KPRC RENTAL SYSTEM

FINAL PROJECT REPORT

Student ID: kap675, rrc370
Student Name: Kundan Patel, Rutwik Chinchole
Course: CS GY 6083(Principles of Database Systems)
Section: B
Date of submission: 12/20/2020

Table of Contents

Introduction

We have chosen to produce a Rental system. In our system, customers can rent a vehicle based on make and model. Our system provides customers to have different pick-up and drop- off locations and will impose late fee if the rental vehicle is returned beyond the return date and time. The Customers can use upto one discount coupon to their final bill. Customers who have a Corporate account will be by default given a certain fixed % discount in their final invoice

Executive Summary

Rental company KPRC. In the United States, KPRC has several locations at different airports, towns and cities. KPRC plans to transform its file system database data management to a sophisticated centralized database system for business creation and effective management of its business. The following information and rules about the organization were given by the KPRC business team.

- Each KPRC office location maintains different groups of vehicles for leasing. Make, model, year, VIN (Vehicle Identification Number), and License Plate number are identified for each vehicle.
- Each location has different vehicle classes, such as small cars, medium-sized cars, luxury cars, SUVs, premium SUVs, mini vanes, station wagons, etc. The class has its own rental rate and over-mileage charges per day of the rental service (if rental service reaches odometer limits/day). A mid-size car rental service, for instance, has a regular service rate of day and mileage fees of over $2/mile. An odometer maximum of 500 miles/day when a customer has rental service for 2 days.
- The rental service has a total limit of 1000 miles. If 1050 miles have been used by this rental service, then customers will be paid 2 days * $ 40 + $ 2 * 50 additional miles, totaling $ 180.
- KPRC has individual or corporate-type clients. With the full address, email address and phone number of the customer, KPRC maintains a list of their customers. For each customer, KPRC keeps just one address.
- For individual customers, KPRC stores Full Customer Name, Driver License Number, Name of Insurance Provider
- KPRC retains records of the company's name and identification number and the employee ID of the customer renting the car on a corporate account for corporate customers.
- KPRC sends discount coupons to its clients and even sends discount coupons to neighborhood residents. KPRC gives discounts to its individual customers who at the time of renting a car, carry in such a voucher. These discount coupons bear dates of validity and percentage of discount offered, such as a 5 percent discount valid from 10/01/2020 to 10/31/2020.

- For affiliated businesses, KPRC offers discounts to corporate customers on fixed corporate discount sets, and discounts vary from company to corporation. This discount is given to corporate customers regardless of the rental service date.
- KPRC offers only one form of discount at a time for the rental service (individual discount coupon or corporate discount) and does not offer the same rental service for both discounts.
- KPRC lists Pickup Location, Drop off Location, Pickup Date, Drop off Date, Start Odometer, End Odometer, Regular Odometer Cap for the rental service for each rental service. Some rental facilities have unlimited mileage.
- KPRC raises an invoice with an invoice date and invoice sum for each rental service given.
- KPRC requires clients to use various ways (credit/debit) to pay an invoice.
- KPRC maintains records of its full address and phone number for each rental office location.

We have addressed all the possible business cases in our system and created a brand new Rental system which will make it easier for the Employees for the KPRC Rental Company to manage and also make it easier for the Customers to use the application

We have created separate dashboards for both the customer and employee of the company using which they can perform all the activities with a click of button. The employees can add vehicle data, class data, location data and send coupons to the customers as well as view the customers in the current system easily and manipulate the data in the system with much more flexibility

Only the employee can access the sensitive parts of the system and we have placed appropriate security measures to make sure that no parts of the database can be used by any unauthorized user. Creating the employee dashboard has proven to be a big boon for the company as they will be able to perform all the actions from one place itself

The customer on the other hand would have to sign up and provide their credentials to login, again we have placed password hashing and proper authentication mechanisms to ensure data integrity and confidentiality in our system. The customer also has a dashboard of its own where they can Start a trip, view all their bookings and End Trip/Generate Invoice. All these activities are interlinked and work together in harmony as we have placed proper Triggers and Procedures in our code to make sure the data that the customer sees is dynamically updated and is always up to date.

