

**PSIT KANPUR  
DEPARTMENT  
OF  
COMPUTER SCIENCE & ENGINEERING**



**CERTIFICATE**

**PSIT KANPUR...DEPARTMENT OF COMPUTER SCIENCE  
ENGINEERING...This is to certify that the course  
based project entitled “RAILWAY RESERVATION  
SYSTEM” is a bonafide work done by MISS ANUBHA in  
partial fulfilment of the requirement for the award  
of degree in “BACHELOR OF TECHNOLOGY in Computer  
Science Engineering” during the academic year  
2021-2022. Faculty In Charge Head of the Department**

“”

**DEPARTMENT  
OF  
COMPUTER SCIENCE & ENGINEERING**



**DECLARATION**

We hereby declare that this project based lab report entitled "**RAILWAY RESERVATION SYSTEM**" has been prepared by us in partial fulfillment of the requirement for the award of degree "**BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE ENGINEERING**" during the academic year **2021-2022**. We also declare that this project based lab report is of our own effort and it has not been submitted to any other university for the award of any degree. Date:**OCTOBER 2021** Place: **PSIT COLLEGE OF ENGINEERING KANPUR..Name-MISS ANUBHA... Student ID-**

# **ACKNOWLEDGMENT**

**It is great pleasure for me to express my gratitude to our honorable CHAIRMAN MR.PRANVEER SINGH, for giving the opportunity and platform with facilities in accomplishing the project based laboratory report.**

**I express the sincere gratitude to our principal-.... for his administration towards our academic growth.**

**I express sincere gratitude to our Coordinator-..... for his leadership and constant motivation provided in successful completion of our academic semester.**

**I record it as my privilege to deeply thank our pioneer-....., HOD CSE Dept., for providing us the efficient faculty and facilities to make our ideas into reality.**

**I express my sincere thanks to our project supervisor-.....for his novel association of ideas, encouragement, appreciation and intellectual zeal which motivated us to venture this project successfully.**

**Finally, it is pleased to acknowledge the indebtedness to all those who devoted themselves directly or indirectly to make this project report success.**

**PROJECT ASSOCIATES  
NAME || STUDENT ID**



# **TABLE OF CONTENTS**

---

<b>ACKNOWLEDGMENTS</b>	<b>4</b>
<b>Abstract</b>	<b>6</b>
<b>Introduction</b>	<b>7</b>
<b>PROJECT DESCRIPTION</b>	<b>8</b>
<b>List of Entities &amp; Attributes</b>	<b>10</b>
<b>ER Diagram (Conceptual Model)</b>	<b>12</b>
<b>Schema Diagram</b>	<b>13</b>
<b>Normalization &amp; Final List of Relations</b>	<b>14</b>
<b>Create &amp; Insert SQL Queries</b>	<b>16</b>
<b>SQL Queries related to Report Generation</b>	<b>19</b>
<b>Conclusion</b>	<b>24</b>

# **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 database 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.**

**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 data base which includes customer details, availability of seats in trains, no of trains and their details.**

# **PROJECT 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 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 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 has to provide the desired train number and the date for which ticket is to be booked. Before booking a 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 has to 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 working of the reservation system. To implement this sample case study, some assumptions have been made, which are as follows:**

- 1. The number of trains has been restricted to 5.**
- 2. The booking is open only for next seven days from the current date.**
- 3. Only two categories of tickets can be booked, namely, AC and General.**
- 4. The total number of tickets that can be booked in each category (AC and General) is 10.**
- 5. The total number of tickets that can be given the status of waiting is 2.**
- 6. The in- between stoppage stations and their bookings are not considered.**

**List of trains has to be maintained. Detailed Passenger information is to be maintained In the booking procedure, the train number, train date, and category are read from the passenger. On the basis of the values provided by the passenger, corresponding record is retrieved from the Train\_Status. If the desired category is AC, then total number of AC seats and number of booked AC seats are compared in order to find whether ticket can be booked or not. Similarly, it can be checked for the general category. If ticket can be booked, then passenger details are read and stored in the Passenger table. In the cancellation procedure, ticket ID is read from the passenger and corresponding record is searched in the Passenger. If the record exists, it is deleted. After deleting the record (if it is confirmed), first record with waiting status for the same train and same category are searched from the Passenger table and its status is changed to confirm.**

# **LIST OF ENTITIES & ATTRIBUTES**

## **ENTITIES**

### **User**

## **ATTRIBUTES**

**User\_Id**

**Password**

**First\_name**

**Last\_name**

**Gender**

**Age**

**Email**

**Aadhar\_no**

**Mobile\_no**

**City**

**State**

**Pincode**

**Security\_ques**

**Security\_ans**

### **Passenger**

### **Passenger\_Id**

**Name**

**Gender**

**Age**

**Pnr\_no**

**Seat\_no**

**Booked\_by**

**Reservation\_status**

# **TRAIN**

---

# **TRAIN NO.**

Train\_name  
Source  
Destination  
Arrival\_time  
Departure\_time  
Availability\_of\_seats  
Train\_no  
A\_seats1 A\_  
seats2 A\_  
seats3 B\_  
seats1 B\_  
seats2 B\_  
seats3 W\_  
Seats1 W\_  
seats2 W\_

