | BTREE | YES    | username |

Index Details

Key Name:  
Index Type:  
Allows NULL:  
Cardinality:  
Comment:  
User Comment:

Columns in table

| Column       | Type         | Nullable | Indexes  |
|--------------|--------------|----------|----------|
| id           | int          | NO       | PRIMARY  |
| password     | varchar(128) | NO       |          |
| last_login   | datetime(6)  | YES      |          |
| is_superuser | tinyint(1)   | NO       |          |
| username     | varchar(150) | NO       | username |
| first_name   | varchar(150) | NO       |          |
| last_name    | varchar(150) | NO       |          |
| email        | varchar(254) | NO       |          |
| is_staff     | tinyint(1)   | NO       |          |
| is_active    | tinyint(1)   | NO       |          |
| date_joined  | datetime(6)  | NO       |          |





## Department of Computer Engineering

Academic Year 2020-2021

### User Token Table

MySQL Workbench interface showing the structure of the User Token Table. The table is named 'railway\_miniproject.auth\_token'. The 'Columns in table' section lists the following columns:

| Column  | Type        | Nullable | Indexes |
|---------|-------------|----------|---------|
| key     | varchar(40) | NO       | PRIMARY |
| created | datetime(6) | NO       |         |
| user_id | int         | NO       | user_id |

The 'Indexes in Table' section shows the following indexes:

| Visible                             | Key     | Type  | Uni... | Columns |
|-------------------------------------|---------|-------|--------|---------|
| <input checked="" type="checkbox"/> | PRIMARY | BTREE | YES    | key     |
| <input checked="" type="checkbox"/> | user_id | BTREE | YES    | user_id |

The 'Index Details' section shows the following details for the selected index:

Key Name:   
Index Type:   
Allows NULL:   
Cardinality:   
Comment:   
User Comment:

Buttons: Drop Index, Create Index for Selected Columns...

### Train Table

MySQL Workbench interface showing the structure of the Train Table. The table is named 'railway\_miniproject.train'. The 'Columns in table' section lists the following columns:

| Column        | Type        | Nullable | Indexes  |
|---------------|-------------|----------|--|
| train_id      | int         | NO       | PRIMARY  |
| train_type    | varchar(1)  | NO       |  |
| train_name    | varchar(20) | NO       | train_name   |
| no_of_seats   | int         | NO       |  |
| start_time    | time(6)     | NO       |  |
| end_time      | time(6)     | NO       |  |
| seat_cost     | int         | NO       |  |
| end_station   | int         | NO       | train_end_station_e17ea8a9_fk_station_station_id   |
| start_station | int         | NO       | train_start_station_aeb9dfb3_fk_station_station_id |

The 'Indexes in Table' section shows the following indexes:

| Visible                             | Key                      | Type  | Uni... | Columns       |
|-------------------------------------|--------------------------|-------|--------|---------------|
| <input checked="" type="checkbox"/> | PRIMARY                  | BTREE | YES    | train_id      |
| <input checked="" type="checkbox"/> | train_name               | BTREE | YES    | train_name    |
| <input checked="" type="checkbox"/> | train_end_station_e...   | BTREE | NO     | end_station   |
| <input checked="" type="checkbox"/> | train_start_station_a... | BTREE | NO     | start_station |

The 'Index Details' section shows the following details for the selected index:

Key Name:   
Index Type:   
Allows NULL:   
Cardinality:   
Comment:   
User Comment:

Buttons: Drop Index, Create Index for Selected Columns...



## Department of Computer Engineering

Academic Year 2020-2021

### Train Status Table

The screenshot shows the MySQL Workbench interface with the 'railway\_miniproject.trainstatus2' table selected. The 'Indexes in Table' section shows two indexes: a PRIMARY index on 'id' and a BTREE index on 'trainstatus2\_train\_id, departure\_date'. The 'Columns in table' section lists the following columns:

| Column         | Type | Nullable | Indexes  |
|----------------|------|----------|--|
| id             | int  | NO       | PRIMARY  |
| seats_booked   | int  | NO       |  |
| departure_date | date | NO       | trainstatus2_train_id_departure_date_1f3f66ca_uniq |
| seat_cost      | int  | NO       |  |
| total_seats    | int  | NO       |  |
| waiting        | int  | NO       |  |
| train_id       | int  | NO       | trainstatus2_train_id_departure_date_1f3f66ca_uniq |

### Station Table

The screenshot shows the MySQL Workbench interface with the 'railway\_miniproject.station' table selected. The 'Indexes in Table' section shows two indexes: a PRIMARY index on 'station\_id' and a BTREE index on 'station\_name'. The 'Columns in table' section lists the following columns:

| Column       | Type        | Nullable | Indexes      |
|--------------|-------------|----------|--------------|
| station_id   | int         | NO       | PRIMARY      |
| station_name | varchar(30) | NO       | station_name |