The main use case of the system i.e to Rent a vehicle has been simplified to its root level in our system and will simplify the life of the Customers and make renting a cherishable experience for them.

Details of Software language and database used

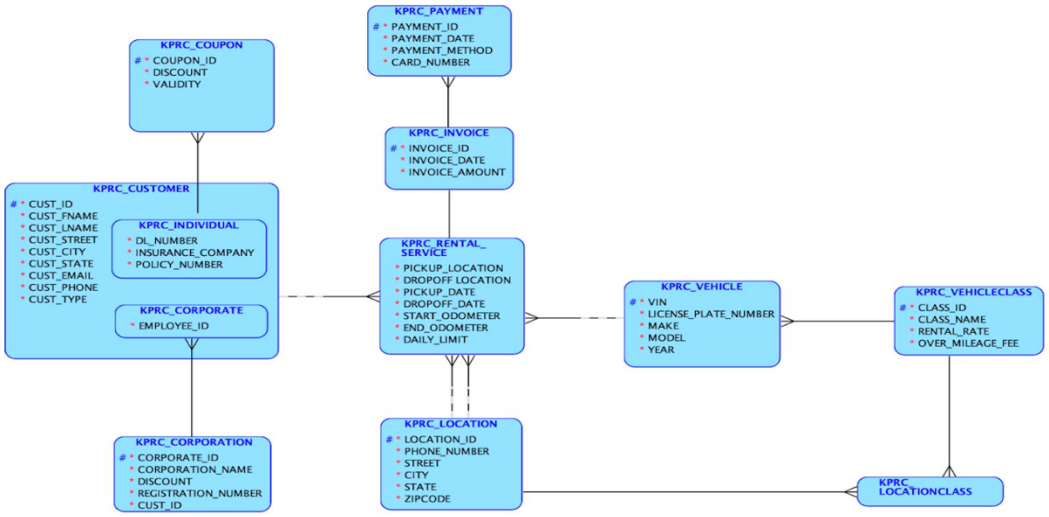
We have used Django as the web framework for our application and used MySQL as the database.

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. It is built by experienced developers and it takes care of much of the hassle of Web development, so we can focus on writing the app without needing to reinvent the wheel. It's free and open source.

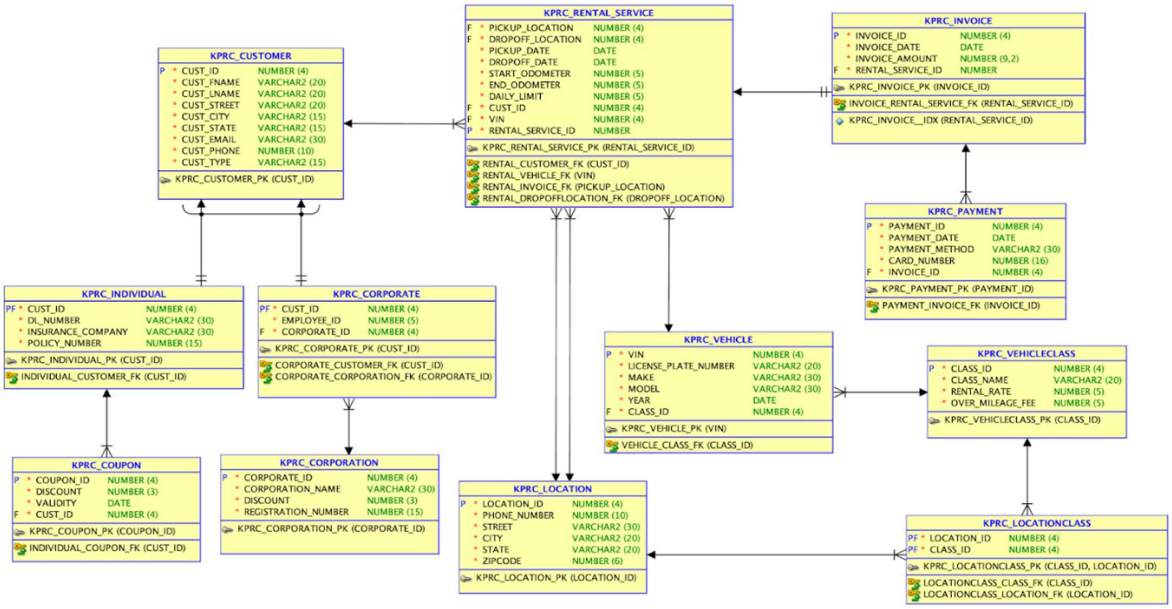
MySQL Database Service is a fully managed database service to deploy cloud-native applications. HeatWave, an integrated, high-performance analytics engine accelerates MySQL performance by 400x.