## **Station**

## **Name**

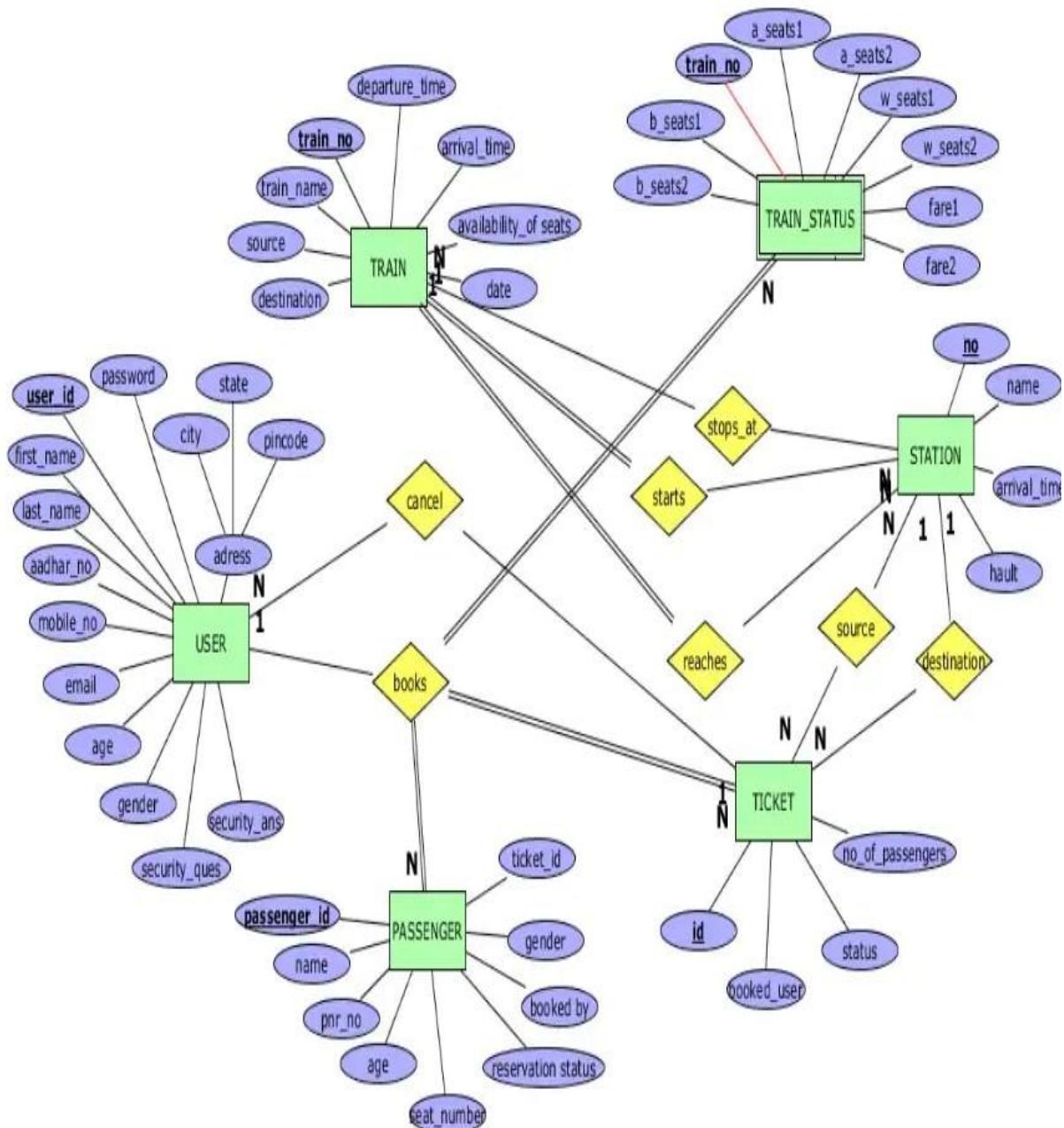
No  
Train\_no  
Arrival\_time  
Hault

## **Ticket**

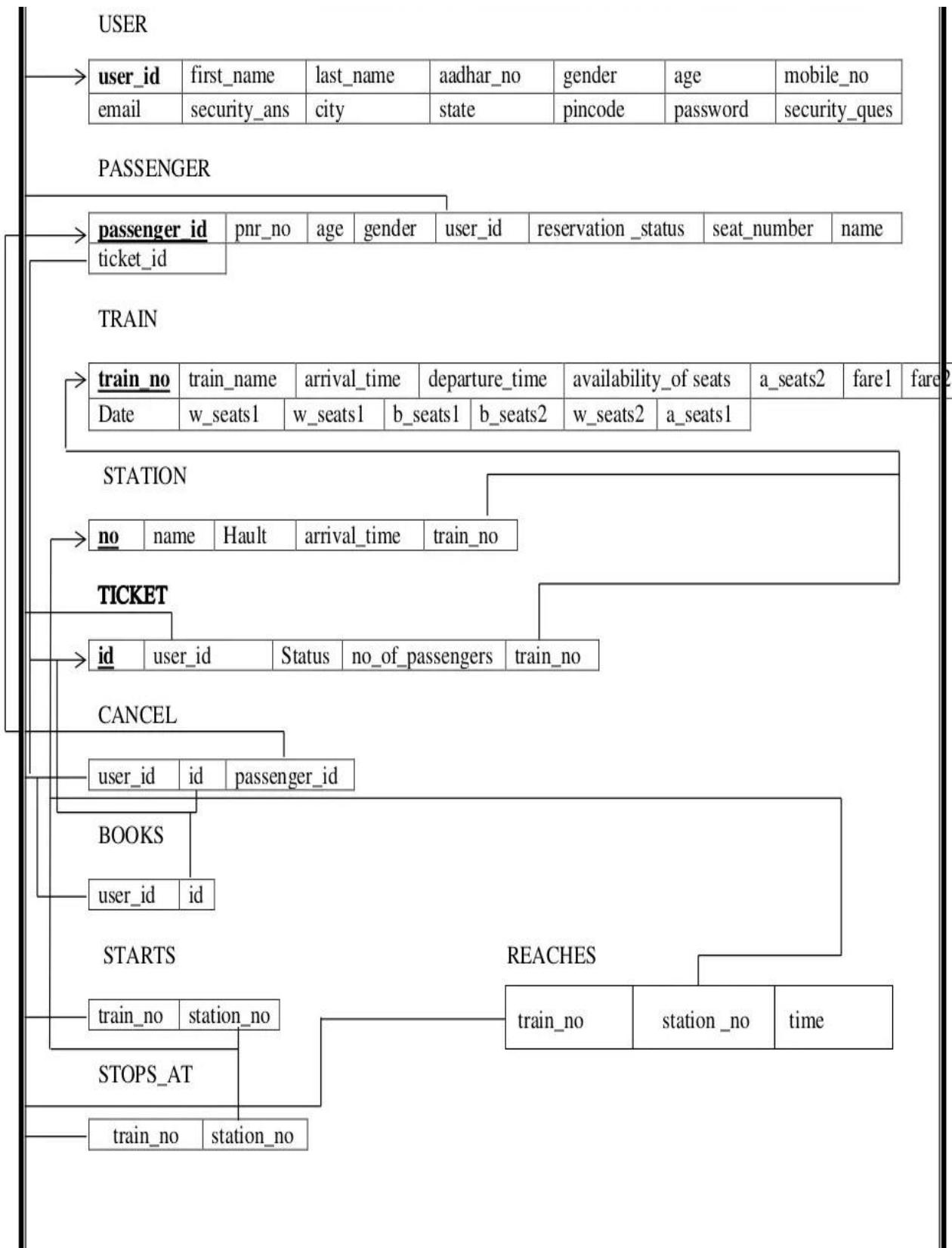
## **ID**

Train\_no  
Booked\_user  
Status  
No\_of\_passengers

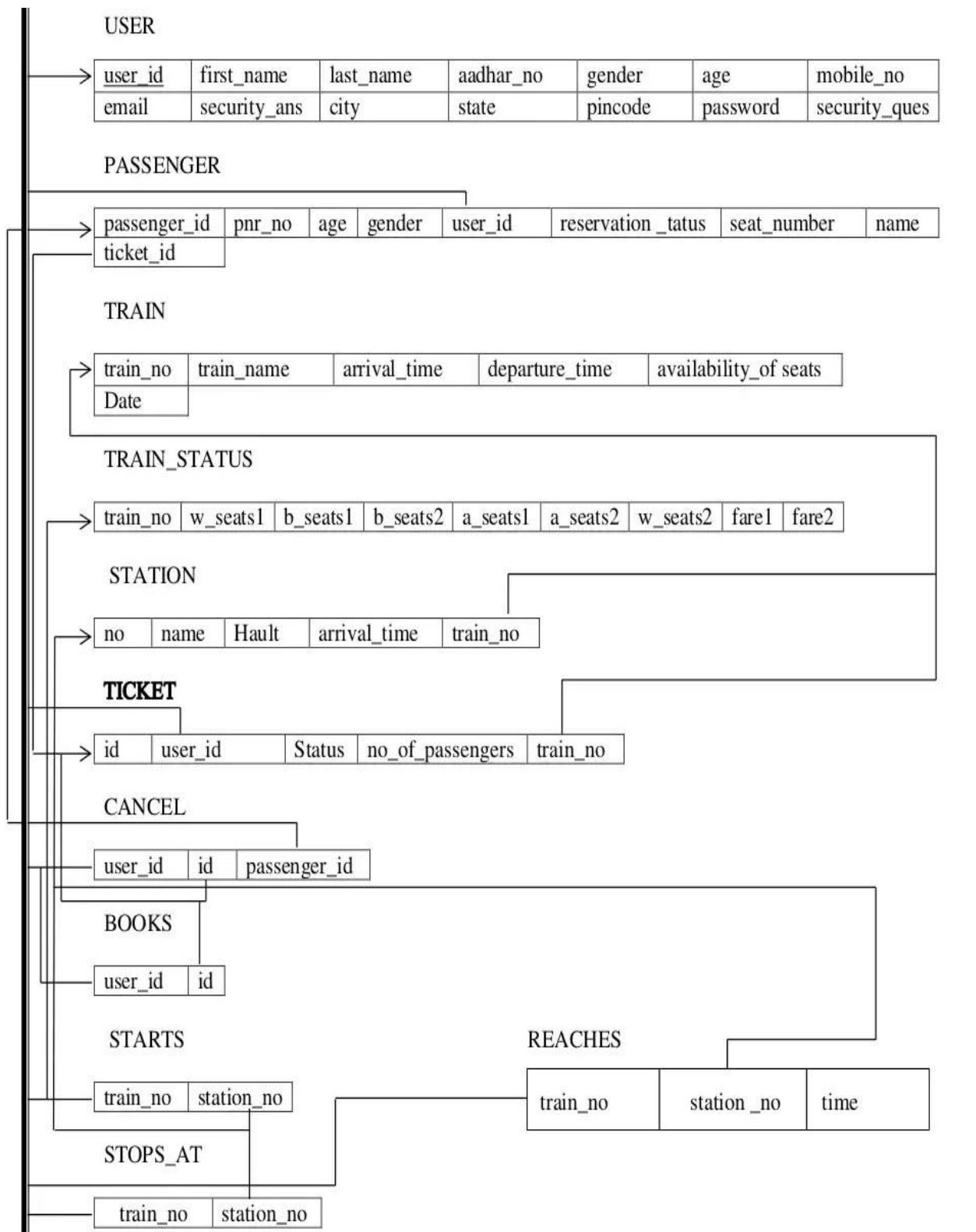
# ER MODEL (CONCEPTUAL MODEL)



# SCHEMA DIAGRAM



# NORMALIZATION & FINAL LIST OF RELATIONS.



## **FINAL LIST OF RELATIONSHIP**

- books -Ternary relationship between **USER,TRAIN,PASSENGER** and **TICKET**.
- starts –Between **TRAIN** and **STATION**
- reaches –Between **TRAIN** and **STATION**
- cancel –Between **USER** and **TICKET**
- stops\_at –Between **TRAIN** and **STATION**

# CREATE & INSERT SQL QUERIES

## CREATE COMMANDS:

```
create table If not exists USER(user_Id int primary key,first_name  
varchar(50),last_name varchar(50),adhar_no varchar(20),gender char,age  
int,mobile_no varchar(50),email varchar(50),city varchar(50),state  
varchar(50),pincode varchar(20),_password varchar(50),security_ques  
varchar(50),security_ans varchar(50));
```

```
