ONLINE CAR RENTAL SYSTEM

BTech/III Year CSE/V Semester

15CSE302/Database Management Systems

Project Review -3

Rollno	Name
CB.EN.U4CSE18008	Balaji D
CB.EN.U4CSE18026	Jinka Maruthi Prasad
CB.EN.U4CSE18036	Nidharshan A
CB.EN.U4CSE18038	Nitheese T
CB.EN.U4CSE18061	Talluri Tarun Teja

Amrita School of Engineering, Coimbatore

Department of Computer Science and Engineering

2020 -2021 Odd Semester

Table of Contents

1.Abstract:

- The purpose of this project is to develop a system design especially for car rental business where customers can book their cars using the website.
- A customer should login with a username and password. A new customer should create an account. Admin also should have an account to modify drivers and cars. There is a point-based system for regular customers. Customers can also choose to have a driver for their car.
- Customers can give their feedbacks and the admin will be able to see them in order to improve the business. Customers can also ask their queries which in turn will be answered by Admin.

2.Bussiness rules:

After logging in, the customer will be able to see cars on different categories (gold, silver, etc,.) and their respective rate/km and availability. The customer should also provide the duration for the required car. After choosing a car, the customer can also hire a driver as per his/her necessity. Based on the total distance traveled by the hired car and driver charges (if hired), the customer's bill will be calculated.

Payment can be done through cash or card. The customer will be rewarded with points according to his/her payment. With this point, the customer can seek concessions in their forthcoming hires. The customers can also shoot their feedbacks and queries in the respective sections.

Coming to the admin interface, the admin also has a username and password. An admin can view or change the details of the cars and drivers. The Admin can also change the price of the cars. He/she can add/remove cars or drivers. The admins will also be able to see the feedback given by a customer. The admin can also decide the concession to be given to a customer if they choose to redeem their points.

Module specification:

Add/remove Car: The Admin can add or remove car so that the user can see the available cars with their feedbacks given by users.

Add/remove Drivers: The Admin can add or remove drivers so that the user can see the available drivers with their feedback given by users.

View Available Cars: According to the category user can view cars with the price for the ride.

Booking Car: The user can book a car with or without a driver as per his/her necessity.

Car on rent: The Customer can easily get the car whenever they need to on the rent with the use of this system.

Feedback: The customer can give feedback on the ride.

Manage Rent: The Admin can manage the rent and discount as per a user's points.

View Feedback: The admin can reply to the given feedbacks.

Payment details: The rent and the payment details of the car with the discount (if redeemed) will be available to the user.

Tables:

- Admin
- User
- Driver
- Enquiry
- Feedback
- Car detail
- Rent detail

Output:

- Details of availability of Cars and drivers
- Order details of customers
- Feedback given (if any)
- Payment details

3.Preview for the project:

Introduction

- Online car rental service is just like any other rental service.
- Businessmen offer car rental service utilizing the use of information technology to improve the level of efficiency.
- This project offers the best of services-both in terms of man and machine.
- It offers the best rates and includes the different categories of cars from budget class to luxury class.
- Customers can also hire drivers for a rented car.

It provides a point system by which regular customers can enjoy discounts.

Need and motivation

The objective of the proposed Online Car Rental System, the users are able to enter the company's website for searching and reserving their favorite cars easily through the Internet and it can be access anywhere anytime in the world.

Tools used

- JavaScript
- Html
- Css
- Php
- Xampp server
- Mysql server
- Apache server

4.Project Analysis:

a. List of modules in the project

- Add/remove Car
- Add/remove Drivers
- View Available Cars
- View Available Cars
- Booking Car
- Car on rent
- Feedback
- Manage Rent
- View Feedback
- Payment details

b. Module-wise explanation

Add/remove Car: The Admin can add or remove car so that the user can see the available cars with their feedbacks given by users.
Add/remove Drivers: The Admin can add or remove drivers so that the user can see the available drivers with their feedback given by users.
View Available Cars: According to the category user can view cars with the price for the ride.
Booking Car: The user can book a car with or without a driver as per his/her necessity.
Car on rent: The Customer can easily get the car whenever they need to on the rent with the use of this system.
Feedback: The customer can give feedback on the ride.
Manage Rent: The Admin can manage the rent and discount as per a user's points.
View Feedback: The admin can reply to the given feedbacks.
Payment details: The rent and the payment details of the car with the discount (if redeemed) will be available to the user.

Process 1: The client or the customer will log into the system by making athe website and selecting the 'Login' option. If the person is a 'first time user', then he/she will select the option 'Register'**.**
Process 2: The reservation dashboard allows a client/ user to create a booking, cancel a booking, schedule an advance booking, change the personal details, renew account as well as change a password.

Process 3: Previously, after selecting the pick-up and drop off points on a particular date and at the particular time, the user has to choose a type of vehicle which would be available for service. The user also needs to select a type of plan with respect to the service avail duration, for instance hourly type plan or distance type plan or to avail executive services
Process 4:After the successful payment, the client is given an unique receipt number, specified in the dashboard about the upcoming booking number After providing these details, the amount is credited to the service vendor's. Therefore, this exchange marks the user as a positive customer and hence the system starts working on the execution of the service asked by the client.
Process 5:After the user has completed a successful transaction and a reservation, the client usermust log out or sign off. If he/she wants to access the dashboard again, then the client canreserve again with the existing login session without signing off. But a time-out session provisionis also present in the system.

5.Project Design:

Entities

- Customer
- Admin
- Booking
- Car
- Driver
- Payment
- Feedback

Attributes

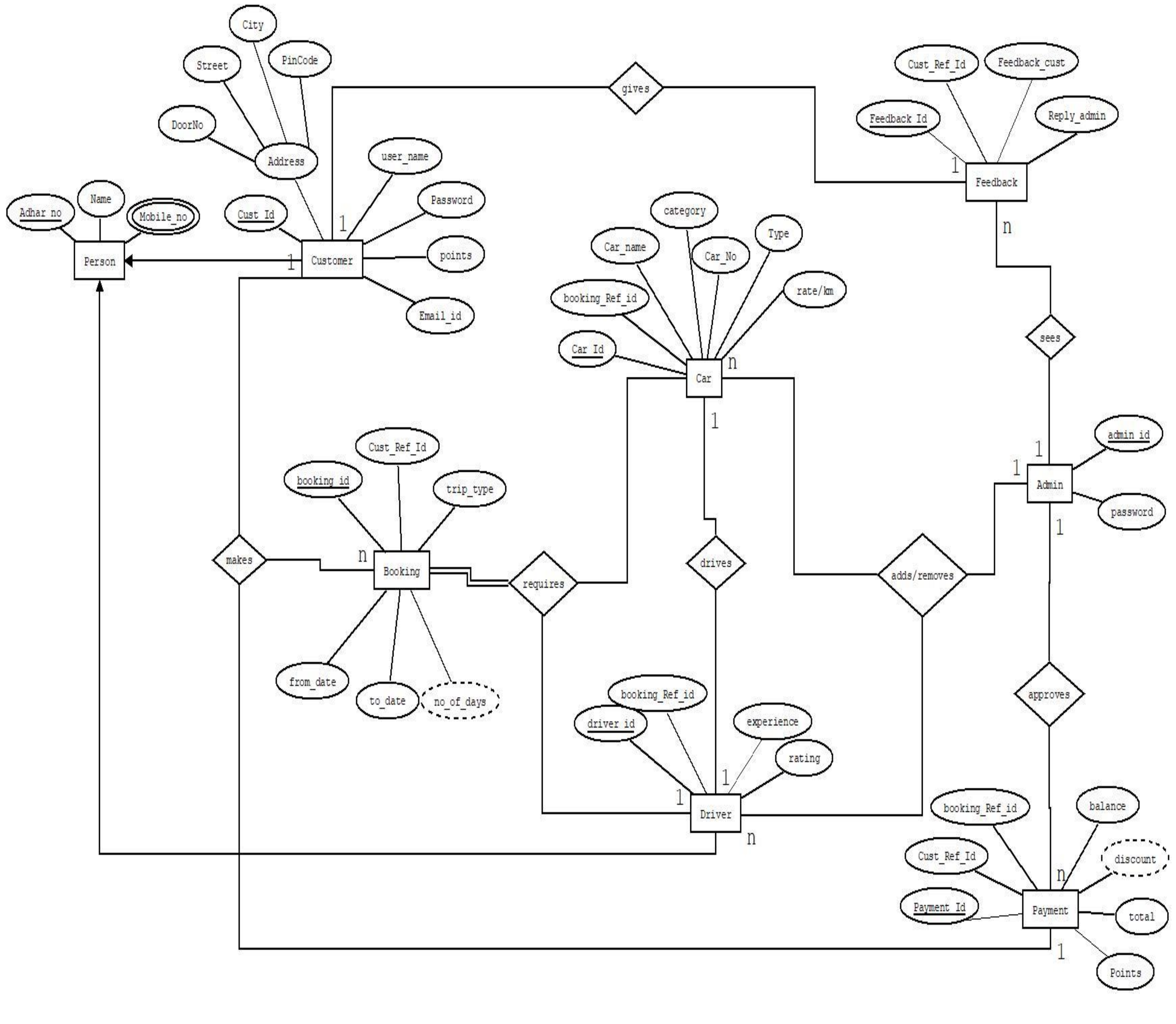
- Customer : Cust_Id,Name,Mobile_No,Address,Email_id,Password,points
- Admin : admin_id,password
- Booking : booking_id,cust_ref_id,to_date,from_date,no_of_days,trip_type
- Car : Car_Id,booking_ref_id,Car_name,Car_No,category,type,rate/km
- Driver : driver_id,booking_ref_id,experience,rating
- Payment : payment_id,cust_ref_id,booking_ref_id,total,discount,balance
- Feedback : feedback_id,cust_ref_id,feedback_cust,reply_admin

Relationships

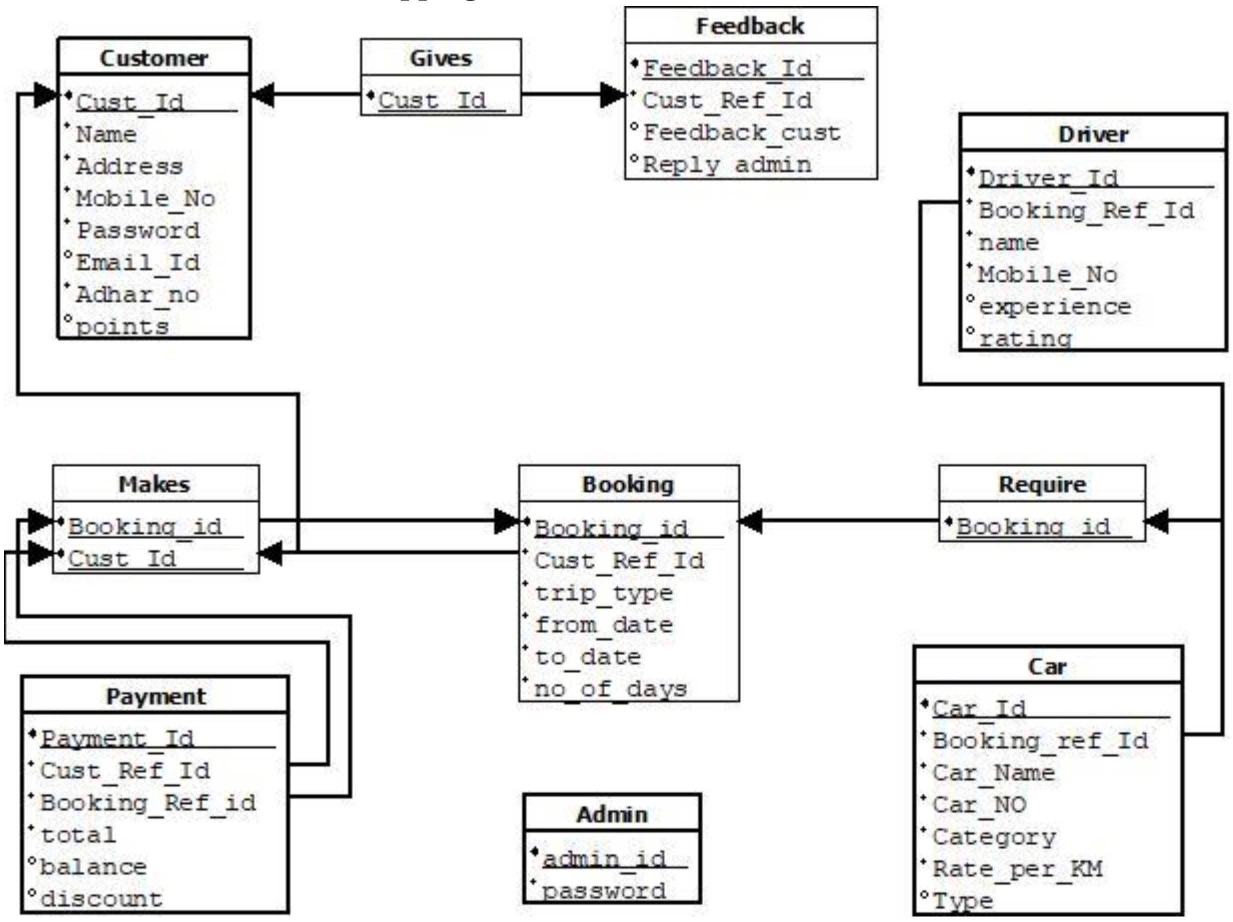
- Customer-Booking/Payment : makes
- Customer-Feedback : gives

- Admin-Car/Driver : adds/removes
- Admin-Payment : approves
- Admin-Feedback : views
- Booking-Car/Driver : requires
- Car-Driver : drives

ER Diagram



ER to Relational Schema Mapping



6.Normalization

Initial Schema:

Customer(Cust_Id, Adhar_Id, Name, Mobile_No, Address, Email_id, user_name,

Password, points)

Admin (admin_id,password)

Car (Car_Id, Car_name,Car_No,category,type,rate/km,availability)

Driver(driver_id,Adhar_Id,experience,rating,Name,Mobile_no,

Address, availability)

Booking (booking_id, cust_ref_id, car_ref_id, driver_ref_id, to_date, from_date, no_of_days, trip_type)

Payment (payment_id,booking_ref_id,total,discount,points)

Feedback (feedback_id,cust_ref_id,feedback_cust)

Queries (query_Id, cust_ref_id, cust_query, reply_admin)

1)Customer:

Customer(Cust_Id,Adhar_Id,Name,Mobile_No,Address,Email_id,user_name,

Password,points)

Cust_ID	**Adhar_Id**	**Name**	**Mobile_No**	**Address**	**Email_Id**	**user_name**	**Password**	**Points**
C01	12345	RAJU	9788657098	41, Raghunayakula Street,Chennai,Tamil Nadu	sabariraj@gmail.com	Raju_c01	rajurocks	13
C02	12346	JOHN	9678546578	108,Harris Rd,Chennai,Tamil Nadu	johnsins@gmail.com	John_c02	john967	45
C03	12347	MICHAEL	9876457656	I-138,vinayagapuram1st Main Roa,Mmda Colony,arumbakkam, Chennai,Tamil Nadu	michaelray@gmail.com	Michael_c03	michaelhere	34

FD Closure:

{ Cust_Id --> Email_id, Cust_Id --> user_name, Cust_Id --> Password, Cust_Id --> points, Adhar_Id --> Email_id, Adhar_Id --> Adhar_Id, Cust_Id --> Password, Adhar_Id --> points, Adhar_Id --> Name, Adhar_Id --> Mobile_No, Adhar_Id --> Address }

Attribute Closure:

(Cust_Id)+ = { Cust_Id, Email_id, user_name, Password, points }

(Adhar_Id)+ = { Adhar_Id, Email_id, user_name, Password, points ,Name,Address,Mobile_No}

SUPER KEYS:

1)Cust_Id

2)Adhar_Id

3)Mobile_No

4)Email_id

5)Cust_Id,Adhar_Id,Name

6)Cust_Id,Name,points

7)Cust_Id,Mobile,points

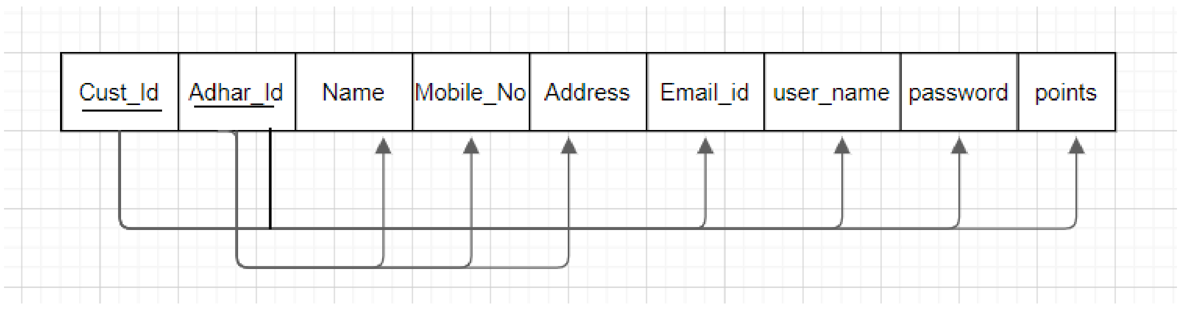
8)Cust_Id,Adhar_Id,Name,Mobile_No

9)Cust_Id,Adhar_Id,Name,Mobile_No,Address

10)Cust_Id,Adhar_Id,Name,Mobile_No,Address,Email_id,user_name,Password,points

Anomalies:

No possible anomalies here.

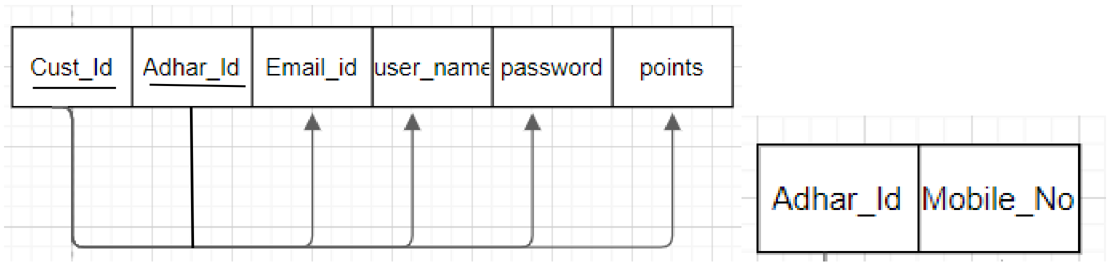


1NF:

So here **Mobile_no** is a **multi valued attribute,** hence we decompose the table into:

1)CustomerMobileNo(Adhar_Id,Mobile_No)

2)Customer(Cust_Id,Adhar_Id,Name,Address,Email_id,user_name,Password,points)



FD Closure:

R1:CustomerMobileNo:

Adhar_Id,Mobile_No -->> Adhar_Id,Mobile_No

CustomerMobileNo relation satifies 1NF(no multi valued attribute),2NF(no partial dependencies),3NF(no transitive dependenices),BCNF(Adhar_Id,Mobile_No is the super key).

R2:Customer:

{ Cust_Id --> Email_id, Cust_Id --> user_name, Cust_Id --> Password, Cust_Id --> points, Adhar_Id --> Email_id, Adhar_Id --> Adhar_Id, Cust_Id --> Password, Adhar_Id --> points, Adhar_Id --> Name, Adhar_Id --> Address }

->There is a **partial dependency** in **Customer** relation.

Chase Method:

	Cust_Id	**Adhar_Id**	**Name**	**Mobile_no**	**Address**	**Email**	**Username**	**password**	**points**
R1	b11	a2	b13	a4	b15	b16	b17	b18	b19
R2	a1	a2	a3	~~b24~~ a4	a5	a6	a7	a8	a9

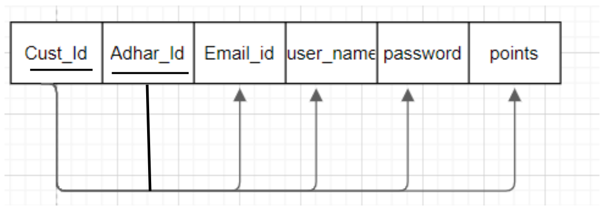
→As in R2 row all entries are a ,this decomposition is **lossless.**

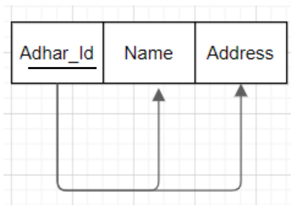
2NF:

Since there is a partical dependency for the relation Customer (name , address is only depentent on adhar_ID) ,we decompose the relation into:

1)Customer(<u>Cust_Id</u>,<u>Adhar_Id,</u>Email_id,user_name,Password,points)

2)CustomerPerson(<u>Adhar_Id</u>,Name,Address)





FD Closure:

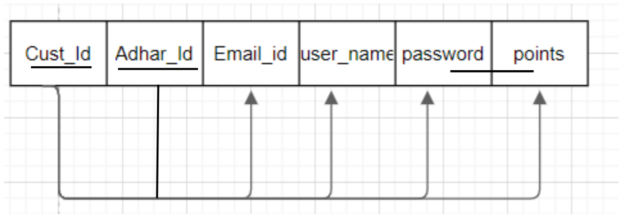
R1:CustomerPerson:

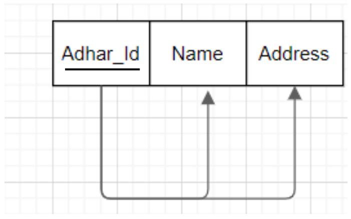
{Adhar_Id -->Name,Address}

Customer Person relation satifies 1NF(no multi valued attribute),2NF(no partial dependencies),3NF(no transitive dependenices),BCNF(Adhar_Id is the super key).

R2:Customer:

{ Cust_Id --> Email_id, Cust_Id --> user_name, Cust_Id --> Password, Cust_Id --> points, Adhar_Id --> Email_id, Email_Id --> user_name,, Cust_Id --> Password, Adhar_Id --> points,}





Chase Method:

	Cust_Id	**Adhar_Id**	**Name**	**Address**	**Email**	**Username**	**password**	**points**
R1	b11	a2	a3	a4	b15	b16	b17	b18
R2	a1	a2	~~b23~~ a3	~~b24~~ a4	a5	a6	a7	a8

→As in R2 row all entries are a ,this decomposition is **lossless.**

3NF:

It satisfies 3NF condition as there are no transitive dependencies.

BCNF:

It satisfies BCNF condition as for all functional dependencis A->B,A is the super key.

Is it Dependency Preserving?

It is dependency preserving because all the functional dependencies are derivable from the tables Customer,CustomerPerson and CustomerMobileNo.

FDs of Customer: (Cust_Id --> Email_id, Cust_Id --> user_name, Cust_Id --> Password, Cust_Id --> points, Adhar_Id --> Name, Adhar_Id --> Mobile_No, Adhar_Id --> Address**)**

FDs of CustomerPerson: (Adhar_Id --> Name, Adhar_Id --> Mobile_No, Adhar_Id --> Address**)**

FDs of CustomerMobileNo: (Adhar_Id,Mobile_No -->> Adhar_Id,Mobile_No)

<u>Canonical Cover:</u>

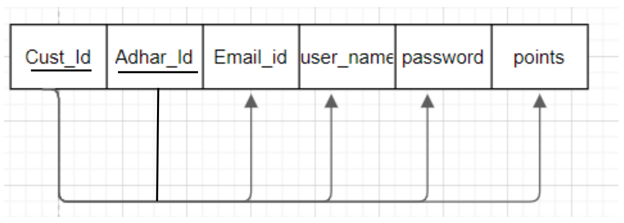
Customer:

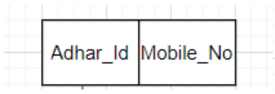
Cust_Id,Adhar_Id --> Email_id,user_name,Password,points

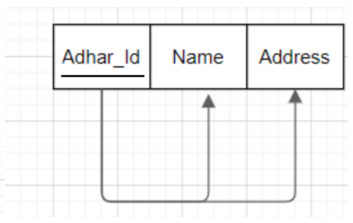
CustomerPerson:

Adhar_Id --> Name, Address

After normalizing:







<u>2)Admin:</u>

Admin (<u>admin_id</u>,password)

admin_id	password
A01	#admin_here

FD Closure:

{admin_id --> password}

Attribute Closure:

(admin_id)+ = {admin_id, password}

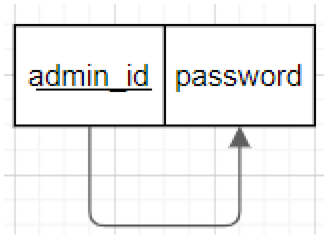
Super Keys:

1) admin_id

2) admin_id,password

Anomalies:

No possible anomalies here.



1NF:

It satisfies 1NF condition as there are no muti-valued attributes.

2NF:

It satisfies 2NF condition as there are no partial dependencies

3NF:

It satisfies 3NF condition as there are no transitive dependencies.

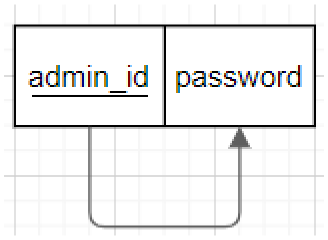
BCNF:

It satisfies BCNF condition as for all functional dependencis A->B,A is the super key.

Canonical Cover:

admin_id --> password

After Normalizing:



<u>3)Car:</u>

Car (<u>Car_Id</u>, Car_name,Car_No,category,type,rate/km,availability)

<u>Car_Id</u>	**Car_name**	**Car_No**	**category**	**type**	**rate/km**	**availability**
V01	BENZ	001	PREMIUM	COZY	60	1
V02	ALTO	002	SILVER	NORMAL	20	1
V03	SELTOZ	003	GOLD	SUV	35	1

FD Closure:

{Car_Id-->Car_name, Car_Id-->Car_No, Car_Id-->category, Car_Id-->type, Car_Id-->rate/km,Car_Id-->availability}

Attribute Closure:

(Car_Id)+ = {Car_Id, Car_name, Car_No, category,type,rate/km,availability}

Super Keys:

1)Car_id

2)Car_id,booking_ref_id

3)Car_id,Car_name

4)Car_No

5)Car_No,Car_id

6)Car_No,Car_id

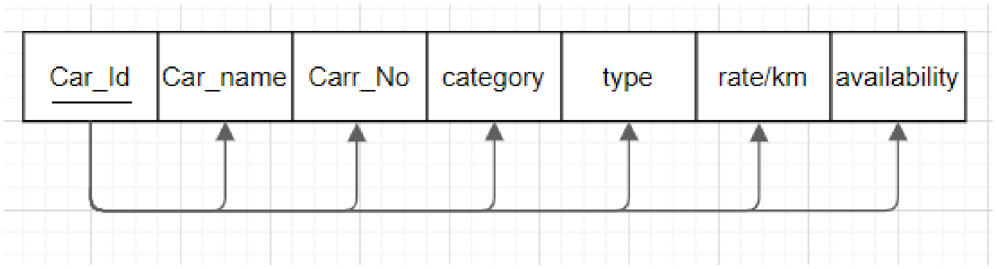
7)Car_No,Car_name

8)Car_name,Car_No,category,type,rate/km

9)Car_Id, Car_name,Car_No,category,type,rate/km,availability

Anomalies:

Both insertion anomaly and update/delete anomaly are possible here due to the presence of a foriegn key 'booking_ref_Id' which is referencing 'Booking_Id' in Booking Table.



1NF:

It satisfies 1NF condition as there are no muti-valued attributes.

2NF:

It satisfies 2NF condition as there are no partial dependencies

3NF:

It satisfies 3NF condition as there are no transitive dependencies.

