MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

Soft Skills Project Report

on

RAILWAY MANAGEMENT SYSTEM

Submitted By:

Romesh Singh Sengar

0901CA211050

Mentor:

Dr. Parul Saxena

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE

GWALIOR - 474005 (MP) est. 1957

July – December 2021

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

CERTIFICATE

This is certified that **Romesh Singh Sengar** (0901CA211050) has submitted the project report titled **Railway Management System** under the mentorship of **Dr. Parul Saxena** (Assistant Professor) as the requirement of skill based mini project in 1st year of Master of Computer Science and Engineering from Madhav Institute of Technology and Science, Gwalior.

Dr. Parul Saxena
(Assistant Professor)
Computer Science and Engineering

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

DECLARATION

I hereby declare that the work being presented in this project report, for the fulfilment of partial requirement of the award of the skill based mini project in 1st year of Master of Computer Application in Computer Science and Engineering at Madhav Institute of Technology & Science, Gwalior is an authenticated and original record of my work under the mentorship of Dr.Parul Saxena(Assistent Professor), MITS GWALIOR.
I declare that I have not submitted the matter embodied in this report anywhere else.

Romesh Singh Sengar
0901CA211050
2021-2023 Year,
Master of Computer Application,
Computer Science and Engineering

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

ACKNOWLEDGEMENT

The full semester project has proved to be pivotal to my career. I am thankful to my institute, **Madhav Institute of Technology and Science** to allow me to continue my disciplinary project. I extend my gratitude to the Director of the institute, **Dr. R. K. Pandit** and Dean Academics, **Dr. Manjaree Pandit** for this.

I would sincerely like to thank my department, **Department of Computer Science and Engineering, for allowing** me to explore this project. I humbly thank **Dr. Manish Dixit**, Professor and Head, Department of Computer Science and Engineering, for his continued support during the course of this engagement, which eased the process and formalities involved.

I am sincerely thankful to my faculty coordinator. I am grateful to the guidance of **Dr. Parul Saxena**, Assistant Professor, Computer Science and Engineering, for his continued support and guidance throughout the project. I am also very thankful to the faculty and staff of the department.

Romesh Singh Sengar
0901CA211050
2021-2023 Year,
Master of Computer Application,
Computer Science and Engineering

ABSTRACT

The railway network of our country is one of the most complex establishments. You can design a database solution for this network and make the management of the same more natural. Your system should have the following pieces of information.

Station name

Track that connect those station (to keep things simple, you can assume that only one track run between two stations) Train IDs with names.

Schedules of the trains

The train schedules should have information on the station from where the train starts and by when it reaches the destination. It should also include information on which stations it passes through during its journey.

To keep things simple, you can assume that every train completes its journey within a day, and they run daily. However, you'll also need store information on the sequence of the station a train passes through. For example, if a train start from Delhi and goes to Kolkata through Lucknow, then you'll need to add the arrival and departure time of the train for all three stations. Keeping the station sequence will allow easy management of train and their data.

Till here, the project is rather easy. You can more challenging by adding the passenger information of every train such as its coaches, seat number, type of coaches, passenger names, and so on. This project might take some time to complete, but it'll help you showcase your knowledge of database management solution while solving significant issue of a public authority.

CONTENT

S.NO		Page No.
1	Cover Page	1
2	Certificate	2
3	Declaration	3
4	Acknowledgement	4
5	Abstract	5
6	Content	6

TITLE		PAGE NO.
7	Introduction	7
8	Objective and Scope of the Project	8
9	Entities, Attributes	9-10
10	Normalization and Final list of Relationship	11
11	Final List of Relationship	12
12	Create and Insert SQL Queries	13-14
13	Insert SQL Queries	15-16
14	SQL Queries related to report generation	17-18
15	Conclusion	19

Introduction

Database is an organisation collection of data. The data is typically organised to model aspect of reality in a way that support processes requiring information. A DBMS essentially serves as an interface between the Database and end users or application programmers, ensuring that data is consistently organised and remains easily accessible. The DBMS manages three important things: the data, 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 in 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 application from needing to know where data is stored or having to be concerned about changes to the physical structure of data. A DBMS makes it possible for users to create read update and delete data in a database.

Objective and Scope of the project

The main purpose of maintaining database for Railway Reservation System is a reduce the manual errors involved in the booking and cancelling of tickets and make if convenient for the customer and providers to maintain the data about the customer and also 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 the clients can make various enquiries about train between stations. Due to this, sometimes a lot of problems occurs and they are facing may disputes with customers. To solve these problems, we design a Database which includes customer details, availability of seats in trains, no of trains and their details.