create table If not exists TRAIN(train_no int primary key,train_name  
varchar(50),arrival_time time,departure_time time,availability_of_seats  
char,date date);
```

```
create table If not exists STATION(no int ,name varchar(50),hault int,arrival_time  
time,train_no int,primary key(station_no,train_no),constraint foreign  
key(train_no) references TRAIN(train_no));
```

```
create table If not exists TRAIN_STATUS(train_no int primary key,b_seats1  
int,b_seats2 int,a_seats1 int,a_seats2 int,w_seats1 int,w_seats2 int,fare1  
float,fare2 float);
```

```
create table If not exists TICKET(id int primary key,user_Id int,status  
char,no_of_passengers int,train_no int,constraint foreign key(user_Id)  
references USER(user_Id),constraint foreign key(train_no) references  
TRAIN(train_no));
```

```
create table If not exists PASSENGER(passenger_Id int primary key,pnr_no int,age  
int,gender char,user_Id int,reservation_status char,seat_number  
varchar(5),name varchar(50),ticket_Id int,constraint foreign key(user_Id)  
references USER(user_Id),constraint foreign key(ticket_Id) references  
TICKET(id));
```

```
create table if not exists STARTS( train_no int primary key,station_no int,constraint foreign key(train_no) references TRAIN(train_no),constraint foreign key(station_no) references STATION(no));
```

```
create table if not exists STOPS_AT( train_no int,station_no int,constraint foreign key(train_no) references TRAIN(train_no),constraint foreign key(station_no) references STATION(no));
```

```
create table if not exists REACHES(train_no int,station_no int,time time,constraint foreign key(train_no) references TRAIN(train_no),constraint foreign key(station_no) references STATION(no));
```

```
create table if not exists BOOKS( user_id int,id int,constraint foreign key(user_id) references USER(user_id),constraint foreign key(id) references TICKET(id));
```

```
create table if not exists CANCEL(user_id int,id int,passenger_id int,constraint foreign key(id) references TICKET(id),constraint foreign key(passenger_id) references PASSENGER(passenger_id),constraint foreign key(user_id) references USER(user_id));
```