## Department of Computer Engineering

Academic Year 2020-2021

### Ticket Table

The screenshot displays the MySQL Workbench interface for the 'railway\_miniproject.ticket2' table. The 'Indexes in Table' section shows three indexes: a PRIMARY index on 'ticket\_id', and two BTREE indexes on 'ticket2\_trainstatus\_id' and 'ticket2\_user\_id'. The 'Columns in table' section lists seven columns: 'ticket\_id' (int, PRIMARY), 'ticket\_status' (varchar(1)), 'seat\_cost' (int), 'seat\_no' (int), 'trainstatus\_id' (int), and two foreign key columns: 'trainstatus\_id\_id' and 'user\_id'.

| Visible                             | Key     | Type  | Uni... | Columns                |
|-------------------------------------|---------|-------|--------|------------------------|
| <input checked="" type="checkbox"/> | PRIMARY | BTREE | YES    | ticket_id              |
| <input checked="" type="checkbox"/> |         | BTREE | NO     | ticket2_trainstatus_id |
| <input checked="" type="checkbox"/> |         | BTREE | NO     | ticket2_user_id        |

| Column            | Type       | Nullable | Indexes   |
|-------------------|------------|----------|---|
| ticket_id         | int        | NO       | PRIMARY   |
| ticket_status     | varchar(1) | NO       |   |
| seat_cost         | int        | NO       |   |
| seat_no           | int        | NO       |   |
| trainstatus_id    | int        | NO       |   |
| trainstatus_id_id | int        | NO       | ticket2_trainstatus_id_id_9851deec_fk_trainstatus2_id |
| user_id           | int        | NO       | ticket2_user_id_9ff0a2b9_fk_auth_user_id              |

### Route Table

The screenshot displays the MySQL Workbench interface for the 'railway\_miniproject.route' table. The 'Indexes in Table' section shows three BTREE indexes: a PRIMARY index on 'id', and two on 'route\_train\_id\_station\_id' and 'route\_station\_id'. The 'Columns in table' section lists three columns: 'id' (int, PRIMARY), 'station\_id' (int), and 'train\_id' (int).

| Visible                             | Key     | Type  | Uni... | Columns                   |
|-------------------------------------|---------|-------|--------|---------------------------|
| <input checked="" type="checkbox"/> | PRIMARY | BTREE | YES    | id                        |
| <input checked="" type="checkbox"/> |         | BTREE | YES    | route_train_id_station_id |
| <input checked="" type="checkbox"/> |         | BTREE | NO     | route_station_id          |

| Column     | Type | Nullable | Indexes   |
|------------|------|----------|---|
| id         | int  | NO       | PRIMARY   |
| station_id | int  | NO       | route_train_id_station_id_4bd244b0_uniq, route_station... |
| train_id   | int  | NO       | route_train_id_station_id_4bd244b0_uniq, route_train...   |





## Department of Computer Engineering

Academic Year 2020-2021

### Stored Procedure to perform General Search

The screenshot shows the MySQL Workbench interface with the 'general\_search' routine selected. The DDL text is as follows:

```
1 • CREATE DEFINER='root'@'localhost' PROCEDURE `general_search`(in train_t char(1), in station_a varchar(30), in station_b varchar(30))
2 begin
3     select s1.train_id, s1.train_type, s1.train_name,
4           s1.no_of_seats, s1.start_time, s1.end_time, s1.seat_cost,
5           s1.end_station, s1.start_station
6     from (
7         select *
8         from train natural join route natural join station
9         where train_type = train_t and station_name = station_a
10      ) as s1 inner join (
11         select *
12         from train natural join route natural join station
13         where train_type = train_t and station_name = station_b
14      ) as s2 on s1.train_id = s2.train_id
15     where s1.id < s2.id;
16 end
```

### Stored Procedure to create Train Status

The screenshot shows the MySQL Workbench interface with the 't\_status\_creator' routine selected. The DDL text is as follows:

```
1 • CREATE DEFINER='root'@'localhost' PROCEDURE `t_status_creator`(in d date)
2 begin
3     insert into trainstatus2(seats_booked, train_id, departure_date, seat_cost, total_seats, waiting)
4     select 0, train_id, d, seat_cost, no_of_seats, 0 from train;
5 end
```



**Department of Computer Engineering**

**Academic Year 2020-2021**

## Conclusion

Thus a 3-tier architecture system has been created to manage railway reservations. It provides automatic, validated and error-free database and allows user to reserve tickets from anywhere using the responsive online website and portal.

On conclusion of this project, we learned how to create database for a project real life application. There are certain features that we could further add to increase the usability of the web app such as:

1. Adding Multi-Language Support as a good portion of the target audience is not comfortable with English.
2. Using map APIs to display routes.
3. Implementing Chatbots to help users book their seat.
4. Implementing Captcha systems
5. Dividing Database into a static and dynamic database and implementing Caching of database.