BCNF:

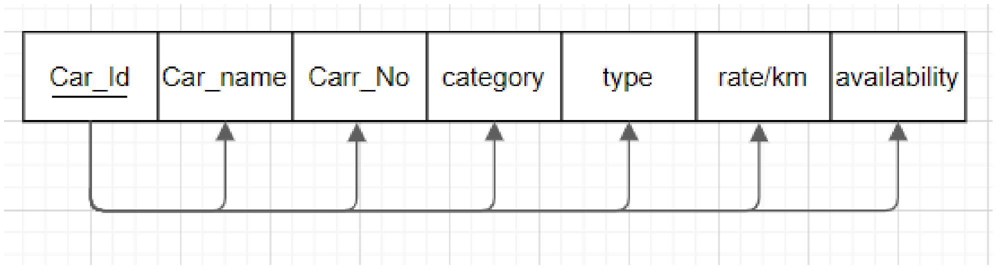
It satisfies BCNF condition as for all functional dependencis A->B,A is the super key.

driver_id	Adhar_Id	experience	rating	Name	Mobile _no	Address	availability

Canonical Cover:

Car_Id --->>> Car_name,Car_No,category,type,rate/km,availibility

After normalizing:



4)Driver:

D01	78956	5	4	Ramesh Kanna	8050201346, 9150136567	9 c, Jeevarathinam St Ksr Ngr, Ambattur, Chennai, Tamil Nadu	1
D02	10578	1	3.25	Suresh Kumar	9875461023	159 , N M Road, Aminjikarai, Chennai, Tamil Nadu	1
D03	61513	6	4.5	Rahul	9543210179	New No 568, Anna Salai, Teynampet, Chennai, Tamil Nadu	1

Driver(<u>driver_id</u>,<u>Adhar_Id</u> , experience,rating,Name,Mobile_no,Address,availability)

FD Closure:

{driver_Id --> experience, driver_Id --> rating, driver_Id --> availability , Adhar_Id --> experience, Adhar_Id --> rating, Adhar_Id --> Name, Adhar_Id --> Mobile_No, Adhar_Id --> Address , Adhar_Id --> availability }

Attribute Closure:

(driver_Id)+ = {driver_Id, experience, rating,availability}

(Adhar_Id)+ = {Adhar_Id, experience, rating,Name,Mobile_No,Address,availability}

Super Keys:

1)driver_id

2)Adhar_Id

3)driver_id,Adhar_id

4)driver_id,Name

5)driver_id,experience

6)Adhar_Id,rating

7)Adhar_id,Name

8)driver_id,Adhar_id,booking_ref_id

9)driver_id,Adhar_id,experience

10)driver_id,Adhar_id_, experience,rating

11)driver_id,Adhar_id_, experience,address

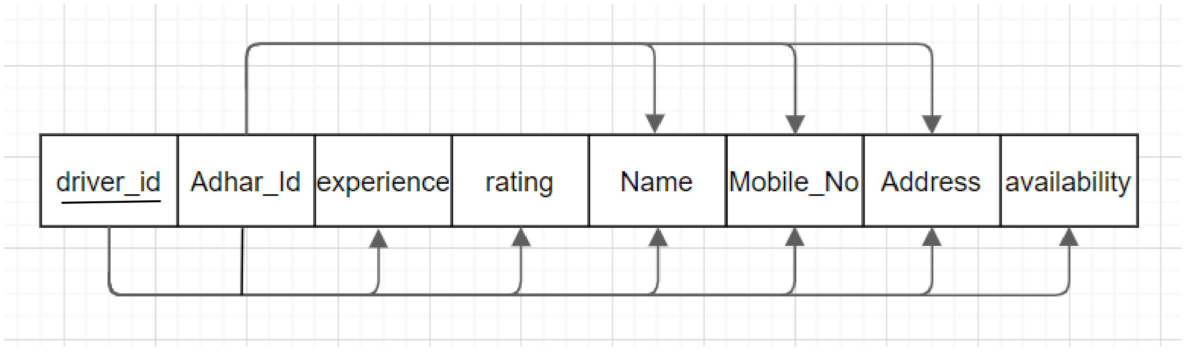
12)driver_id,Adhar_id_, experience,mobile_no

13)driver_id,Adhar_id_, experience,name

14)driver_id,Adhar_id,experience,rating,Name,Mobile_no,Address,availibility

Anomalies:

Both insertion anomaly and update/delete anomaly are possible here due to the presence two of foriegn keys 'Adhar_Id' which is referencing Adhar_Id in Customer Table and 'booking_ref_Id' which is referencing 'Booking_Id' in Booking Table.

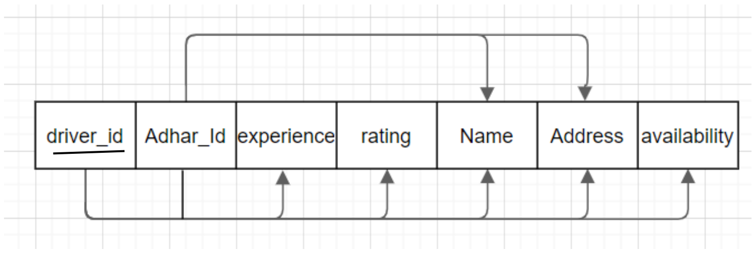


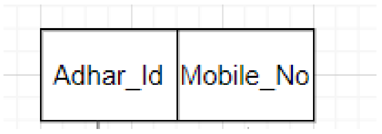
1NF:

So here **Mobile_no** is a **multi valued attribute,** hence we decompose the table into:

1)Driver (driver_id, Adhar_Id ,booking_ref_id,experience,rating,Name)

2)DriverMobileNo(Adhar_Id,Mobile_No)





R1:DriverMobileNo:

{Adhar_Id ,Mobile_no-->Adhar_Id,Mobile_No}

DriverMobileNo relation satifies 1NF(no multi valued attribute),2NF(no partial dependencies),3NF(no transitive dependenices),BCNF(Adhar_Id is the super key).

R2:Driver:

{driver_Id --> experience, driver_Id --> rating, driver_Id --> availability , Adhar_Id --> experience, Adhar_Id --> rating, Adhar_Id --> Name, Adhar_Id --> Address , Adhar_Id --> availability }

Chase Method:

	driver_Id	**Adhar_Id**	**Name**	**Mobile_no**	**Address**	**experience**	**rating**	**availability**
R1	b11	a2	b13	a4	b15	b16	b17	b18
R2	a1	a2	a3	~~b24~~ a4	a5	a6	a7	a8

→As in R2 row all entries are a ,this decomposition is **lossless.**

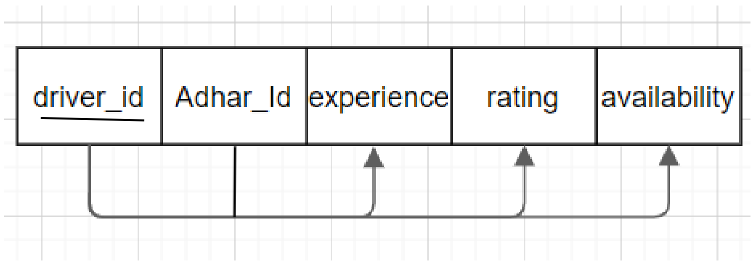
->There is a **partial dependency** in **Driver** relation.

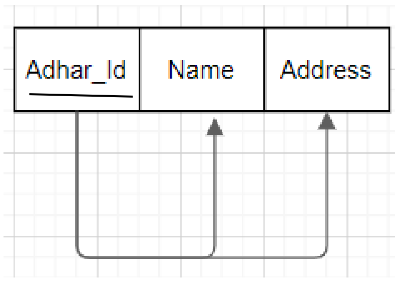
2NF:

Since there is a partical dependency for the relation Driver (name , address is only depentent on adhar_ID) ,we decompose the relation into:

1)Driver (<u>driver_id</u>, <u>Adhar_Id</u> ,booking_ref_id,experience,rating)

2)DriverPerson(<u>Adhar_Id</u>,Name,Address)





R1:DriverPerson:

{Adhar_Id -->Name,Address}

DriverMobileNo relation satifies 1NF(no multi valued attribute),2NF(no partial dependencies),3NF(no transitive dependenices),BCNF(Adhar_Id is the super key).

R2:Driver:

{driver_Id --> experience, driver_Id --> rating, driver_Id --> availability , Adhar_Id --> experience, Adhar_Id --> rating, Adhar_Id --> availability }

Chase Method:

	driver_id	**Adhar_Id**	**Name**	**Address**	**experience**	**rating**	**Address**	**availability**
R1	b11	a2	a3	a4	b15	b16	b17	b18
R2	a1	a2	~~b23~~ a3	~~b24~~ a4	a5	a6	a7	a8

→As in R2 row all entries are a ,this decomposition is **lossless.**

3NF:

It satisfies 3NF condition as there are no transitive dependencies.

BCNF:

It satisfies BCNF condition as for all functional dependencis A->B,A is the super key.

Is it Dependency preserving?

It is dependency preserving because all the functional dependencies are derivable from the tables Driver,DriverPerson and DriverMobileNo.

FDs of Driver: (driver_Id --> booking_ref_id driver_Id --> experience, driver_Id --> rating, Adhar_Id --> Name, Adhar_Id --> Mobile_No, Adhar_Id --> Address **)**

FDs of DriverPerson:(Adhar_Id --> Name, Adhar_Id --> Mobile_No, Adhar_Id --> Address **)**

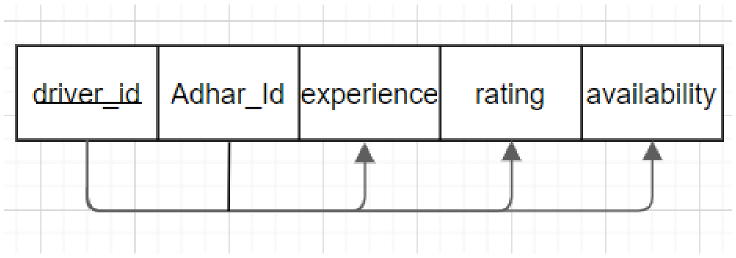
FDs of DriverMobileNo:(Adhar_Id ,Mobile_no-->Adhar_Id,Mobile_No**)**

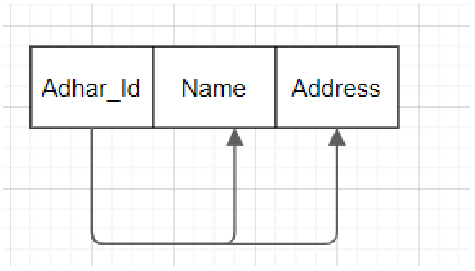
Canonical Cover:

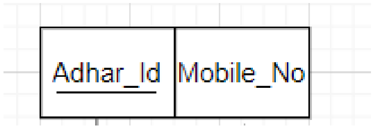
driver_id,Adhar_Id --> experience, rating,availability

Adhar_Id --> Name, Address

After normalizing:







5)Booking:

Booking (booking_id,cust_ref_id,car_ref_id,driver_ref_id,to_date,from_date,no_of_days,trip_type)

booking_id	**cust_ref_id**	**car_ref_id**	**driver_ref_id**	**to_date**	**from_date**	**No_of_days**	**trip_type**
B01	C02	V01	D03	30/05/2020	30/05/2020	1	SINGLE TIME
B02	C01	V02	D02	12/05/2020	13/05/2020	2	MULTI ROUTE
B03	C03	V03	D01	22/07/2020	26/07/2020	5	OUTSTATION

FD Closure:

{ booking_id -->to_date, booking_id -->from_date, booking_id -->no_of_days, booking_id--> trip_type,booking_id-->cust_ref_id

car_ref_id -->to_date, car_ref_id -->from_date, car_ref_id -->no_of_days, car_ref_id--> trip_type, car_ref_id-->cust_ref_id

driver_ref_id -->to_date, driver _ref_id -->from_date, driver _ref_id -->no_of_days, driver _ref_id--> trip_type, driver_ref_id-->cust_ref_id }

Attribute Closure:

(booking_id)+ = { booking_id, cust_ref_id ,to_date,from_date,no_of_days,trip_type}

(car_ref_id)+ = { car_ref_id ,cust_ref_id, to_date,from_date,no_of_days,trip_type}

(driver_ref_id)+ = { driver_ref_id ,cust_ref_id, to_date,from_date,no_of_days,trip_type}

Super Keys:

1)booking_id

2)booking_id,car_ref_id,driver_ref_id

3)booking_id,driver_ref_id,to_date

4)booking_id,no_of_days

5)booking_id,trip_type

6)booking_id,car_ref_id,to_date,from_date

7)booking_id,driver_ref_id,to_date,from_date,trip_type

8)booking_id,car_ref_id,to_date

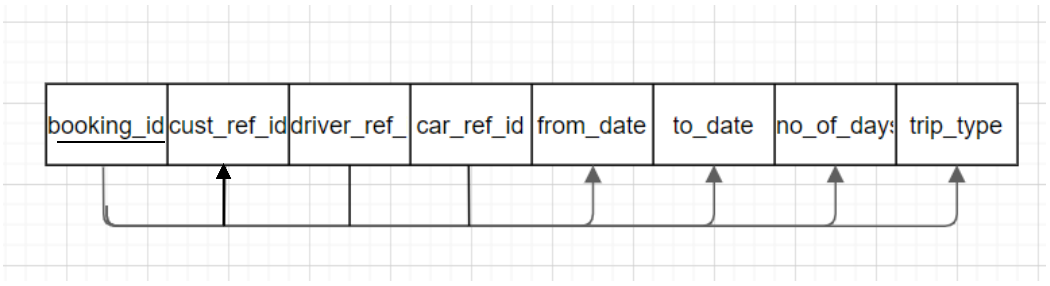
9)booking_id,car_ref_id,trip_type

10)booking_id,driver_ref_id,to_date,from_date,no_of_days

11)booking_id,car_ref_id,to_date,from_date,no_of_days,trip_type

Anomalies:

Both insertion anomaly and update/delete anomaly are possible here due to the presence of a foriegn key 'cust_ref_Id' which is referencing 'Cust_Id' in Customer Table.



1NF:

It satisfies 1NF condition as there are no muti-valued attributes.

2NF:

It satisfies 2NF condition as there are no partial dependencies

3NF:

It satisfies 3NF condition as there are no transitive dependencies.

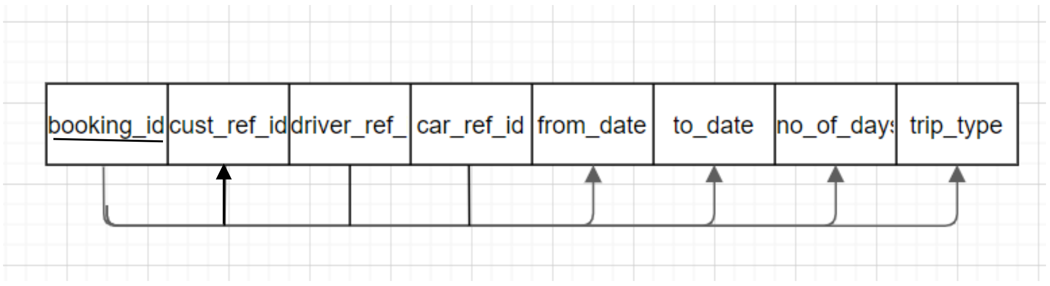
BCNF:

It satisfies BCNF condition as for all functional dependencis A->B,A is the super key.