## **INSERT QUERIES:**

```
insert into
USER(user_id,first_name,last_name,aadhar_no,gender,age,mobile_no,email,city,state,pincode,_password,security_ques,security_ans)
values(1701,'vijay','sharma','309887340843','M',34,'9887786655','vijay1@gmail.com','vijayawada','andhrapradesh','520001','12345@#','favouritecolour','red'),(1702,'rohit','kumar','456709871234','M',45,'9809666555','rohit1kumar@gmail.com','gujarat','andhrapradesh','522004','12@#345','favouritebike','bmw'),(1703,'manasvi','sre
```

e','765843210987','F',20,'9995550666','manasvi57@gmail.com','guntur','andhra pradesh','522004','0987hill','favourite flower','rose');

Insert Into  
TRAIN(train\_no,train\_name,arrival\_time,departure\_time,availability\_of seats,date) values(12711,'plnaklml exp','113000','114000','A',20170410),(12315,'cormandel exp','124500',125000,'NA',20170410);

Insert Into STATION(no,name,hault,arrival\_time,train\_no)  
values(111,'vijayawada',10,'113000',12711),(222,'tirupathi',5,'114500',12315);

Insert Into  
TRAIN\_STATUS(train\_no,w\_seats1,b\_seats1,b\_seats2,a\_seats1,a\_seats2,w\_seats 2,fare1,fare2) values(12711,10,4,0,1,1,0,100,450),(12315,10,5,0,0,2,1,300,600);

Insert Into TICKET(id,user\_Id,status,no\_of\_passengers,train\_no)  
values(4001,1701,'C',1,12711),(4002,1702,'NC',1,12315);

Insert Into PASSENGERS(passenger\_Id,pnr\_no,age,gender,user\_Id,reservation \_status,seat\_number,name,ticket\_Id) values(5001,78965,45,'M',1701,'C','B6-45','ramesh',4001),(5002,54523,54,'F',1701,'W','B3-21','surekha',4002);

Insert Into STARTS(train\_no,station\_no) values(12711,111),(12315,222);

Insert Into STOPS\_AT(train\_no,station\_no) values(12711,222),(12315,111);

Insert Into REACHES(train\_no,station\_no,time) values(12711,222,'040000'),  
(12315,111,'053500');

Insert Into BOOKS(user\_Id,id) values(1701,4001),(1702,4002);

Insert Into CANCEL(user\_Id,id,passenger\_Id) values(1701,4001,5001);

# SQL QUERIES RELATED TO REPORT GENERATION...

1.print user id and name of all those user who booked ticket for pinakini express

```
select u.user_id,concat(u.first_name,u.last_name)as name  
from user u,train t,ticket tc  
where u.user_id=tc.user_id and t.train_no=tc.train_no and t.train_name like 'pinakini  
exp';
```

```
mysql> select u.user_id,concat(u.first_name,u.last_name)as name  
-> from user u,train t,ticket tc  
-> where u.user_id=tc.user_id and t.train_no=tc.train_no and t.train_name  
-> like'pinakini exp';  
+-----+-----+  
| user_id | name |  
+-----+-----+  
| 1701 | vijaysharma |  
| 1701 | vijaysharma |  
+-----+-----+  
2 rows in set (0.00 sec)
```

2. print details of passengers travelling under ticket no 4001

```
select *  
from passenger  
  
where ticket_id like 4001;
```

```
mysql> select *  
-> from passenger  
-> where ticket_id like 4001;  
+-----+-----+-----+-----+-----+-----+-----+-----+  
| passenger_id | por_no | age | gender | user_id | reservation_status | seat_number | name | ticket_id |  
+-----+-----+-----+-----+-----+-----+-----+-----+  
| 5001 | 78965 | 45 | M | 1701 | C | B6-45 | ranesh | 4001 |  
+-----+-----+-----+-----+-----+-----+-----+-----+  
1 row in set (0.00 sec)
```