ER diagram

A. Logical Model



B. Relational Model



Tables

a) **kprc_customer**:

Attributes: cust_id, cust_fname, cust_lname, cust_street, cust_city, cust_state, cust_email, cust_phone, cust_type

Total Records: 10

b) **kprc_corporate**:

Attributes: cust_id, employee_id, corporate_id

Total Records: 5

c) **kprc_individual**:

Attributes: cust_id, dl_number, insurance_company, policy_number

Total Records: 5

d) **kprc_corporation**:
Attributes: corporate_id, corporation_name, discount, registration_number
Total Records: 5

e) **kprc_coupon**:
Attributes: coupon_id, discount, validity, cust_id
Total Records: 6

f) **kprc_invoice**:
Attributes: invoice_id, invoice_date, invoice_amount, rental_service_id
Total Records: 8

g) **kprc_location**:
Attributes: location_id, phone_number, street, city, state, zipcode
Total Records: 6

h) **kprc_locationclass**:
Attributes: location_id, class_id
Total Records: 10

i) **kprc_payment**:
Attributes: payment_id, payment_date, payment_method, card_number, invoice_id
Total Records: 8

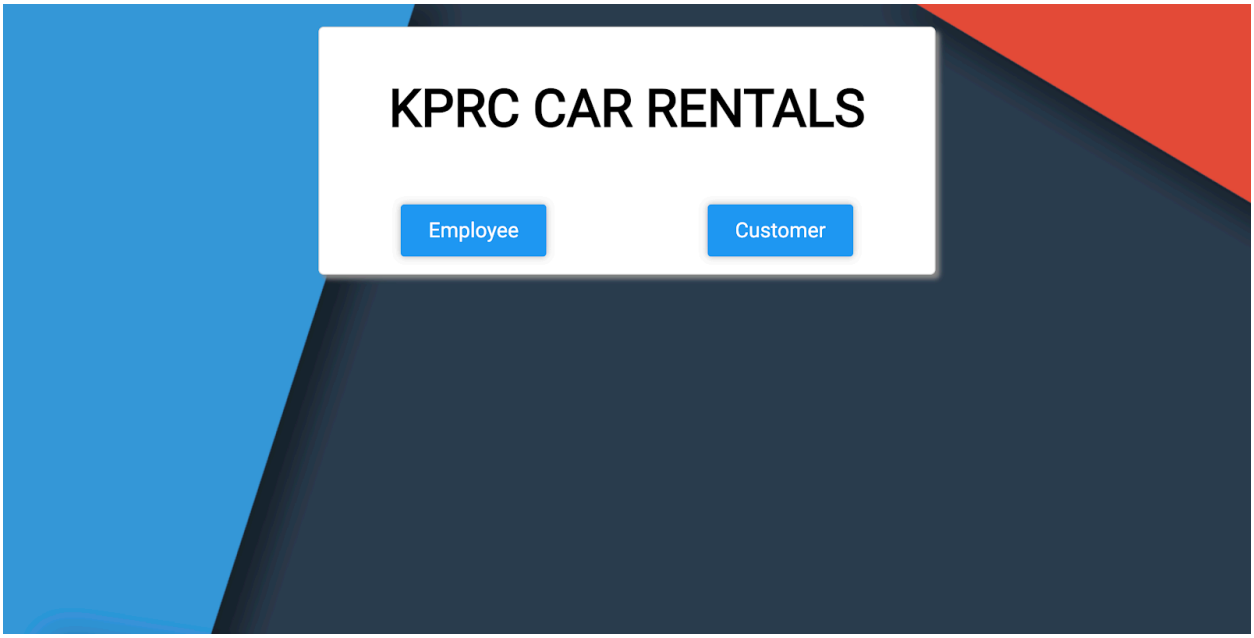
j) **kprc_rental_service**:
Attributes: pickup_location, dropoff_location, pickup_date, dropoff_date, start_odometer, end_odometer, daily_limit, cust_id, vin, rental_service_id
Total Records: 8

