

# **[Railway Reservation Management]**

## **PROJECT Report**

*Submitted by*

**Vamsi krishan(RA1911003010950)**

**Lakshman (RA1911003010965)**

**Charan Teja (RA1911003010966)**

**Satish (RA1911003010968)**

*partial fulfillment for the award of the degree of*  
**B.TECH**

*in*

**COMPUTER SCIENCE and ENGINEERING**

**IN**

**CLOUD COMPUTING**

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**KATTANKULATHUR**

**[April, 2022]**

## **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 the Railways reservation system in .It is the computerized system of reserving the seats of train seats in advance. It is mainly used for long routes. 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 relation model. There is also design of the database of the railway reservation system based on relation model.

## TABLE OF CONTENT

<b>CHAPTER NO</b>	<b>TITLE</b>	<b>PAGE NO</b>
<b>1</b>	<b>ABSTRACT</b>	<b>2</b>
<b>2</b>	<b>INTRODUCTION</b>	<b>4</b>
<b>3</b>	<b>LIST OF RealationShips</b>	<b>4</b>
<b>4</b>	<b>Entity Relationship Diagram</b>	<b>5</b>
<b>5</b>	<b>Normalized Database Table</b>	<b>6</b>
<b>6</b>	<b>SQL Queries with results</b>	<b>12</b>
<b>7</b>	<b>Conclusion and Future Work</b>	<b>13</b>
<b>8</b>	<b>References</b>	<b>13</b>

## **INTRODUCTION:-**

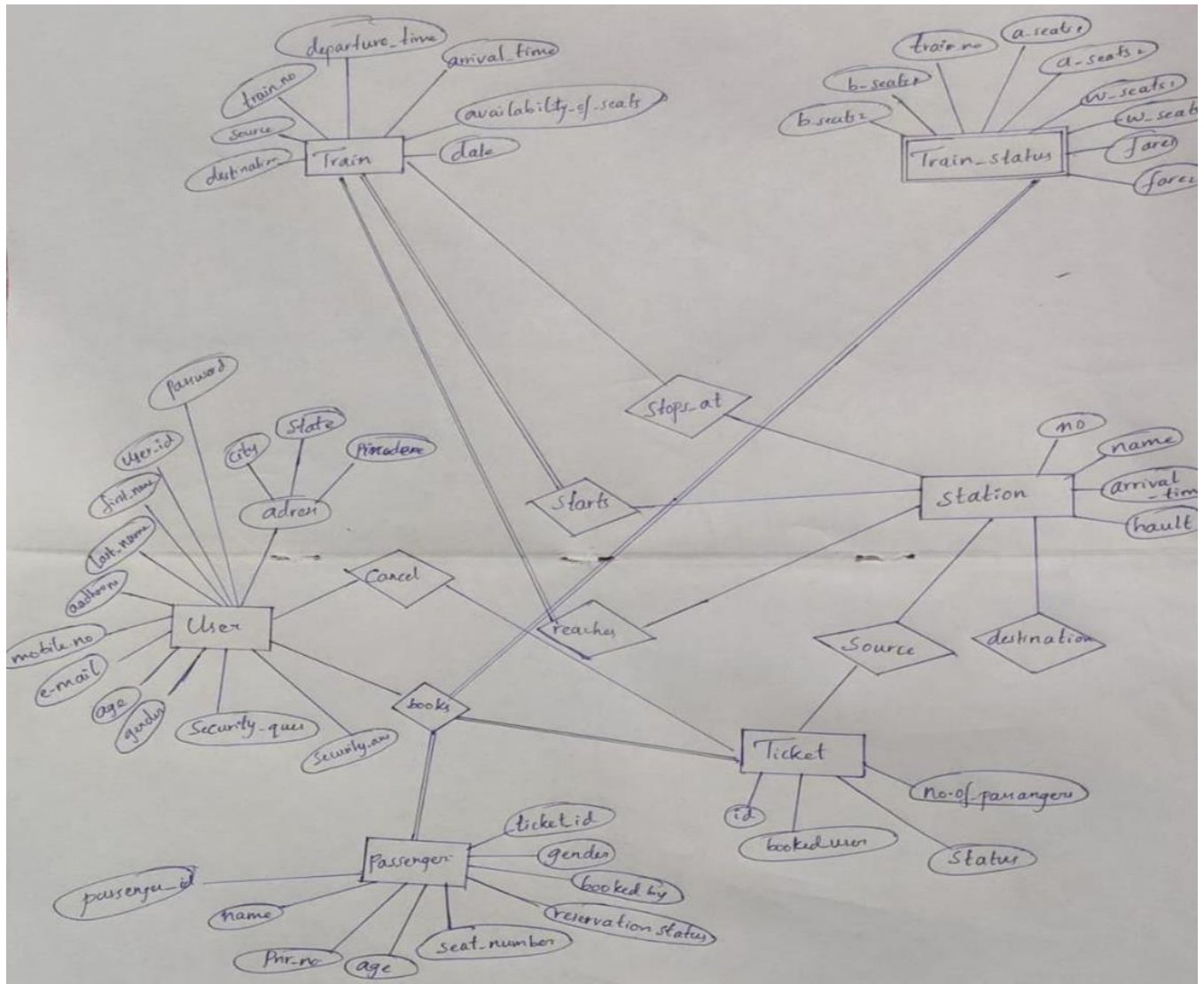
Database is an organized collection of data. The data is typically organized to model aspects of reality in a way that supports processes requiring information. A DBMS makes it possible for end users to create, read, update and delete data in a database. The DBMS essentially serves as an interface between 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.