**3. display all those train no's which reach station no —**

**select t.\***

**from train t.station s.reaches r**

**where t.train\_no=r.train\_no and r.station\_no=s.no and s.name like 'Vijayawada';**

```
mysql> select t.*  
-> from train t,station s,reaches r  
-> where t.train_no=r.train_no and r.station_no=s.no and s.name like 'vijayawada';  
+-----+-----+-----+-----+-----+  
| train_no | train_name | arrival_time | departure_time | availability_of_seats | d |  
+-----+-----+-----+-----+-----+  
| 12315 | comandrel exp | 12:45:00 | 12:50:00 | N | 2 |  
| 12255 | shatabdhi exp | 13:55:00 | 14:00:00 | N | 2 |  
+-----+-----+-----+-----+-----+  
2 rows in set (0.00 sec)
```

4. display time at which train no— reaches station no —

**select r.\* s.name**

**from reaches r.station s**

**where r.station\_no=s.no;**

-> from read

```
+-----+-----+-----+
| train_no | station_no | time
+-----+-----+-----+
| 12711   |      222 | 04:00
| 12315   |      111 | 05:35
| 12255   |      111 | 06:00
+-----+-----+-----+
3 rows in set (0.00 sec)
```

**5. display details of all those users who canceled tickets for train no—**

**select u.\***

**from user u.cancel c.ticket t**

**where c.user\_id=u.user\_id and c.id=t.id and t.train\_no like 12711;**

## **6. display the train no with Increasing order of the fares of class 1**

**select ts.train no,ts.fare1,t.train name**

**from train\_status ts,train t**

**where t.train\_no=ts.train\_no**

**order by fare1 asc;**

```
mysql> select ts.train_no,ts.fare1,t.train_name
    -> from train_status ts,train t
    -> where t.train_no=ts.train_no
    -> order by fare1 asc;
+-----+-----+-----+
| train_no | fare1 | train_name |
+-----+-----+-----+
| 12711   | 100  | pinakini exp |
| 12315   | 300  | cormandel exp |
| 12255   | 400  | shatabdhi exp |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

## **7. display passenger details for train plnaknl.**

**select p.\***

from passenger p.train t.ticket tc

**where tc.train\_no=t.train\_no and tc.tid=p.ticket\_id and t.train\_name like**

'plnaklnl exp'

## 8. display immediate train from tirupathi to Vijayawada

**select distinct t.\***

**from train t,station s,starts st,stops\_at sa**

**where st.station\_no=(select no from station where name like 'tirupathi')**

**and sa.station\_no=(select no from station where name like 'vijayawada')**

**order by date;**

```
mysql> select distinct t.*  
-> from train t,station s,starts st,stops_at sa  
-> where st.station_no=(select no from station where name like 'tirupathi')  
-> and sa.station_no=(select no from station where name like 'vijayawada')  
-> order by date;  
+-----+-----+-----+-----+-----+-----+  
| train_no | train_name | arrival_time | departure_time | availability_of_seats | date |  
+-----+-----+-----+-----+-----+-----+  
| 12315 | cormandel exp | 12:45:00 | 12:50:00 | N | 2017-04-10 |  
| 12711 | pinakini exp | 11:30:00 | 11:40:00 | A | 2017-04-10 |  
| 12255 | shatabdhi exp | 13:55:00 | 14:00:00 | N | 2017-04-11 |  
+-----+-----+-----+-----+-----+-----+  
3 rows in set (0.01 sec)
```

## 9. display the train no which haults for more time in station no —————

**select train\_no**

**from station**

**having max(hault);**

```
mysql> select train_no  
-> from station  
-> having max(hault);  
+-----+  
| train_no |  
+-----+  
| 12711 |  
+-----+  
1 row in set (0.00 sec)
```

**10. display details of all those passengers whose status is confirmed for train no— —**

**select t.\***

**from ticket t**

**where t.status like 'c' and t.train\_no=12711;**