k) **kprc_vehicle**:
Attributes: vin, license_plate_number, make, model, year, class_id
Total Records: 6

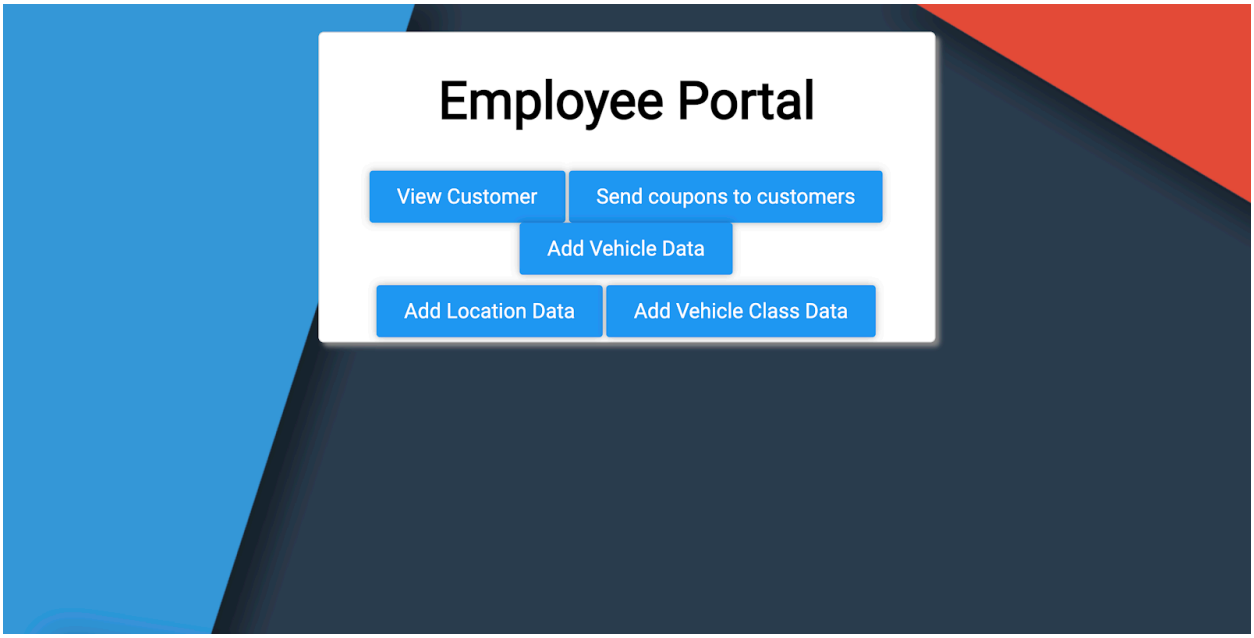
l) **kprc_vehicleclass**:
Attributes: class_id, class_name, rental_rate, over_mileage_fee
Total Records: 5