Canonical Cover:

booking_id, car_ref_id,driver_ref_id--->>>cust_ref_id to_date,from_date,no_of_days,trip_type

After normalizing:



6)Payment:

Payment (payment_id,cust_ref_id,booking_ref_id,total,discount,points)

Payment_id	**cust_ref_id**	**booking_ref_id**	**total**	**points**	**discount**
P01	C03	B02	5000.00	34	3.00
P02	C02	B01	4572.50	45	4.00
P03	C01	B03	1116.20	13	1.00

FD Closure:

{ payment_id--> cust_ref_id, payment_id--> total, payment_id--> discount,payment_id-->points, booking_ref_id--> cust_ref_id, booking_ref_id--> total, booking_ref_id--> discount, booking_ref_id--> points,cust_ref_id-->points}

Attribute Closure:

(payment_id)+ = {payment_id,cust_ref_id,total,discount,points}

(booking_ref_id)+ = {booking_ref_id,cust_ref_id,total,discount,points}

(cust_ref_id)+ = {cust_ref_id,points)

Super Keys:

1)payment_id

2)payment_id,cust_ref_id

3)payment_id,booking_ref_id

4)payment_id,total

5)payment_id,discount

6)payment_id,points

7)payment_id,cust_ref_id,booking_ref_id

8)payment_id,cust_ref_id,total

9)payment_id,cust_ref_id,points

10)payment_id,cust_ref_id,discount

11)payment_id,booking_ref_id,total

12)payment_id,booking_ref_id,discount

13)payment_id,booking_ref_id,points

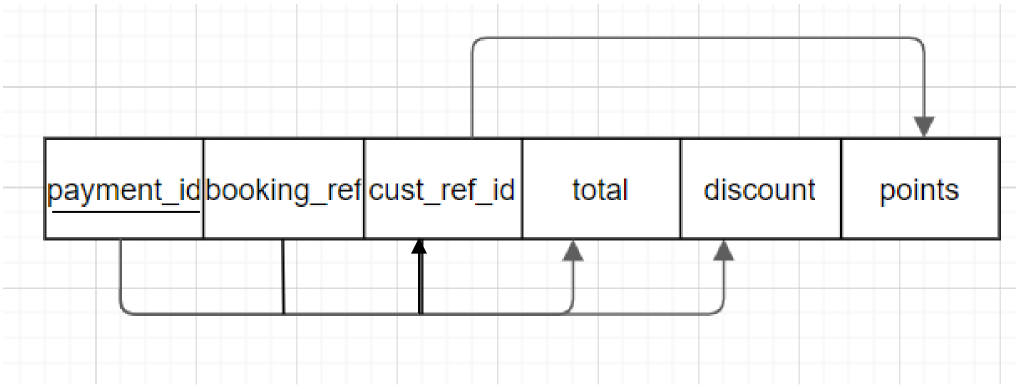
14)payment_id,booking_ref_id,points,discount

15)payment_id,booking_ref_id,points,cust_ref_id.

16)payment_id,cust_ref_id,booking_ref_id,total,discount,points

Anomalies:

Both insertion anomaly and update/delete anomaly are possible here due to the presence two of foriegn keys 'cust_ref_id' which is referencing Cust_Id in Customer Table and 'booking_ref_Id' which is referencing 'Booking_Id' in Booking Table.



1NF:

It satisfies 1NF condition as there are no muti-valued attributes.

2NF:

It satisfies 2NF condition as there are no partial dependencies

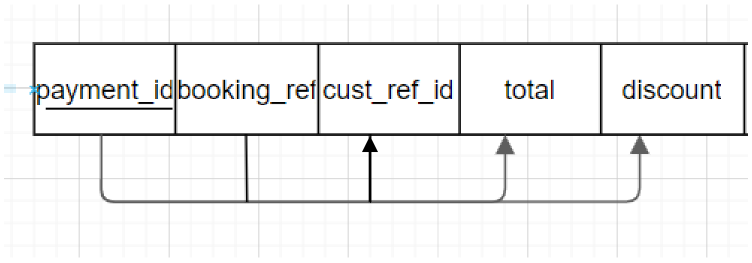
3NF:

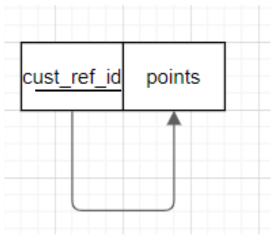
Here payment->cust_ref_id && cust_ref_id->points which is transtive dependency, hence decompose.

After decomposing:

1)Payment (payment_id,cust_ref_id,booking_ref_id,total,discount)

2)PaymentPoints (cust_ref_id,points)





FD Closure:

R1:PaymentPoints:

{cust_ref_id-->points}

R2:Payment:

{payment_id--> total, payment_id--> discount, booking_ref_id--> total, booking_ref_id--> discount,cust_ref_id--> total, cust_ref_id--> discount }

Chase Method:

	Payment_id	**cust_ref_id**	**booking_ref_id**	**total**	**points**	**discount**
R1	b11	a2	b13	b14	a5	b16
R2	a1	a2	a3	a4	~~b25~~ a5	a6

→As in R2 row all entries are a ,this decomposition is **lossless.**

BCNF:

It satisfies BCNF condition as for all functional dependencis A->B,A is the super key.

Canonical Cover:

payment_id, booking_ref_id ,cust_ref_id--->>> cust_ref_id,total,discount,points

cust_ref_id --->>> points

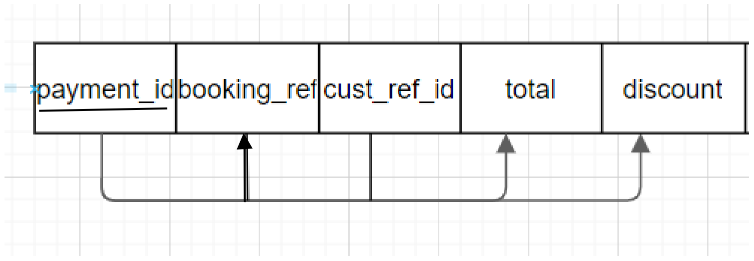
Is it Dependency Preserving?

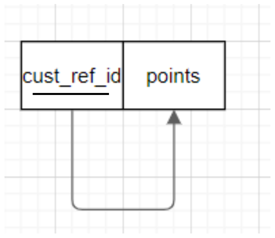
It is dependency preserving because all the functional dependencies are derivable from the tables Payment and PaymentPoints.

FDs of Payment: (payment_id--> cust_ref_id, payment_id--> total, payment_id--> discount, booking_ref_id--> cust_ref_id, booking_ref_id--> total, booking_ref_id--> discount **)**

FDs of PaymentPoints:(cust_ref_id-->points **)**

After normalizing:





7)Feedback:

Feedback (feedback_id,cust_ref_id,feedback_cust)

feedback_id	**cust_ref_id**	**feedback_cust**
F01	C02	4.5
F02	C03	3.0
F03	C01	2.5

FD Closure:

{feedback_id-->feedback_cust,cust_ref_id-->feedback_cust}

Attribute Closure:

(feedback_id)+ = {feedback_id,feedback_cust}

(cust_ref_id)+ = {cust_ref_id,feedback_cust}

Super Keys:

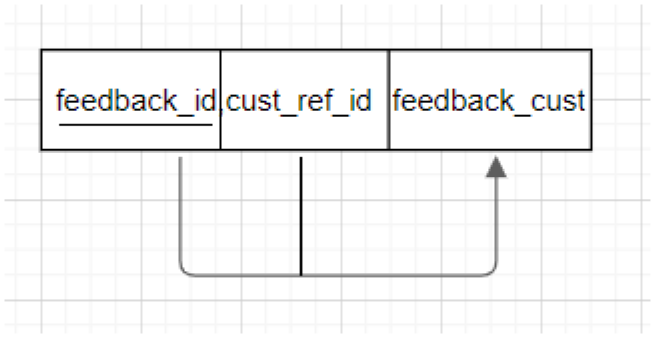
1) feedback_id
2) feedback_id,cust_ref_id,
3) Feedback_id,cust_ref_id,feedback_cust
4) Feedback_id,feedback_cust

Anomalies:

Both insertion anomaly and update/delete anomaly are possible here due to the presence of foriegn key 'cust_ref_id' which is referencing Cust_Id in Customer Table.

Canonical Cover:

feedback_id,cust_ref_id --->>> feedback_cust



1NF:

It satisfies 1NF condition as there are no muti-valued attributes.

2NF:

It satisfies 2NF condition as there are no partial dependencies

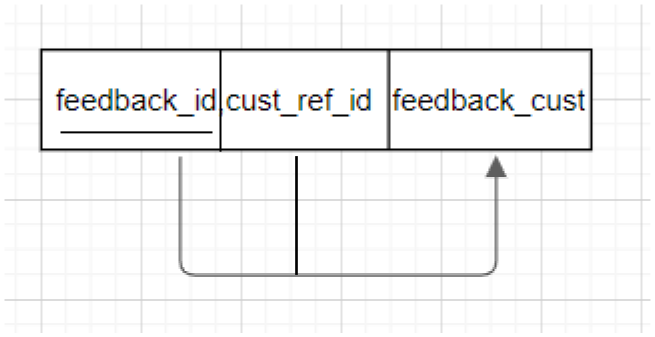
3NF:

It satisfies 3NF condition as there are no transitive dependencies.

BCNF:

It satisfies BCNF condition as for all functional dependencis A->B,A is the super key.

After normalizing:



8)Query:

Query (query_Id, cust_ref_id, cust_query, reply_admin)

query_id	cust_ref_id	cust_query	reply_admin
Q01	C01	How many bookings can a customer make?	Any number of bookings.
Q02	C03	Can we pay using credit card?	Yes.

FD Closure:

{query_id-->cust_query,query_id-->reply_admin, cust_ref_id-->cust_query,cust_ref_id-->reply_admin ,}

Attribute Closure:

(query_id)+ = {query_id, query,reply_admin}

(cust_ref_id)+ = {cust_ref_id,cust_query,reply_admin}

Super Keys:

1)query_Id

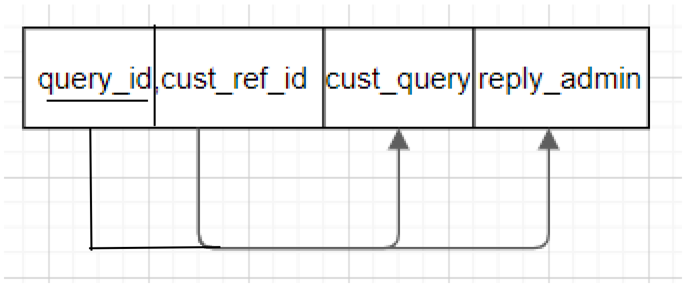
2)query_Id,cust_ref_id

3)query_Id,reply_admin

4)query_Id,cust_ref_id,reply_admin

Anomalies:

No anomalies are possible here.



1NF:

It satisfies 1NF condition as there are no muti-valued attributes.

2NF:

It satisfies 2NF condition as there are no partial dependencies

3NF:

It satisfies 3NF condition as there are no transitive dependencies.

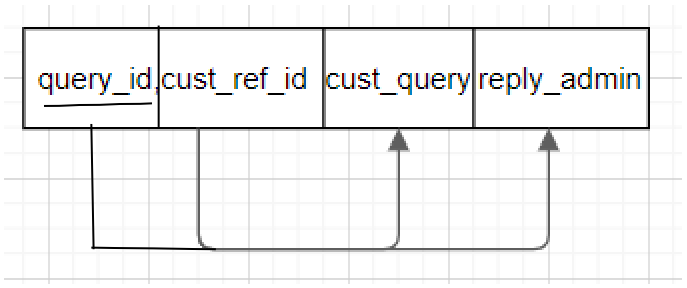
BCNF:

It satisfies BCNF condition as for all functional dependencis A->B,A is the super key.

Canonical Cover:

query_Id, cust_ref_id --->>> cust_query, reply_admin

After normalizing:



Schemas after decomposion:

Customer(Cust_Id,Adhar_Id,Email_id,user_name,Password,points)

CustomerMobileNo(Adhar_Id,Mobile_No)

CustomerPerson(Adhar_Id,Name,Address)

Admin (admin_id,password)

Car (Car_Id, Car_name,Car_No,category,type,rate/km,availability)

Driver (driver_id, Adhar_Id ,experience,rating, availability)

DriverPerson(Adhar_Id,Name,Address)

DriverMobileNo(Adhar_Id,Mobile_No)

Booking (booking_id,cust_ref_id,car_ref_id,driver_ref_id,to_date,from_date,no_of_days,trip_type)

Payment (payment_id,cust_ref_id,booking_ref_id,total,discount,points)

PaymentPoints (cust_ref_id,points)

Feedback (feedback_id,cust_ref_id,feedback_cust)

Queries (query_Id, cust_ref_id,cust_query ,reply_admin)

Representing Entire E-R diagram As A Single Schema

Car_Rental_Schema(Cust_Id,Adhar_Id,Name,Mobile_No,Address,Email_id,user_name,Password,points,admin_id,password,Car_Id,Car_name,Car_No,category,type,rate/km,availability,driver_id,Adhar_Id,experience,rating,Name,Mobile_no,Address,availability,booking_id,cust_ref_id,car_ref_id,driver_ref_id,to_date,from_date,no_of_days,trip_type,payment_id,cust_ref_id,booking_ref_id,total, discount,points,feedback_id,cust_ref_id,feedback_cust,query_Id,cust_ref_id,cust_query,reply_admin**)**

Checking for 1st Normal Form:

- Schema contains many non - atomic attributes and many repeating attributes.
- Hence , the table is not in first normal form.
- To make it into 1st Normal form , we decompose the table into some relations such that all table contains only unique attributes and no multi valued attributes.