```
mysql> select t.*  
-> from ticket t  
-> where t.status like 'c' and t.train_no=12711;  
+----+----+----+----+----+  
| id | user_id | status | no_of_passengers | train_no |  
+----+----+----+----+----+  
| 4001 | 1701 | C | | 1 | 12711 |  
| 4003 | 1701 | C | | 1 | 12711 |  
+----+----+----+----+----+  
2 rows in set (0.00 sec)
```

# **MODULAR DESCRIPTION**

- **RESERVATION FORM :**

This form is used for the reservation of ticket. The main advantage of the form is that it has the easiest of the user interface. This makes it user friendly and easy to use. It has Passenger's name, address, age , date of ticket booking, source and destination station name and codes.

- **CANCELLATION FORM :**

This form consists of cancelation. the user interface is again easy. One just needs the PNR number to cancel a ticket.

- **FARE RECORDS :**

This form is used for the fare between two station with a specific class.

- **TRAIN ENQUIRY**

Train enquiry is used for knowing available trains between two stations. We need to provide source and destination names or codes and then we will be given the train names between those two stations.

- **RESERVATION ENQUIRY**

This form is used to know if there is any seat available in a train. Here we need to provide date, train no, source and destination stations.

- SEAT DETAILS This form is used to know about the seat later using our PNR no.

## Working Of Present System

### ■ Data redundancy:

It means that same data fields appear in many different files and often in different formats. In manual system, it poses quite a big problem because the data has to be maintained in large volumes but in our system, this problem can be overcome by providing the condition that if the data entered is duplicate, it will not be entered, otherwise, updating will take place.

### ■ Difficulty in accessing the data:

In manual system, searching information is time consuming but in our system, any information can be accessed by providing the primary key.

### ■ Unsatisfactory security measures

In manual system, no security measures were provided but in this system, password security has been provided. The person can access the system by providing the correct password otherwise he is denied the access

# Data Flow Diagram

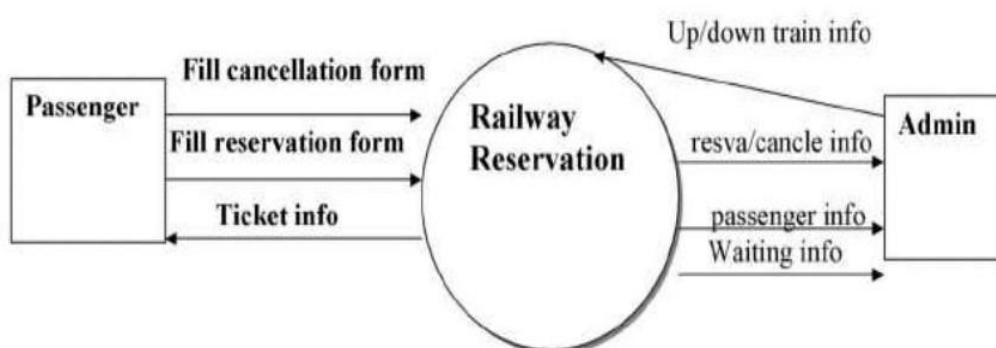
## Definition of DFD:

The data flow diagram is a graphical representation that depicts information flow and the transforms that are applied as data moves from input to output. The DFD may be used to represent a system or software at any level of abstraction. In fact DFD may be partitioned into levels that represent increasing information flow and functional detail.

## **LEVEL 0 DFD OR CONTEXT FREE DIAGRAM:**

The level 0 DFD or a context model represents the entire software element as a single bubble with input and output data indicated by incoming and outgoing arrows, respectively.

In level 0 diagram shown below, the passenger fills either the reservation or cancellation form as input. He gets the ticket as the output and the report is sent to the administration.



## **LEVEL 2 DFD:**

A level 1 DFD is the further refinement of level 0 DFD showing greater details and functionalities. In this, the single bubble of level 0 DFD is refined further . Each of the processes depicted at level 1 is a subfunction of the overall system depicted in the context model.

As shown in the DFD above, the passenger either enquires about the trains or goes directly for the reservation or the cancellation processes as a result of which he gets the ticket generated. The reports are then sent to the administration

# **DATA TABLE**

## **Login Table:-**

S.NO.	Field name	Data type	Description
1	User Name	Text	Store user name for checking correct username
2	Password	Text/Number	Store password corresponding to user name

## **RAILWAY RESERVATION RECORD TABLE:-**

Sr.no.	Field name	Data type
1	Class	Text
2	Name	Text
3	Age	Number