This project is about creating the database for railway reservation system. The aim is to design and develop a database maintaining the records of different train, passengers, tracks, stations, schedule and routes.

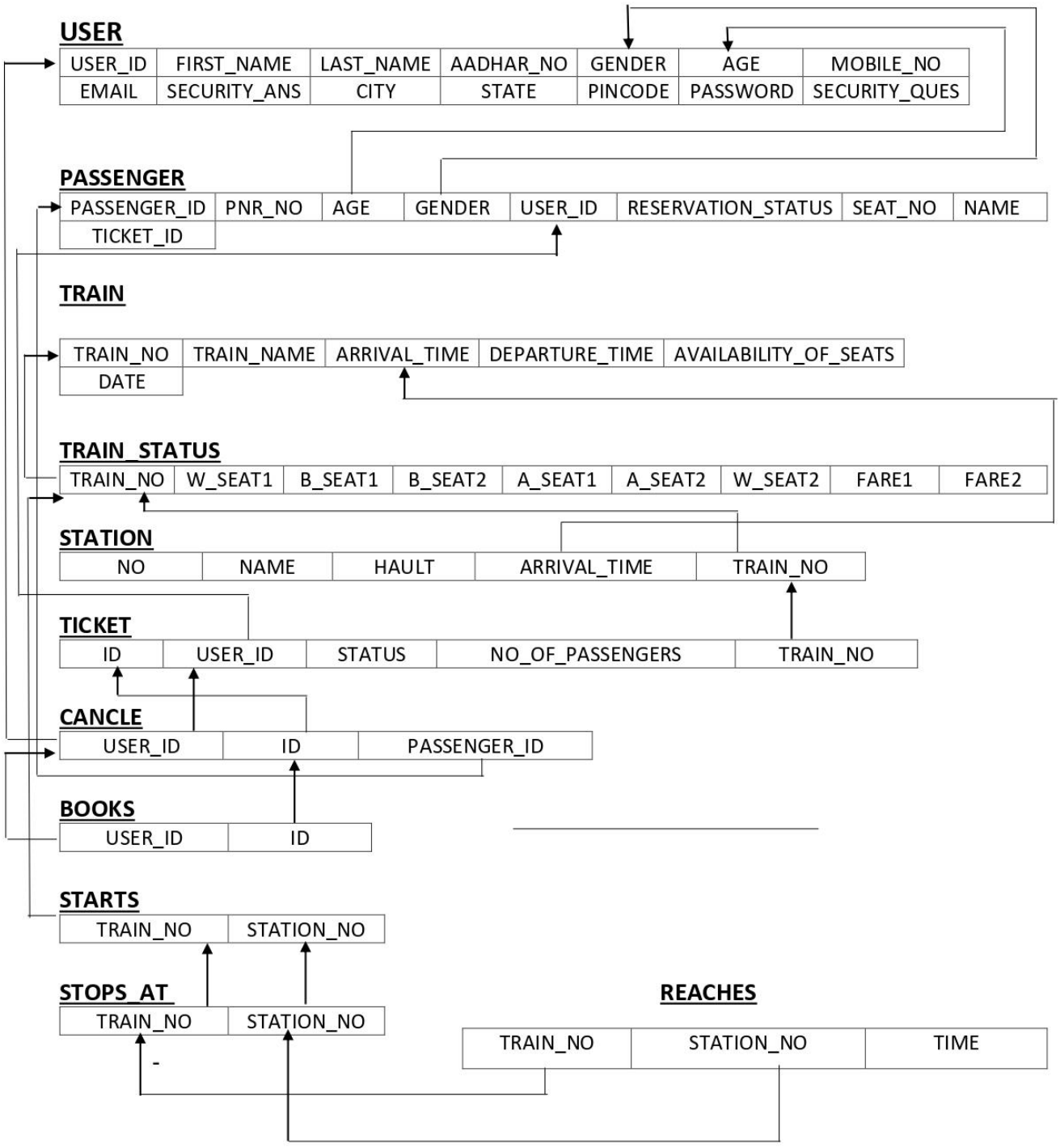
The project also consists of station names, the track that connect those stations, schedules of the train and the information of the station.