## **LIST OF RELATION SHIPS:**

- 1)books -Ternary relation ship between USER, TRAIN,PASSENGER and TICKET.
- 2)starts -Between TRAIN and STATION
- 3)reaches -Between TRAIN and STATION
- 4)cancel- Between USER and TICKET
- 5)stops at- Between TRAIN and STATION

## Entity Relationship Diagram:-



## Normalized Database Table:-

### USER

<u>user_id</u>	first_name	last_name	aadhar_no	gender	age	mobile_no
email	security_ans	city	state	pincode	password	security_ques

### PASSENGER

<u>passenger_id</u>	pnr_no	age	gender	user_id	reservation_tatus	seat_number	name
ticket_id							

### TRAIN

<u>train_no</u>	train_name	arrival_time	departure_time	availability_of seats
Date				

### TRAIN\_STATUS

<u>train_no</u>	w_seats1	b_seats1	b_seats2	a_seats1	a_seats2	w_seats2	fare1	fare2
-----------------	----------	----------	----------	----------	----------	----------	-------	-------

### STATION

<u>no</u>	name	Hault	arrival_time	train_no
-----------	------	-------	--------------	----------

### TICKET

<u>id</u>	user_id	Status	no_of_passengers	train_no
-----------	---------	--------	------------------	----------

### CANCEL

<u>user_id</u>	<u>id</u>	passenger_id
----------------	-----------	--------------

### BOOKS

<u>user_id</u>	<u>id</u>
----------------	-----------

### STARTS

<u>train_no</u>	<u>station_no</u>
-----------------	-------------------

### STOPS\_AT

<u>train_no</u>	<u>station_no</u>
-----------------	-------------------

### REACHES

<u>train_no</u>	<u>station_no</u>	<u>time</u>
-----------------	-------------------	-------------