4	Gender	Text
5	Date	Number

## **CANCELLATION RECORD TABLE:-**

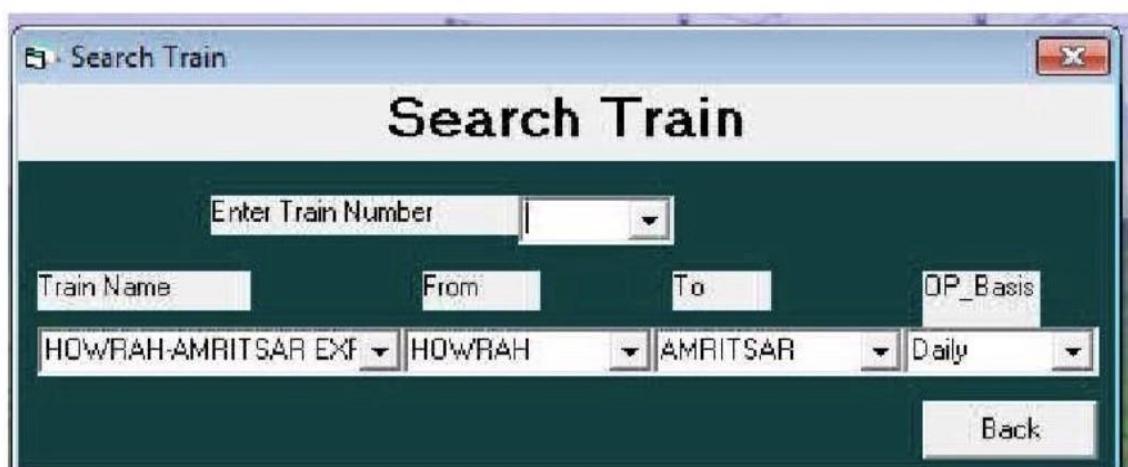
Sr.No.	Field name	Data type
1	P.N.R. Number	Number

# SCREEN SHORT'S OF FORM

Login Form



Train Search Form



## Ticket Reservation Form

The screenshot shows a Windows application window titled "Reservation". The main title bar says "RESERVATION". Inside, there are fields for "Train Number" (5), "Train Name" (VARANASI-LUCKNOW EXPRESS), "From" (VARANASI), "To" (LUCKNOW), "Date" (16/07/2015), and "class" (1 class). Below these are six rows for passenger information, each with "Sr No.", "Name of Passenger", "Age", "Sex" (dropdown menus), and "Senior Citizen" (checkbox). The first two rows are populated: row 1 has "KIRAN SHARMA" and "F", and row 2 has "PINTU SAHU" and "M". To the right of the passenger table is a vertical toolbar with buttons for "Book Ticket", "Clear", and "Cancel". A note at the bottom states: "Note: Tickets Once Reserved cannot be exchanged or edited." There is also a small image of a train.

## Ticket Cancel Form

The screenshot shows a Windows application window titled "Cancellation". The header includes "HAPPY JOURNEY" and "INDIAN RAILWAY". It has fields for "Train Number" (4), "Date", "PNR No" (12345683), "Train Name", "class", "From", "To", and "Adult", "Child", "SC" checkboxes. Below this is a large orange section containing "CANCELLED", "THIS TICKET HAS BEEN CANCELLED", and "Total Amount Incurred(Cancellation Fee) Rs:160". At the bottom is a "Print" button and a decorative graphic of a train engine with radiating lines.

## Reservation Record List Form

Form8

PNR NO	Train No	Train Name	From	To	Date Travel	Class	Passenger Name	Age	Sex
12345678	1	HOWRAH-AMRITSAR	HOWRAH	AMRITSAR	26/12/2009	II Sleeper	Sukanya	17	F
12345679	4	HOWRAH-JAMMUTAWA HOWRAH		JAMMUTAWA	26/12/2009	II sleeper	Ananya	12	F
12345680	1	HOWRAH-AMRITSAR	HOWRAH	AMRITSAR	26/12/2009	II sleeper	Mr Sahari	51	M
12345681	2	MALDA BIHWANI EXP(MALDA)	BHWINI	BHWINI	11/08/2015	II class	vijay sahu	15	M
12345682	7	PATNA-NEW DELHI EXP(PATNA)	NEW DELHI	NEW DELHI	16/10/2015	I class	amit dhabi	21	M
12345681	2	MALDA BIHWANI EXP(MALDA)	BHWINI	BHWINI	11/08/2015	II class	sunil sahu	23	M
12345682	7	PATNA-NEW DELHI EXP(PATNA)	NEW DELHI	NEW DELHI	16/10/2015	I class	abhishek khwai	20	M
12345680	1	HOWRAH-AMRITSAR	HOWRAH	AMRITSAR	26/12/2009	II sleeper	Mrs. Sahari	43	F
12345681	2	MALDA BIHWANI EXP(MALDA)	BHWINI	BHWINI	11/08/2015	II class	avniika sahu	21	F
12345682	7	PATNA-NEW DELHI EXP(PATNA)	NEW DELHI	NEW DELHI	16/10/2015	I class	sachin patmar	21	M

Back

## Fare Detail Form

Fare Details

Enter Train Number	6	<input type="button" value="Get Fares"/>
Enter Class	II class	
Train Number	6	
Train Name	BOMBAY-FAIZABAD SAKET EXPRESS	
Adult	330	Child 290
II class	330	290
Back		

# CONCLUSION

In our project Railway reservation system we have stored all the information about the Trains scheduled and the users booking tickets and even status of trains, seats etc. This data base is helpful for the applications which facilitate passengers to book the train tickets and check the details of trains and their status from their place itself it avoids inconveniences of going to railway station for each and every query they get. We had considered the most important requirements only, many more features and details can be added to our project inorder to obtain even more user friendly applications. These applications are already in progress and in future they can be upgraded and may become part of amazing technology.



**Thank You!**

**DEPARTMENT  
OF  
COMPUTER SCIENCE & ENGINEERING  
PSIT COLLEGE OF ENGINEERING  
KANPUR**

**SUBMITTED BY:**

**PSIT KANPUR**

**PSIT**  
*Kanpur*

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**B.TECH-2ND YEAR [THIRD SEMESTER]**  
**ACADEMIC YEAR 2021-2022**

**MINI PROJECT BASED  
ON  
RAILWAY RESERVATION SYSTEM**

**SUBMITTED BY  
SECTION-2A**

**STUDENT ID**

**STUDENT NAME**

**DEPARTMENT**