Entities, Attributes

ENTITIES	ATTRIBUTES
USER	**USER_ID** PASSWORD FIRST_NAME LAST_NAME GENDER AGE EMAIL_ID 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
TRAINS	**TRAIN_NO** TRAIN_NAME SOURCE DESTINATION ARRIVAL_TIME DEPARTURE_TIME AVAILIBILITY_OF_SEATS TRAIN_NO A_SEATS1 A_SEATS2 A_SEATS3 B_SEATS1 B_SEATS2 B_SEATS3

	W_SEATS1 W_SEATS2 W_SEATS3
STATIONS	NAME **NO** TRAIN_NO ARRIVAL_TIME HAULT
TICKET	**ID** TRAIN_NO BOOKED_USER STATUS NO_OF_PASSENGERS

NORMALIZATION AND FINAL LIST OF RELATIONS



FINAL LIST OF RELATIONSHIPS

- BOOKING - Temary Relationship Between USERs, TRAINs, PASSENGERs & TICKET.
- STARTS - Between TRAINs & STATIONs.
- REACHES - Between TRAINs & STATIONs.
- CANCEL – Between USERs & TICKET.
- STOPS_at - Between TRAINs & STATIONs.

CREATE AND 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(50) , gender char, age, int ,mobile_no varchar(50), email varchar(50), city varchar(50), state varchar(50), pin code 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, availability_of_seats, date,)

- Create table if not exists STATION (no int, name varchar(50), haul tint, arrival_time, train_noint, primary key(station_no, train_no)constariant forgein key(train_no) references TRAIN(train_no).

- Create table if not exists TRAIN_STATUS (train_no int primary key, b_eats1int, b_seats2 int, a_seats1 int, a_seats2 int, w_seats1int, 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 forgein key(user_id) references USER(user_id), constraint forgein key (train_no) references TRAIN(train_no)).

- Create table if not exists PASSENGER (passenger_id int primary key, pnr_no int, gender char, user_id int, reservation status char, seat_no 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, constraint foreign key(train_no) referencrs 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) referencrs 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 (2023,'Ramesh','Senger', 764723374002','M',23,' 7509923006',' romesh.sengar@gmail.com','Gwalior',Madhya Pradesh','474005','12345@#','favouritecolour','black'),(1993,'Aakash','Agarwal','332957946796'.'M',29,'332957946796','aakashagarwal.social@gmail.com,',Bhagalpur','Bihar','812001','Aakash@1993','favouratefood','Chinese')

- Insert Into

TRAIN(train_no, train_name, arrival_time, departure_time, availability_of_seats, date)

Values(12711','pinakini exp','113000','114000'.A'20230112'),(12315,',Coromandel Exp','124500','125000','NA',20230112)

- Insert Into

TRAIN_STATUS (train_no, w_seats1, b_seats1, b_seats2, a_seats1, a_sears2, w_seats2, fare, fare2)

Values(12711,10,4,,0,1,1,0,100,450),(12315,15,10,5,,0,0,2,1,300,600)

- Insert Into

TICKET (id, user_id, ,status, no_of_passengers, train_no)

Values(4001,2023,'C',1,12711),(3002,1993,'NC','1,'12315)

- Insert Into

PASSENGERS (passenger_id, pnr_no, age, gender, user_id, reservation_status, seat_no, name, ticket_id

Values(5001,78965,23,',M',2023,'C','B6-45','Ramesh,4001),(5002,54523,29,'M',1993,'W'.'B3-21,'Aakash',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(12311,222,'040000'),(12315,111,'053500)

Insert Into:

BOOKS(user_id, id)

Values(2023,4001),(1993,4002)

Insert Into:

CANCEL(user_id, id, passenger_id)

Values(1993,4001,500`):

SQL QUERIES RELATED TO REPORT GENERATION

1. **Print User Id And Name Of All Those User Who Booked Ticket For Pinakini Express.**

Select a user_id,concat(u.first_name,u.last_name)as name from user.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'
(Placeholder1)

2. **Print Details Of Passengers Travelling Under Ticket No 4001.**

Select*

From passenger
Where ticket _id like 4001;

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'

4. **Display Time At Which Train No. ----- Reaches Station No. -----**

Select r.*,s.name
From reaches r.station
Where r.station_no=s.no

5. **Display Details Of All Those Users Who Cancelled Tickets For The 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 Orderof 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 fare1asc

7. **Display Passenger Details For The Train Pinakini Express.**

Select p.*
From passenger p.train t, ticket tc
Where tc.train_no = t.train_no and tc.id = p.ticket_idand t.train_name like 'Pinakini Exp'

8. **Display Immediate Train From Tirupathi To Vijaywada.**

Select distinct t.*
From train t.station, s.starts st.stops_at sa.
Where st.station_no = (select no from station where name like 'vijayawada'

9.Display The Train No. Which Haults For More Time In Station No --

Select train_no
From station
Having max(hault)

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

CONCLUSION

In our project railway system, we have all the information saved regarding the train, passengers, tracks, where and how The Train moves, station, schedules, routes and what it consists of. We had considered the most important requirements only many more feature and details can be added to our project in order to obtain even more user -friendly applications.

These applications are already in progress and in future they can be upgraded and many become part of Amazing Technology