## SQL Queries:-

```
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 (1),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(station_no INT, namee VARCHAR(50),halt 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,statuss
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));
`train`
```

```
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),namee VARCHAR(50),ticket_id
INT,CONSTRAINT (fk_userid) FOREIGN KEY(user_id) REFERENCES USER user_id,CONSTRAINT
FOREIGN KEY(ticket_id)
REFERENCES TICKET(id));
```

```
CREATE TABLE IF NOT EXISTS START( train_no INT PRIMARY KEY,station_no INT,CONSTRAINT
FOREIGN
KEY(train_no) REFERENCES TRAIN(train_no),CONSTRAINT FOREIGN KEY(station_no) REFERENCES
STATION(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(station_no));
```

```
CREATE TABLE IF NOT EXISTS REACHES(train_no INT,station_no INT,timee TIME,CONSTRAINT
FOREIGN
KEY(train_no) REFERENCES TRAIN(train_no),CONSTRAINT FOREIGN KEY(station_no) REFERENCES
STATION(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 INTO

USER VALUES

(1701,'Vijay','Sharma','309887340843','M',34,'9887786655','vijay1@gmail.com','vijayawada','andhrapradesh','520001','12345@#','favouritecolour','red');

INSERT INTO

USER VALUES

(1702,'Rohith','Kumar','456709871234','M',45,'9809666555','rohith1kumar@gmail.com','guntur','andhrapradesh','522004','12@#345','favouritebike','bmw');

INSERT INTO

USER VALUES

(1703,'Mananasvi','','765S43210987','F',20,'999555Ofifi6','mananasvi57@gmail.com','guntur','andhra pradesh','522004','O987hii','favouriteflower','rose');

INSERT INTO TRAIN VALUES (12711,'pinakini exp','113000','114000','A',20170410);

INSERT INTO TRAIN VALUES (12315,'corimandel exp','124500','125000','N',20170410 );

INSERT INTO STATION (station\_no,namee,hault,arrival\_time,train\_no) VALUES

(111,'vijayawada',10,'11:30:00',12711);

INSERT INTO

station VALUES (222,'tirupathi',5,'11:45:00',12315);

INSERT INTO PASSENGER (passenger\_id,pnr\_no,age,gender,user\_id,reservation\_status, seat\_number,NAME,ticket\_id) VALUES ( 5001,78965,45,'M',1701,'C' ,'B6-45','rainesh',4001),(5002,54523,54,'F',1701,'W','B3-21','surekha',4002);

INSERT INTO STARTS(train NO,station NO) VALUES(12315,222);

INSERT INTO STOPS\_AT(train NO,station NO) VALUES(12711,222),(12315,111);

INSERT INTO REACHES(train NO,station NO,TIMEe) VALUES(12711,222,'12O000'), (12315,111,'O53500');

INSERT INTO BOOKS(user\_id,id) VALUES(1701,77),(1702,4002);

INSERT INTO CANCEL(USER id,id,passenger id) VALUES(1701,4001,5001);



## SQL Queries with results

1. Print user id and name of all those user who booked ticket for corimandel 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
'corimandel exp';
```

Scheduled Backups presents a wizard driven interface to take backups : Reason #43 to upgrade

el passenger reaches starts station stops\_at train ticket train\_status user Railway Rese...t(98,99,10) + ▾

```
1 SELECT u.user_id, CONCAT(u.first_name, u.last_name) AS NAME
2 FROM USER u, train t, ticket tc
3 WHERE u.user_id=tc.user_id AND t.train_no=tc.train_no AND t.train_name LIKE 'corimandel exp';
4
```

1 Result 2 Profiler 3 Messages 4 Table Data 5 Info

(Read Only) Limit rows First row 0 # of rows 1000

user_id	Name
1701	vijaysharma

2. Print details of passengers travelling under ticket no 77?

```
SELECT * FROM passenger WHERE ticket_id = 77;
```

```
5
6 SELECT * FROM passenger WHERE ticket_id = 77;
7
```

1 Result 2 Profiler 3 Messages 4 Table Data 5 Info

(Read Only) Limit rows First row 0 # of rows 1000

passenger_id	pnr_no	age	gender	user_id	reservation_status	seat_number	name	ticket_id
5001	78965	45	M	1701	C	B6-45	dinesh	77

3. Display all those train no's which reach station no 111?

```
SELECT t.*
FROM train t, station s, reaches r
WHERE t.train_no=r.train_no AND r.station_no=s.station_no AND s.name LIKE
'vijayawada';
```

```

8
9 SELECT t.*
10 FROM train t,station s,reaches r
11 WHERE t.train_no=r.train_no AND r.station_no=s.station_no AND s.name LIKE 'vijayawada';
12

```

train_no	train_name	arrival_time	departure_time	availability_of_seats	date
12315	corimandel exp	12:45:00	12:50:00	N	2017-04-10

4. Display time at which train no 12315 reaches station no 111?

```

SELECT r.*,s.name
FROM reaches r,station s WHERE r.station_no=s.station_no;

```

```

14 SELECT r.*,s.name
15 FROM reaches r,station s WHERE r.station_no=s.station_no;
16
17

```

train_no	station_no	time	name
12315	111	05:35:00	vijayawada
12711	222	12:00:00	tirupati