- So , taking non - atomic attributes into account , the relations we get are,

CustomerMobileNo(Adhar_Id,Mobile_No)

DriverMobileNo(Adhar_Id,Mobile_No)

Car_Rental_Schema(Cust_Id,Adhar_Id,Name,Address,Email_id,user_name,Password,points,admin_id,password,Car_Id,Car_name,Car_No,category,type,rate/km,availability,driver_id,Adhar_Id,experience,rating,Name,Address,availability,booking_id,cust_ref_id,car_ref_id,driver_ref_id,to_date,from_date,no_of_days,trip_type,payment_id,cust_ref_id,booking_ref_id,total,discount,points,feedback_id,cust_ref_id,feedback_cust,query_Id,cust_ref_id,cust_query,reply_admin**)**

- **Car_Rental_Schema** has many attributes with the same name(**Name,Address,cust_ref_id,booking_ref_id,Adhar_Id,password**) , so it is also decomposed and the final relations are

  CustomerMobileNo(Adhar_Id,Mobile_No)

  DriverMobileNo(Adhar_Id,Mobile_No)

  Customer(Cust_Id,Adhar_Id,Name,Address,Email_id,user_name,Password,points)

  Driver(driver_id,Adhar_Id,experience,rating,Name,Mobile_no,Address,availability)

  Booking(booking_id,cust_ref_id,car_ref_id,driver_ref_id,to_date,from_date,no_of_days,trip_type)

  Payment (payment_id,cust_ref_id,booking_ref_id,total,discount,points)

  Feedback (feedback_id,cust_ref_id,feedback_cust)

  Query (query_Id, cust_ref_id, cust_query, reply_admin)

  Admin (admin_id,password)

  Car_Rental_Schema(Car_Id,Car_name,Car_No,category,type,rate/km,availability**)**

Thus all the tables above are of 1st normal form**.**

Checking for 2nd Normal Form:

- The relations Customer and Driver contain partial dependencies among them.
- In Customer, subset{Adhar_Id} of candidate key{Cust_Id,Adhar_Id} determines name and address.Likewise , in Driver subset{Adhar_Id} of candidate key{Cust_Id,Adhar_Id}.
- So table Customer , gets decomposed into

Customer(Cust_Id,Adhar_Id,Email_id,user_name,Password,points)

CustomerPerson(Adhar_Id,Name,Address)

- Table Driver, gets decomposed into

  Driver(<u>driver_id</u>,<u>Adhar_Id</u>,experience,rating,Mobile_no,availability)

  DriverPerson(<u>Adhar_Id</u>,Name,Address)

Checking for 3rd Normal Form:

- The relation payment has transitive dependency.
- Here payment->cust_ref_id && cust_ref_id->points which is transtive dependency, hence decompose.
- After decomposing:

  1)**Payment** (<u>payment_id</u>,cust_ref_id,booking_ref_id,total,discount)

  2)**PaymentPoints** (<u>cust_ref_id,</u>points)

Checking for BCNF

- All the relations present satisfy the conditions needed for BCN as all the determinants are super keys.

FINAL SCHEMAS :

Customer(<u>Cust_Id</u>,<u>Adhar_Id,</u>Email_id,user_name,Password,points)

CustomerMobileNo(<u>Adhar_Id</u>,Mobile_No)

CustomerPerson(<u>Adhar_Id</u>,Name,Address)

Admin (<u>admin_id</u>,password)

Car (<u>Car_Id</u>, Car_name,Car_No,category,type,rate/km,availability)

Driver (<u>driver_id</u>, <u>Adhar_Id</u> ,experience,rating, availability)

DriverPerson(<u>Adhar_Id</u>,Name,Address)

DriverMobileNo(<u>Adhar_Id</u>,Mobile_No)

Booking (<u>booking_id</u>,<u>cust_ref_id</u>,car_ref_id,driver_ref_id,to_date,from_date,no_of_days,trip_type)

Payment (<u>payment_id</u>,cust_ref_id,booking_ref_id,total,discount,points)

PaymentPoints (<u>cust_ref_id,</u>points)

Feedback (<u>feedback_id</u>,cust_ref_id,feedback_cust)

Queries (<u>query_Id</u>, cust_ref_id,cust_query ,reply_admin)

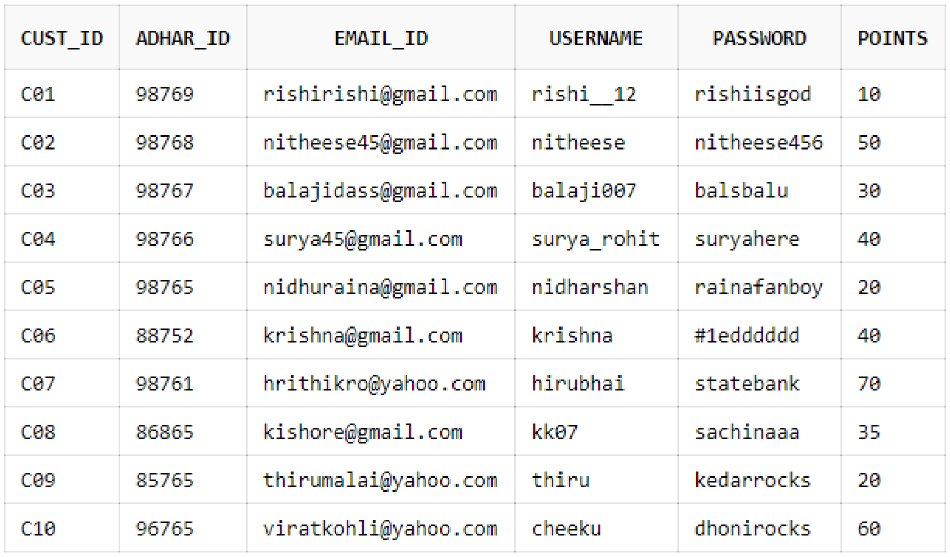
7.Backend design:

Table Creation:

- create table Customer(Cust_id varchar2(10) primary key, Adhar_id varchar2(20) not null unique, Email_id varchar2(20), username varchar2(20) not null unique, password varchar2(20) not null,points integer default 0);
- create table Customer_Person(Adhar_id varchar2(20) primary key,Name varchar2(20) not null, Address varchar2(50) not null);
- create table Customer_Mobile(Adhar_id varchar2(20),Mobile_no long not null);
- create table Admin(Admin_id varchar2(10) primary key, password varchar2(20) not null);
- create table Car(Car_id varchar2(10) primary key,Car_name varchar2(10) , Car_no varchar2(10) not null unique, category varchar2(10) not null,type varchar2(10) not null,rate_per_km float not null,availability integer not null);
- create table Driver(Driver_id varchar2(10) primary key, Adhar_id varchar2(20) not null unique, experience integer, rating integer,availability integer not null);
- create table Driver_Person(Adhar_id varchar2(20) primary key,Name varchar2(20) not null, Address varchar2(50) not null);
- create table Driver_Mobile(Adhar_id varchar2(20),Mobile_no long not null);
- create table Booking(Booking_id varchar2(10) primary key, cust_ref_id varchar2(10),car_ref_id varchar2(10),driver_ref_id varchar2(10) ,to_date date not null, from_date date not null, no_of_days integer, trip_type varchar2(10) not null, constraint fk1 foreign key(Cust_ref_id) references Customer(Cust_id),constraint fk2 foreign key(car_ref_id) references Car(Car_id),constraint fk3 foreign key(driver_ref_id) references Driver(Driver_id));
- create table Payment(Payment_id varchar2(10) primary key,cust_ref_id varchar2(10),booking_ref_id varchar2(10), total float,discount float default 0, balance float default 0,check (total>0),constraint fk4 foreign key(cust_ref_id) references Customer(Cust_id),constraint fk5 foreign key(booking_ref_id) references Booking(Booking_id));
- create table Payment_Points(cust_ref_id varchar2(10) primary key,points integer default 0,constraint fk6 foreign key(cust_ref_id) references Customer(Cust_id));
- create table Feedback(Feedback_id varchar2(10) primary key, cust_ref_id varchar2(10),feedback_cust varchar2(100) not null,constraint fk7 foreign key(cust_ref_id) references Customer(Cust_id));
- create table Query(Query_id varchar2(10) primary key, cust_ref_id varchar2(10),cust_query varchar2(100) not null,reply_admin varchar2(100) not null,constraint fk8 foreign key(cust_ref_id) references Customer(Cust_id));

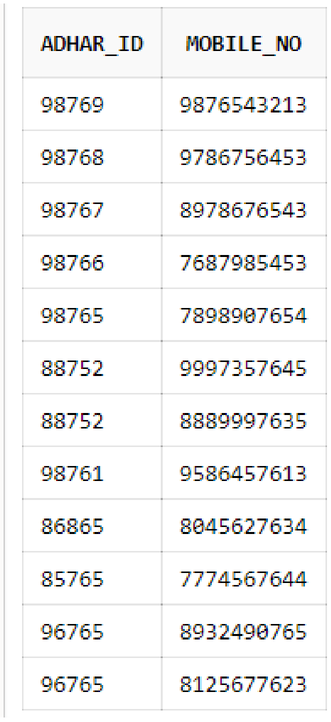
Sample Instances of the tables:

Customer:



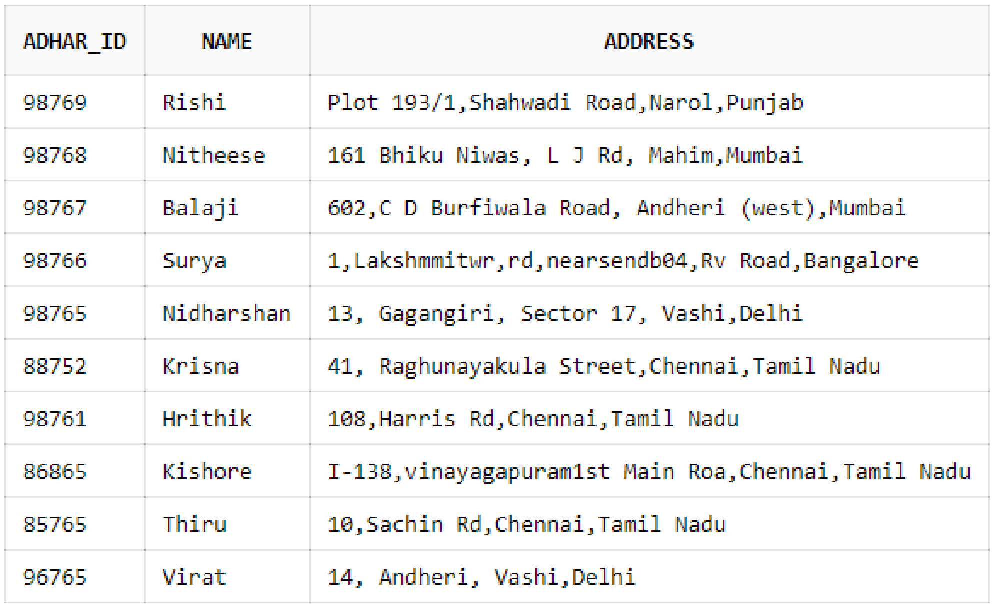
CUST_ID	ADHAR_ID	EMAIL_ID	USERNAME	PASSWORD	POINTS
C01	98769	rishirishi@gmail.com	rishi__12	rishiisgod	10
C02	98768	nitheese45@gmail.com	nitheese	nitheese456	50
C03	98767	balajidass@gmail.com	balaji007	balsbalu	30
C04	98766	surya45@gmail.com	surya_rohit	suryahere	40
C05	98765	nidhuraina@gmail.com	nidharshan	rainafanboy	20
C06	88752	krishna@gmail.com	krishna	#1edddddd	40
C07	98761	hrithikro@yahoo.com	hirubhai	statebank	70
C08	86865	kishore@gmail.com	kk07	sachinaaa	35
C09	85765	thirumalai@yahoo.com	thiru	kedarrocks	20
C10	96765	viratkohli@yahoo.com	cheeku	dhonirocks	60

Customer_Mobile:



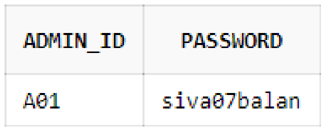
ADHAR_ID	MOBILE_NO
98769	9876543213
98768	9786756453
98767	8978676543
98766	7687985453
98765	7898907654
88752	9997357645
88752	8889997635
98761	9586457613
86865	8045627634
85765	7774567644
96765	8932490765
96765	8125677623

Customer_Person:



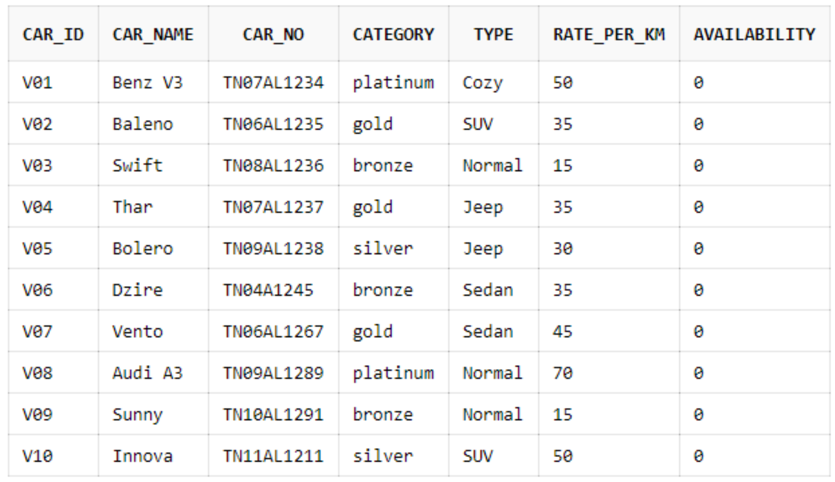
ADHAR_ID	NAME	ADDRESS
98769	Rishi	Plot 193/1,Shahwadi Road,Narol,Punjab
98768	Nitheese	161 Bhiku Niwas, L J Rd, Mahim,Mumbai
98767	Balaji	602,C D Burfiwala Road, Andheri (west),Mumbai
98766	Surya	1,Lakshmmitwr,rd,nearsendb04,Rv Road,Bangalore
98765	Nidharshan	13, Gagangiri, Sector 17, Vashi,Delhi
88752	Krisna	41, Raghunayakula Street,Chennai,Tamil Nadu
98761	Hrithik	108,Harris Rd,Chennai,Tamil Nadu
86865	Kishore	I-138,vinayagapuram1st Main Roa,Chennai,Tamil Nadu
85765	Thiru	10,Sachin Rd,Chennai,Tamil Nadu
96765	Virat	14, Andheri, Vashi,Delhi

Admin:



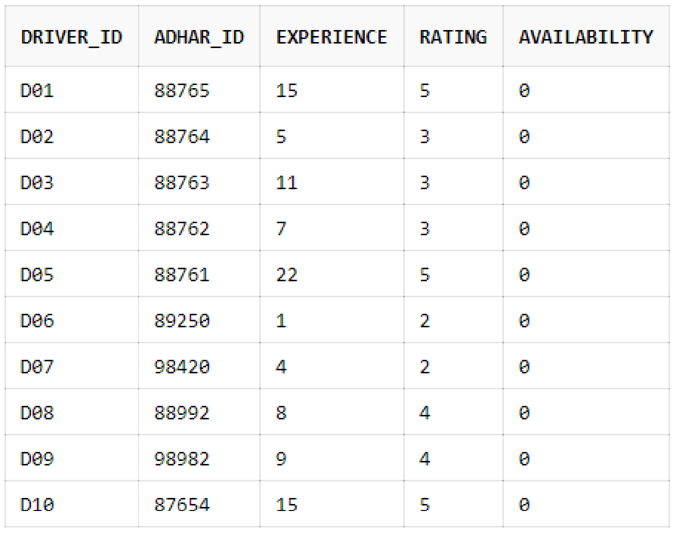
ADMIN_ID	PASSWORD
A01	siva07balan

Car:



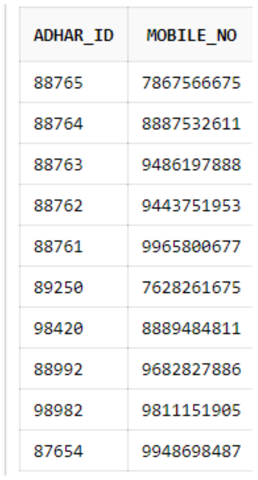
CAR_ID	CAR_NAME	CAR_NO	CATEGORY	TYPE	RATE_PER_KM	AVAILABILITY
V01	Benz V3	TN07AL1234	platinum	Cozy	50	0
V02	Baleno	TN06AL1235	gold	SUV	35	0
V03	Swift	TN08AL1236	bronze	Normal	15	0
V04	Thar	TN07AL1237	gold	Jeep	35	0
V05	Bolero	TN09AL1238	silver	Jeep	30	0
V06	Dzire	TN04A1245	bronze	Sedan	35	0
V07	Vento	TN06AL1267	gold	Sedan	45	0
V08	Audi A3	TN09AL1289	platinum	Normal	70	0
V09	Sunny	TN10AL1291	bronze	Normal	15	0
V10	Innova	TN11AL1211	silver	SUV	50	0

Driver:



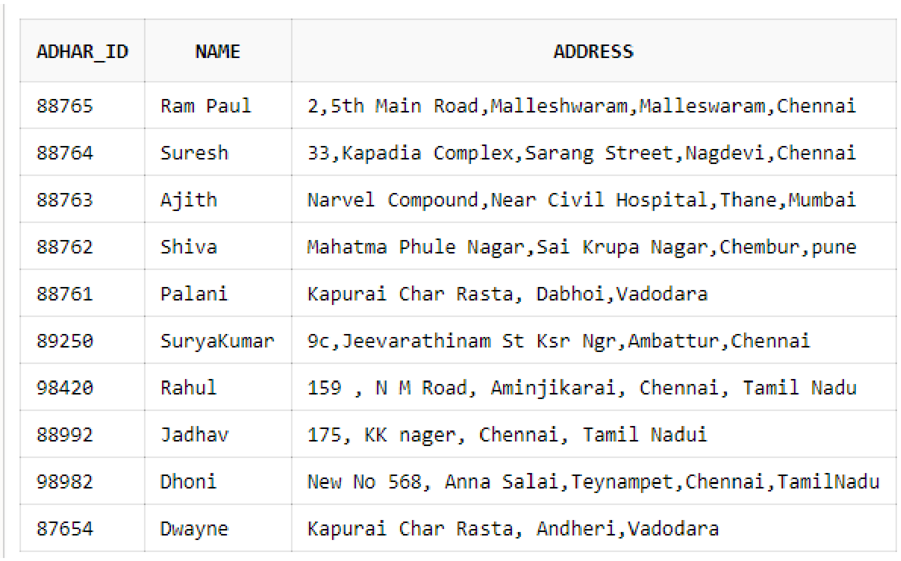
DRIVER_ID	ADHAR_ID	EXPERIENCE	RATING	AVAILABILITY
D01	88765	15	5	0
D02	88764	5	3	0
D03	88763	11	3	0
D04	88762	7	3	0
D05	88761	22	5	0
D06	89250	1	2	0
D07	98420	4	2	0
D08	88992	8	4	0
D09	98982	9	4	0
D10	87654	15	5	0

Driver_Mobile:



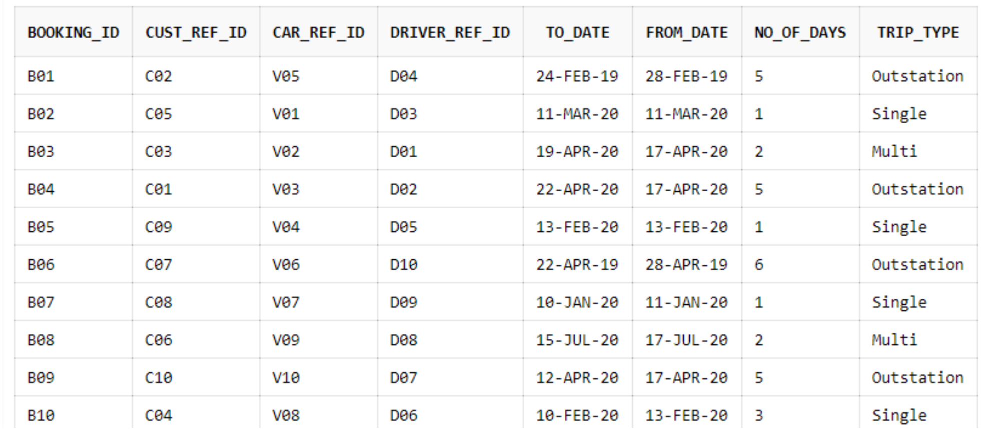
ADHAR_ID	MOBILE_NO
88765	7867566675
88764	8887532611
88763	9486197888
88762	9443751953
88761	9965800677
89250	7628261675
98420	8889484811
88992	9682827886
98982	9811151905
87654	9948698487

Driver Person:



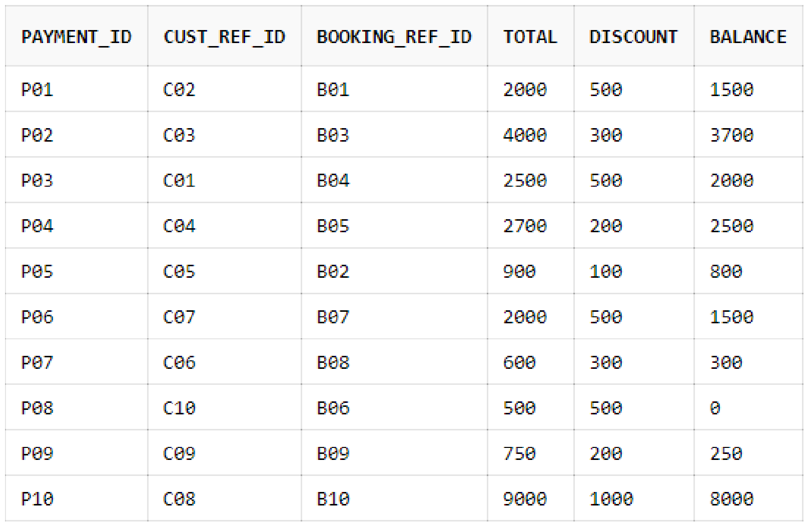
ADHAR_ID	NAME	ADDRESS
88765	Ram Paul	2,5th Main Road,Malleshwaram,Malleswaram,Chennai
88764	Suresh	33,Kapadia Complex,Sarang Street,Nagdevi,Chennai
88763	Ajith	Narvel Compound,Near Civil Hospital,Thane,Mumbai
88762	Shiva	Mahatma Phule Nagar,Sai Krupa Nagar,Chembur,pune
88761	Palani	Kapurai Char Rasta, Dabhoi,Vadodara
89250	SuryaKumar	9c,Jeevarathinam St Ksr Ngr,Ambattur,Chennai
98420	Rahul	159 , N M Road, Aminjikarai, Chennai, Tamil Nadu
88992	Jadhav	175, KK nager, Chennai, Tamil Nadui
98982	Dhoni	New No 568, Anna Salai,Teynampet,Chennai,TamilNadu
87654	Dwayne	Kapurai Char Rasta, Andheri,Vadodara

Booking:



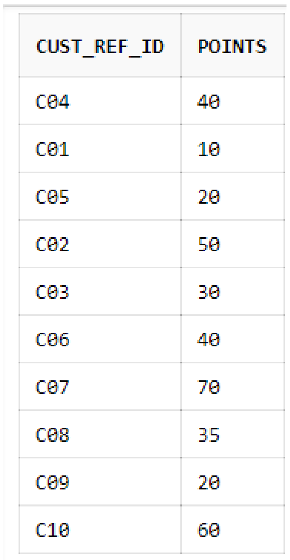
BOOKING_ID	CUST_REF_ID	CAR_REF_ID	DRIVER_REF_ID	TO_DATE	FROM_DATE	NO_OF_DAYS	TRIP_TYPE
B01	C02	V05	D04	24-FEB-19	28-FEB-19	5	Outstation
B02	C05	V01	D03	11-MAR-20	11-MAR-20	1	Single
B03	C03	V02	D01	19-APR-20	17-APR-20	2	Multi
B04	C01	V03	D02	22-APR-20	17-APR-20	5	Outstation
B05	C09	V04	D05	13-FEB-20	13-FEB-20	1	Single
B06	C07	V06	D10	22-APR-19	28-APR-19	6	Outstation
B07	C08	V07	D09	10-JAN-20	11-JAN-20	1	Single
B08	C06	V09	D08	15-JUL-20	17-JUL-20	2	Multi
B09	C10	V10	D07	12-APR-20	17-APR-20	5	Outstation
B10	C04	V08	D06	10-FEB-20	13-FEB-20	3	Single

Payment:



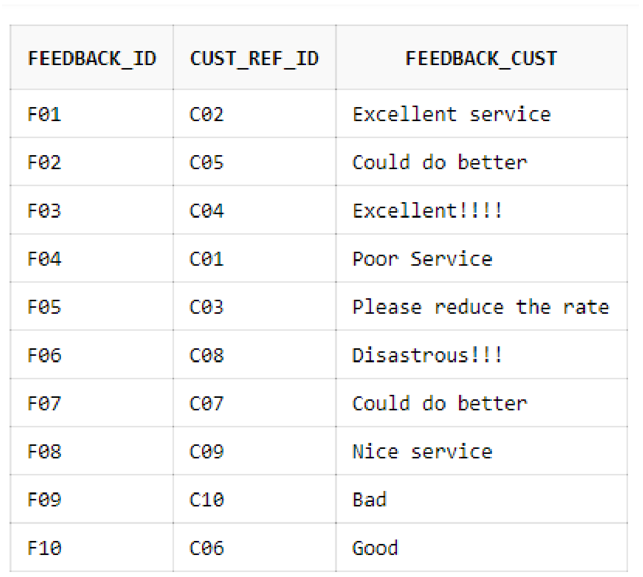
PAYMENT_ID	CUST_REF_ID	BOOKING_REF_ID	TOTAL	DISCOUNT	BALANCE
P01	C02	B01	2000	500	1500
P02	C03	B03	4000	300	3700
P03	C01	B04	2500	500	2000
P04	C04	B05	2700	200	2500
P05	C05	B02	900	100	800
P06	C07	B07	2000	500	1500
P07	C06	B08	600	300	300
P08	C10	B06	500	500	0
P09	C09	B09	750	200	250
P10	C08	B10	9000	1000	8000

Payment_Points:



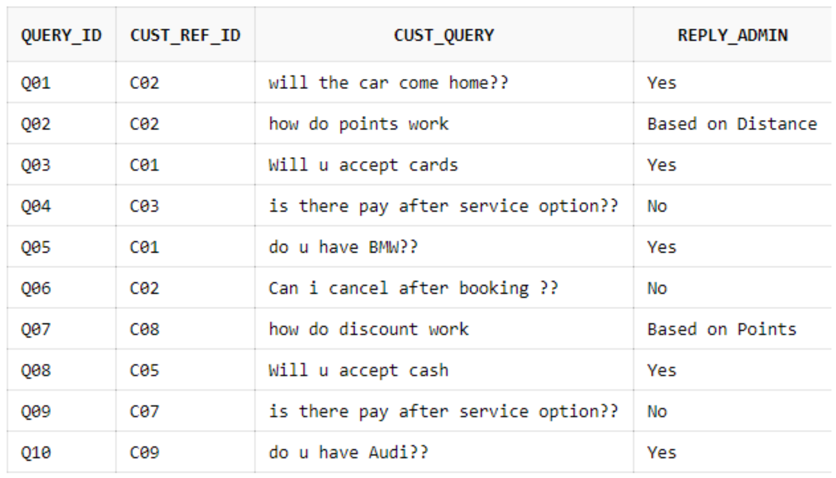
CUST_REF_ID	POINTS
C04	40
C01	10
C05	20
C02	50
C03	30
C06	40
C07	70
C08	35
C09	20
C10	60

Feedback:



FEEDBACK_ID	CUST_REF_ID	FEEDBACK_CUST
F01	C02	Excellent service
F02	C05	Could do better
F03	C04	Excellent!!!!
F04	C01	Poor Service
F05	C03	Please reduce the rate
F06	C08	Disastrous!!!
F07	C07	Could do better
F08	C09	Nice service
F09	C10	Bad
F10	C06	Good

Query:



QUERY_ID	CUST_REF_ID	CUST_QUERY	REPLY_ADMIN
Q01	C02	will the car come home??	Yes
Q02	C02	how do points work	Based on Distance
Q03	C01	Will u accept cards	Yes
Q04	C03	is there pay after service option??	No
Q05	C01	do u have BMW??	Yes
Q06	C02	Can i cancel after booking ??	No
Q07	C08	how do discount work	Based on Points
Q08	C05	Will u accept cash	Yes
Q09	C07	is there pay after service option??	No
Q10	C09	do u have Audi??	Yes

8.Frontend design:

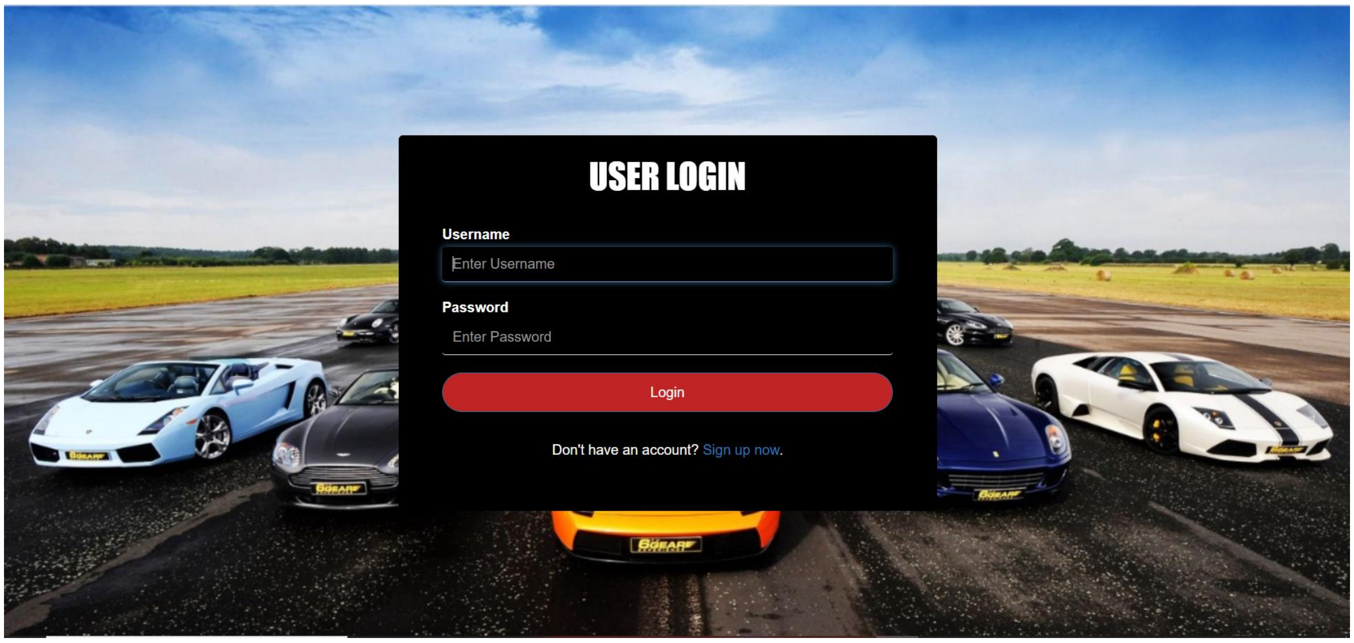
Introduction to the tools used in the project:

- **HTML5:**
  - HTML5 is a markup language used for structuring and presenting content on the World Wide Web.
- **CSS3:**
  - Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML.
- **JAVASCRIPT:**
  - JavaScript, often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm.
- **PHP:**
  - PHP is a general-purpose scripting language especially suited to web development. It was originally created by Danish-Canadian programmer Rasmus Lerdorf in 1994.

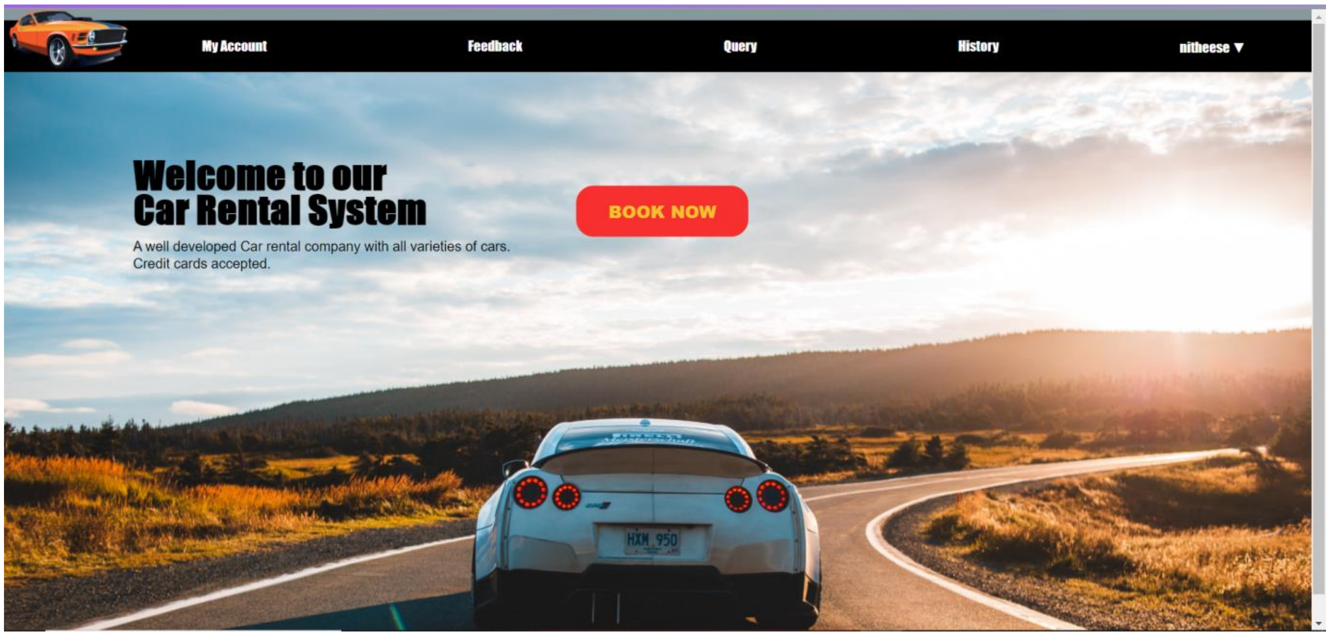
Screenshots of UI:

CUSTOMER:

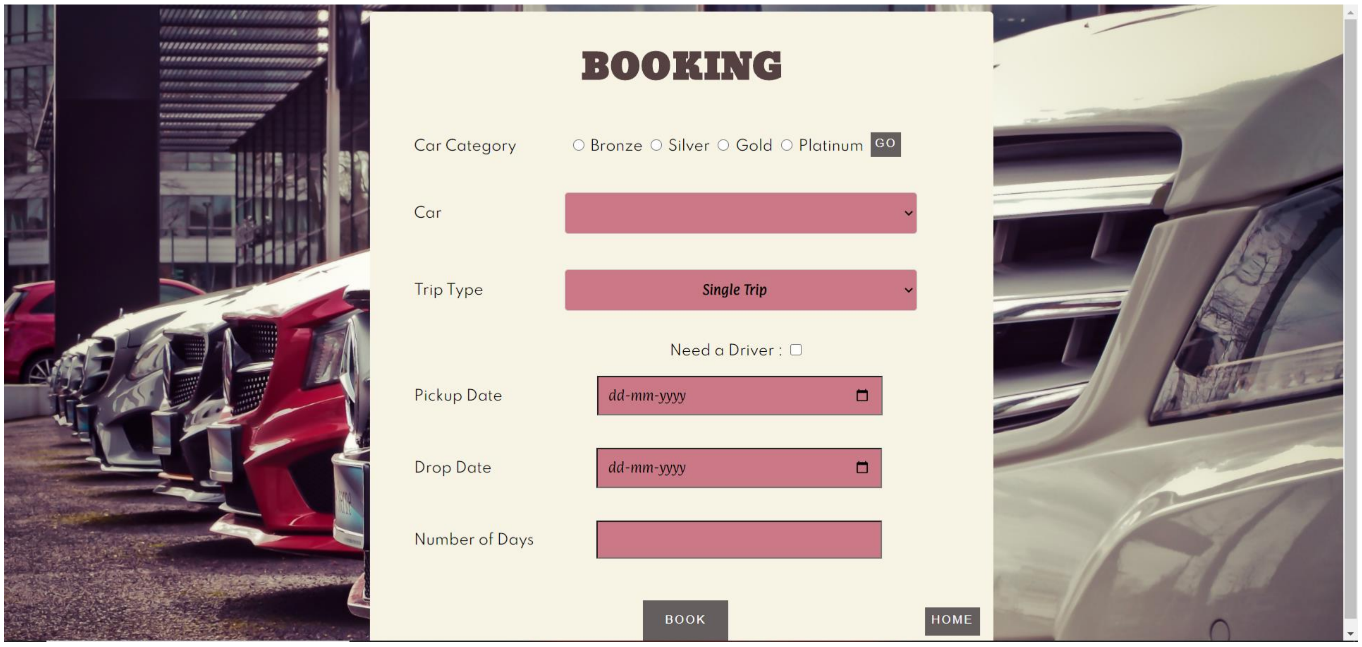
LOGIN:



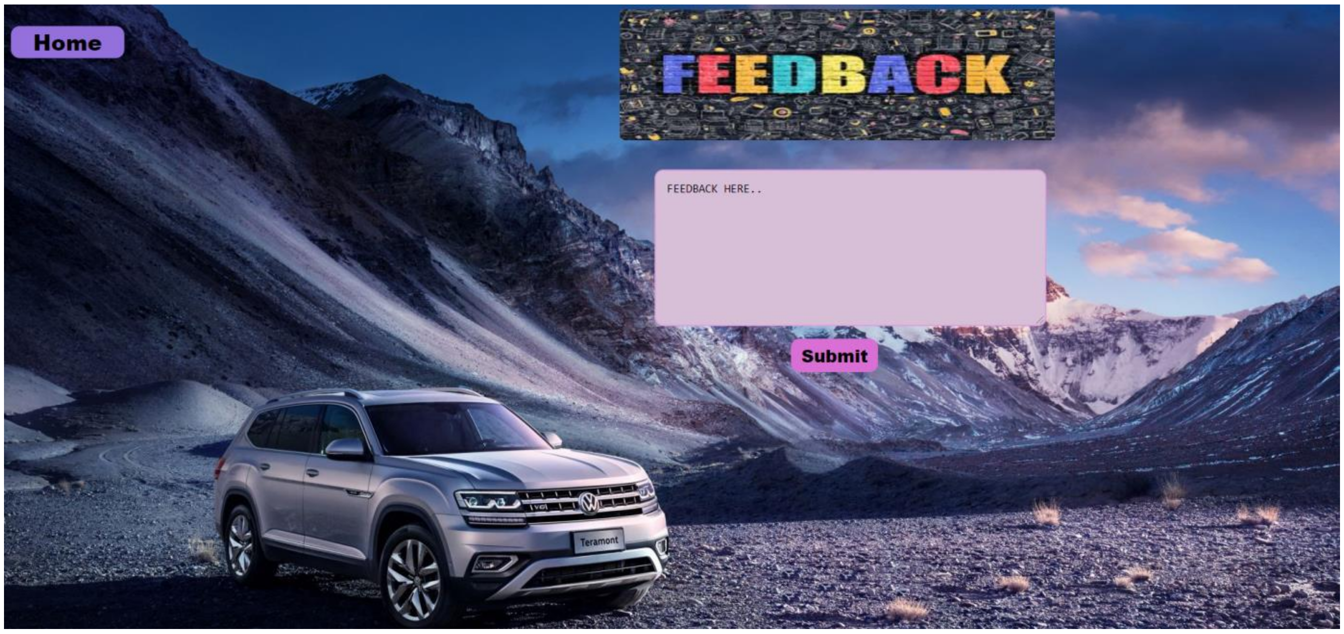
HOMEPAGE:



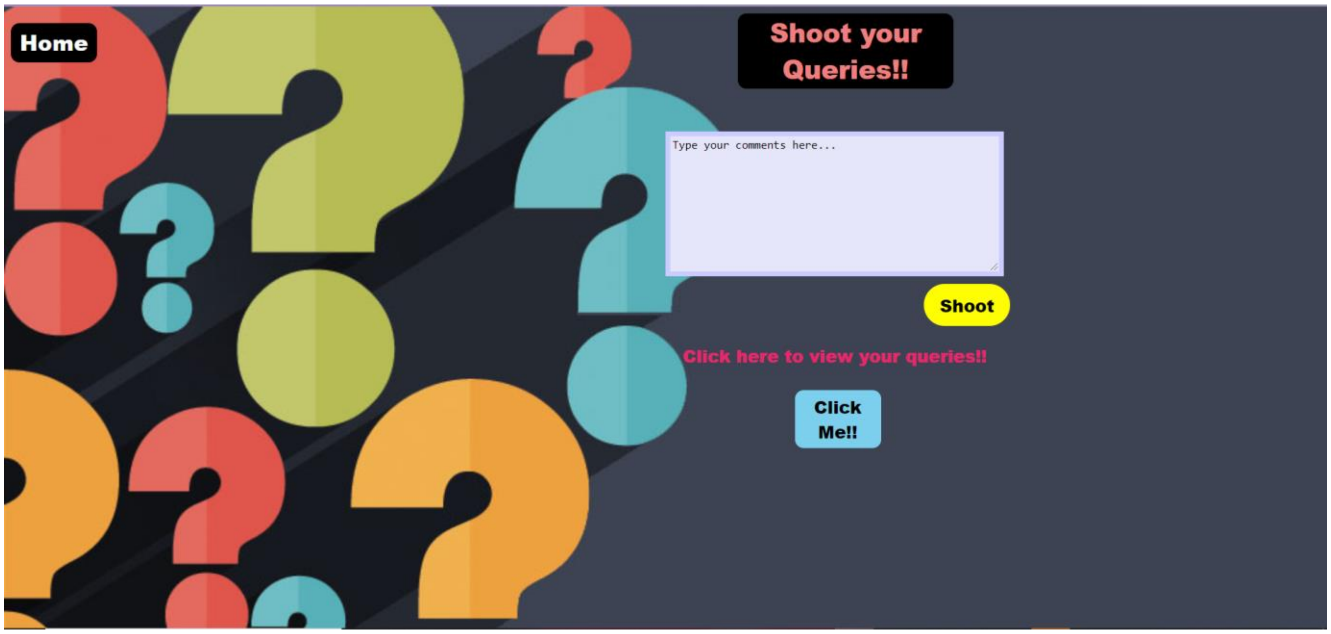
BOOKING:



FEEDBACK:



QUERY:



9.Database Connectivity:

Introduction to the connectivity standard:

We have used PHP as a tool to connect frond with frontend and backend. XAMPP was used to it. We created a database called **‘dbdb’** and corresponding tables in localhost/myphpadmin. We connected our frontend with the database using a file called ‘db_connection.php’ which is given below. This file we included in all other files of our project.