5. Display details of all those users who canceled tickets for train no 12315?

```

SELECT u.* FROM USER u,ticket t,cancel c
WHERE c.user_id=u.user_id AND c.id=t.id AND t.train_no LIKE 12315;

```

```

18 SELECT u.* FROM USER u,ticket t,cancel c
19 WHERE c.user_id=u.user_id AND c.id=t.id AND t.train_no LIKE 12315;
20

```

user_id	first_name	last_name	adhar_no	gender	age	mobile_no	email	city	state	pincode	password	security_que
1701	vijay	sharma	309887340843	M		34 9887786655	vijay1@gmail.com	vijayawada	andhrapradesh	520001	12345@#	favouritecol

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;



The screenshot shows a database query interface with a query editor at the top and a results pane below. The query is: `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;` The results pane shows two rows of data.

train_no	fare1	train_name
12711	100	pinakini exp
12315	300	corimandel exp

7. Display the train no with decreasing 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 DESC;



The screenshot shows a database query interface with a query editor at the top and a results pane below. The query is: `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 DESC;` The results pane shows two rows of data.

train_no	fare1	train_name
12315	300	corimandel exp
12711	100	pinakini exp

8. Display passenger details for train corimandel?

SELECT p.\*

FROM passenger p,train t,ticket tc

WHERE tc.train\_no=t.train\_no AND tc.id=p.ticket\_id AND t.train\_name LIKE 'corimandel exp';

el passenger reaches starts station stops\_at train ticket train\_status user Railway Rese...t(98,99,10)

```

30
31
32 SELECT p.*
33 FROM passenger p,train t,ticket tc
34 WHERE tc.train_no=t.train_no AND tc.id=p.ticket_id AND t.train_name LIKE 'corimandel exp';

```

1 Result 2 Profiler 3 Messages 4 Table Data 5 Info

(Read Only) Limit rows First row 0 # of rows 1000

passenger_id	pnr_no	age	gender	user_id	reservation_status	seat_number	name	ticket_id
5001	78965	45	M	1701	C	B6-45	dinesh	77

9. Display immediate train from tirupati to Vijayawada?

```

SELECT DISTINCT t.*
FROM train t,station s,STARTS st,stops_at sa
WHERE st.station_no=(SELECT station_no FROM station WHERE namee LIKE 'tirupati')
AND sa.station_no=(SELECT station_no FROM station WHERE namee LIKE 'vijayawada') ORDER BY datee;

```

el passenger reaches starts station stops\_at train ticket train\_status user Railway Rese...t(98,99,10)

```

36
37 SELECT DISTINCT t.*
38 FROM train t,station s,STARTS st,stops_at sa
39 WHERE st.station_no=(SELECT station_no FROM station WHERE namee LIKE 'tirupati')
40 AND sa.station_no=(SELECT station_no FROM station WHERE namee LIKE 'vijayawada') ORDER BY datee;

```

1 Result 2 Profiler 3 Messages 4 Table Data 5 Info

(Read Only) Limit rows First row 0 # of rows 1000

train_no	train_name	arrival_time	departure_time	availability_of_seats	datee
12315	corimandel exp	12:45:00	12:50:00	N	2017-04-10
12711	pinakini exp	11:30:00	11:40:00	A	2017-04-10

10. Display the train no which haults for more time in station no 111?

```

SELECT train_no FROM station HAVING MAX(hault);

```

42

```

43 SELECT train_no FROM station HAVING MAX(hault);

```

1 Result 2 Profiler 3 Messages 4 Table Data 5 Info

(Read Only) Limit rows First row 0 # of rows 1000

train_no
12711

11. Display details of all those passengers whose status is confirmed for train no 12315?



The screenshot shows a database management system interface with a query editor and a results pane. The query editor contains the following SQL query:

```
45  
46 SELECT t.*  
47 FROM ticket t  
48 WHERE t.status LIKE 'c' AND t.train_no=12315;  
49
```

The results pane displays a table with the following data:

id	user_id	status	no_of_passengers	train_no
77	1701	c	2	12315

### Conclusion and Future Work:-

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.

### References:-

<https://www.slideshare.net/shashankkarnati/railway-management-system-database-mini-project#:~:text=The%20Railway%20Reservation%20System%20facilitates,of%20the%20booked%20ticket%2C%20etc>