Connectivity Code:

```
<?php
function OpenCon()

 {$dbhost = "localhost";

 $dbuser = "root";

 $dbpass = "";

 $db = "dbdb";

 $conn = new mysqli($dbhost, $dbuser, $dbpass,$db) or die("Connect failed: %s\n". $conn -
> error);

 return $conn;}

function CloseCon($conn)

 {$conn -> close();}

?>
```

10.Sample Codes:

Homepage:

```php
<?php
    include 'db_connection.php';

    session_start();

    // Check if the user is logged in, if not then redirect him to login page

        if(!isset($_SESSION["loggedin"]) || $_SESSION["loggedin"] !== true){

        header("location: login.php");

        exit;}

        //Checking if Cars are available

        $conn = OpenCon();

        $cars_list = $conn->query("SELECT Car_name from car where availability=0");

        $user = $_SESSION['username'];

        $val = 0;

        if(mysqli_num_rows($cars_list)==0){

            $val = 1;

        }



        $val1 = 0;

        $id = $_SESSION['id'];

        $sql7 = "SELECT booking.Booking_id,booking.from_date,booking.to_date,car.Car_name,booking.driver_ref_id,payment.total,payment.balance,payment.discount_applied FROM booking INNER JOIN payment on booking.Booking_id=payment.booking_ref_id INNER JOIN car on booking.car_ref_id=car.Car_id where booking.cust_ref_id='$id'";

        $past_books = $conn->query($sql7);

        if(mysqli_num_rows($past_books)==0){

            $val1 = 1;
        }
        CloseCon($conn);
?>
    <html>
    <head>
        <meta charset="utf-8">
        <title>Home page</title>
        <link rel="stylesheet" type="text/css" href="css/homepage.css">
        <script type="text/javascript">
            function booking(){
                var ct = "<?php echo $val ?>";
                if(ct==1){
                  alert("Sorry! No cars are available right now");}
                else{ window.location.href = "book.php";}}
            function past(){
                var ct = "<?php echo $val1 ?>";
                if(ct==1){
                  alert("You have not made any bookings with us before !!");
                }
                else{
                  window.location.href = "history.php";}}</script>
    </head>
    <body>
        <img src="images/logo.png" alt="Preview not avaliable" id="logo">
        <header>
            <nav>
                <a href="account.php" id="account">My Account</a>
                <a href="feedback.php" id="feedback">Feedback</a>

                <a href="query.php" id="query">Query</a>

                <a id="history" onclick=past();>History</a>

                <div class="dropdown">

                <button class="dropbtn"><?php echo $user ?> &#9660;</button>

                        <div class="dropdown-content">

                        <a href="logout.php">Logout</a>

                </div>

            </nav>

        </header>

        <div class="main">

       <section class="left">

           <h1 id="h1">Welcome to our </h1>

           <h1 id="h2">Car Rental System</h1>

           <p id="p1">A well developed Car rental company with all varieties of cars.</p>

           <br><br>

           <button class="button" onclick=booking();><span>BOOK NOW</span></button>

           <p id="p2">Credit cards accepted.</p>

       </section>

       </div>

    </body>

</html>
```

Booking:

```
<?php
    session_start();

    // Check if the user is logged in, if not then redirect him to login page
```

```
        if(!isset($_SESSION["loggedin"]) || $_SESSION["loggedin"] !== true){

        header("location: login.php");

        exit;}

?>


<!DOCTYPE html>

<html>

<head>

  <title>Booking</title>

  <link rel="stylesheet" type="text/css" href="css/booking.css?v=<?php echo time(); ?>">

  <script type="text/javascript">

    //No_of_days function

    function calculate() {

      var date1 = new Date(document.getElementById('from_date').value);

      var date2 = new Date(document.getElementById('to_date').value);

      var Difference_In_Time = date2.getTime() - date1.getTime();

      var Difference_In_Days = Difference_In_Time / (1000 * 3600 * 24);

      document.getElementById('no_of_days').value = parseInt(Difference_In_Days);

      }


      //Show Driver list

      function showdriver(show_driver) {

        var show = document.getElementById("driver_select");

        show.style.display = show_driver.checked ? "block" : "none";

        }


    //Booking Successfull message
    function success() {

        alert("Booking Successfull !!");

      }



   </script>


  <!--Database Connection-->

  <?php

    include 'db_connection.php';

    ?>

</head>

<!--<header>

  <img src="images/book_logo.png">

</header>-->

<body>



  <div class="container">

    <form method="post"  target="_self" >



    <h1><p>BOOKING</p></h1>



    <br><br>

  <div class="category">

  <form method="post">

  <label for="category">Car Category</label>

  <input type="radio" id="bronze" value="bronze" name="category" >

  <label for="bronze">Bronze</label>
  <input type="radio" id="silver"  value="silver" name="category" >

  <label for="silver">Silver</label>

  <input type="radio" id="gold"  value="gold" name="category">

  <label for="gold">Gold</label>

  <input type="radio" id="platinum"  value="platinum" name="category">

  <label for="gold">Platinum</label>

  <input type="submit" name="go" value="GO" id="go_button">

  <form>

    </div>

    <br><br>

  <div class="car">

    <label for="car">Car</label>

    <select id="car" name="car">


    <!--CAR SELECT-->

    <?php

    $conn = OpenCon();

    if(isset($_POST['go'])){

    $k = $_POST['category'];

    $records = $conn-
>query("SELECT Car_name FROM Car where category='$k' AND availability=0");

    while ($car = mysqli_fetch_assoc($records)){

          echo "<option>".$car['Car_name']."</option>";

      }}

    CloseCon($conn);

    ?>
    </select>

  </div>

  <br>

  <div class="trip_type">

    <label for="trip">Trip Type</label>

    <select id="trip_type" name="trip_type">

      <option value="Single">Single Trip</option>

      <option value="Round">Round Trip</option>

      <option value="Outstation">Outstation</option>

    </select>

  </div>



  <br>

  <div class="need_driver">

      <label for="need_driver">Need a Driver : </label>

      <input type="checkbox" name="need_driver" id="need_driver" onclick="showdriver(this)"/>

  </div>

  <div id="driver_select" style="display: none">

      <label for="driver">Driver</label>

      <select id="driver" name="driver" style="margin-left: 115px">

      <option value=""></option>



      <!--DRIVER SELECT-->

      <?php

      $conn = OpenCon();
```

```
      $drivers = $conn-
>query("SELECT driver_person.Name,driver.rating FROM driver NATURAL JOIN driver_person WHERE d
river.availability=0");

      while ($driver = mysqli_fetch_assoc($drivers)){

          echo "<option>".$driver['Name']."---".$driver['rating'].'/5'."</option>";

      }

      CloseCon($conn);

      ?>

      </select></div>

<br>

  <div class="from_date">

    <label for="from_date">Pickup Date</label>

    <input type="date" name="from_date"  id="from_date" min='27/10/2020' max='27/02/2021'>

  </div>



    <br><br>

    <div class="to_date">

    <label for="to_date">Drop Date</label>

    <input type="date" name="to_date" id="to_date" onchange="calculate()" min='27/10/2020' max
='27/02/2021' >

    </div>



    <br><br>

    <div class="no_of_days">

      <label for_numberofdays>Number of Days</label>

      <input type="text" name="no_of_days" id="no_of_days">

    </div>
```

```
<br><br>


<br>


<div class="book">

  <input type="submit" name="BOOK" value="BOOK">

</div>



  <!--INSERT IN BOOKING-->

  <?php

    $conn = OpenCon();

  if(isset($_POST['BOOK'])){

    $var1 = $_POST['car'];

    $var2 = $_POST['driver'];

    $var2 = strtok($var2, '-');

    $cu = $_SESSION["id"];


    //Car ID

    $result = $conn->query("SELECT car_id from car WHERE car_name='$var1'  ");

    while ($X = mysqli_fetch_assoc($result)){

      $c_id = $X['car_id'];

    }


    //Driver ID

    if ($_POST['driver'] === '')

    {

        $_POST['driver'] = 'NULL';
      }

      else{

      $result = $conn-
>query("SELECT Driver_id from driver where Adhar_id=(SELECT Adhar_id from driver_person where
Name='$var2') ");

      while ($X = mysqli_fetch_assoc($result)){

        $d_id = $X['Driver_id'];

      }}


      //Booking Id
      $b_id = null;

      $row = $conn-
>query("SELECT Booking_id FROM booking ORDER BY Booking_id DESC LIMIT 1; ");

      while ($X = mysqli_fetch_assoc($row)){

        $b_id = $X['Booking_id'];

      }

      if($b_id == null){

        $b_id ='B01';

      }

      else{

      $bid_num = (int) filter_var($b_id, FILTER_SANITIZE_NUMBER_INT);

      $bid_num = $bid_num+1;

      if($bid_num<10){

      $b_id = 'B0'.$bid_num;

      }

      else{

        $b_id = 'B'.$bid_num;

      }}
      //Insert into booking

      $sql = "INSERT INTO booking VALUES(?,?,?,?,?,?,?,?)";

      $stmt = mysqli_prepare($conn,$sql);

      $stmt-
>bind_param("ssssssss",$b_id,$cu,$c_id,$d_id,$_POST['from_date'], $_POST['to_date'],$_POST['no
_of_days'],$_POST['trip_type']);


      if(mysqli_stmt_execute($stmt)){

      //Update availability of car

      $sql4 = "UPDATE car SET availability=1 where Car_id='$c_id'";

      $stmt1 = mysqli_prepare($conn,$sql4);

      $stmt1->execute();


      //Update availability of driver

      if($d_id != null){

      $sql4 = "UPDATE driver SET availability=1 where Driver_id='$d_id'";

      $stmt2 = mysqli_prepare($conn,$sql4);

      $stmt2->execute();}


      echo "<script type='text/javascript'>alert('Booked :) Remember this id :  '+'$b_id'+'  f
or payment');</script>";

      }


      CloseCon($conn);

    }


    ?>
```

```
  </form>

  <div>

  <button class="gohome" id="gohome" onclick="window.location.href='homepage.php';" ><span>HOM
E</span></button>

  <div>

</div>

</body>

</html>
```

Feedback:

```
<?php
    session_start();

    // Check if the user is logged in, if not then redirect him to login page

        if(!isset($_SESSION["loggedin"]) || $_SESSION["loggedin"] !== true){

        header("location: login.php");

        exit;}

?>



<!DOCTYPE html>
<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1">



<style>

* {

  box-sizing: border-box;
```

```
}


body{
  background-image: url("images/feedimg.jpg");
  background-repeat: no-repeat;
  background-size:cover;
}


form
{
margin-left: 800px;
width: 100px;
align:right;
}


header img{
  background-color: #FFFFFF;
  border-radius: 5px;
  width: 500px;
  height: 150px;
  margin-left: 700px;
}


#comments{
  background-color: #FFFFFF;
  border-radius: 5px;
  width: 50px;
  height: 100px;

  margin-top:100px;

  margin-left: 700px;

}


textarea {

  width: 450%;

  padding: 12px;

  border: 2px solid #DDA0DD;

  background-color:#D8BFD8;

  border-radius: 10px;

  resize: vertical;

  margin-left:-60px;

  margin-top:30px;

}


label {

  padding: 12px 12px 12px 0;

  display: inline-block;

}


.button1{
  padding-left: 5px;

  padding-top: 5px;

  padding-right: 5px;

  padding-bottom: 5px;

  border-radius: 10px;
```

```
    background-color: #DA70D6;

    border: none;

  color: #000000;

  margin-left:97px;

  margin-top:10px;

  width:100px;

  font-family: "Arial Black", Gadget, sans-serif;

    text-align: center;

    font-size: 20px;


}


.button1:hover{

  background-color:#000000;

  color: #ffffff;


}


.button2{
  padding-left: 0px;

  padding-right: 0px;

  border-radius: 10px;

    background-color: #9370DB;

    border: none;

  color: #000000;

  margin-left:0px;

  margin-top:-780px;
  width:130px;

  font-family: "Arial Black", Gadget, sans-serif;

    text-align: center;

    font-size: 25px;


}


.button2:hover{

  background-color:#DA70D6;

  color: #000000;

  font-size: 30px;

}



}
</style>



<!--Database Connection-->
  <?php

    include 'db_connection.php';

    ?>



</head>
<header>

<img src="images/feedhd.jpg">

</header>



<body>
```

```
<form method="post" target="_self">


  <textarea name="feed" rows="10" cols="10">FEEDBACK HERE..</textarea>


  <input type="submit" name="go" class="button1" value="Submit">

</form>

<button class="button2" onclick="window.location.href='homepage.php';" style="vertical-
align:middle"><span>Home </span></button>

    <?php

      $conn = OpenCon();

    if(isset($_POST['go'])){

      $sql = "INSERT INTO feedback VALUES(?,?,?)";

      $var1 = $_POST['feed'];


      $cu = $_SESSION["id"];


      //Feedback_id

      $f_id = null;

      $row = $conn-
>query("SELECT Feedback_id FROM feedback ORDER BY Feedback_id DESC LIMIT 1; ");

      while ($X = mysqli_fetch_assoc($row)){

        $f_id = $X['Feedback_id'];

      }

      if($f_id == null){

        $f_id ='F01';

      }

      else{
```

```
      $fid_num = (int) filter_var($f_id, FILTER_SANITIZE_NUMBER_INT);

      $fid_num = $fid_num+1;

      if($fid_num<10){

      $f_id = 'F0'.$fid_num;

      }

      else{

        $f_id = 'F'.$fid_num;

      }}


      $stmt = mysqli_prepare($conn,$sql);

      $stmt->bind_param("sss",$f_id,$cu,$_POST['feed']);

      $stmt->execute();

      CloseCon($conn);

    }

    ?>

</body>
</html>
```

11.Conclusions:

This project is being considered in order to reduce and eliminate loss of customers competitors, and save the company from folding up. The current system is manual and it is time consuming. It is also cost ineffective, and average return is low and diminishing. Currently, customers can call or walk-in in order to rent or reserve a vehicle. The staff of the company will check their file to see which vehicle is available for rental. The current system is error prone and customers are dissatisfied. The goal of this project is to automate vehicle rental and reservation so that customers do not need to walk-in or call in order to reserve a vehicle. They can go online and reserve any kind of vehicle they want from the inventory of available vehicles. Even when a customer chooses to walk-in, computers are available for him to go online and perform his reservation. When he choose to reserve by phone, any of the customer service representatives can help him reserve the vehicle speedily and issue him a reservation number.

12.References

[1] Abraham Silberschatz ,Henry F. Korth ,S. Sudarshan, Database System Concepts.

7th ed. New York : McGraw-Hill Education, 2020