Overall flow of the application

Home Page:



Employee portal:



View Customer:

Customer View

ID	First Name	last Name	Street	City	State	Email	Phone	Type
1	demo	demo	32	fsdf	shfd	dhfs@gmail	241525235	kprc_individual
4	dds	gdg	w	sd	dfsd	dsgds	341	kprc_corporate
5	dds	gdg	w	sd	dfsd	dsgds	341	kprc_individual
6	dds	gdg	w	sd	dfsd	fgd	341	kprc_individual
7	Harry	Potter	Baker street	London	London	harry@potter.com	432523153	kprc_corporate
8	Joe	Hardy	43	New york	New York	joe@hardy.com	653786525	kprc_individual
9	kap	kap	434	jjghk	gfjdskjg	kap@kap	3852	kprc_individual
10	kap	kfd	4234	hghdh	hfgh	kap@kprc	64823652	kprc_individual

Send coupons to Customer:

Send coupon to customer

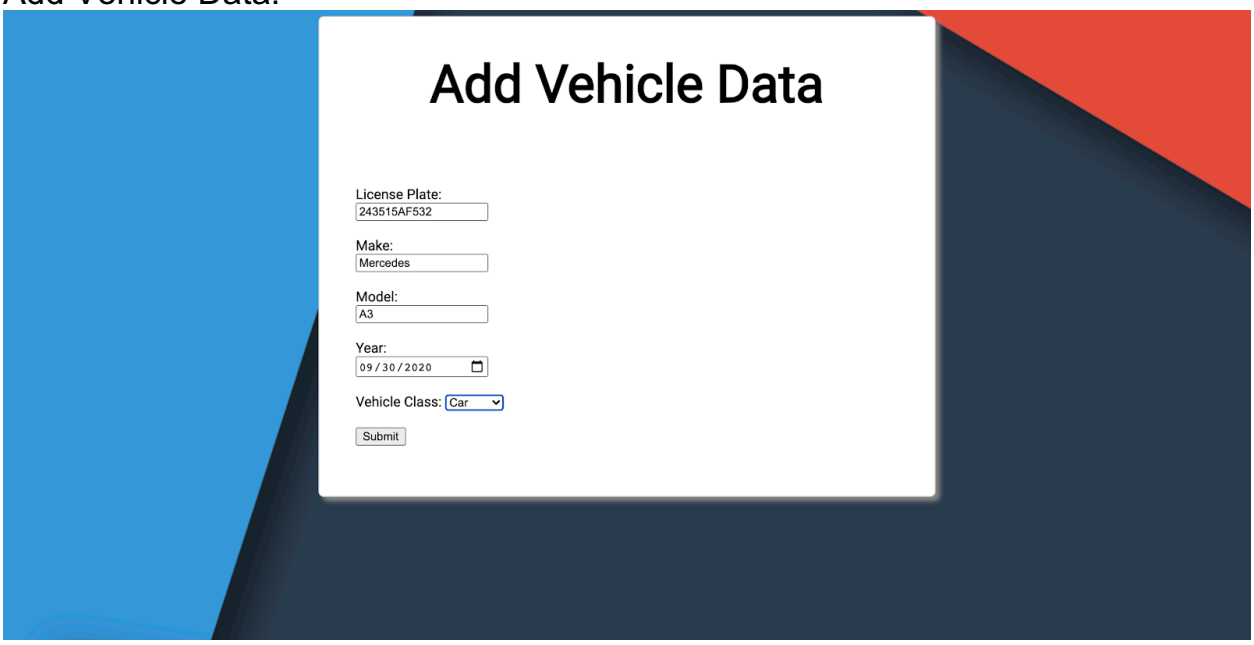
Customer Id: 6

Discount:
15

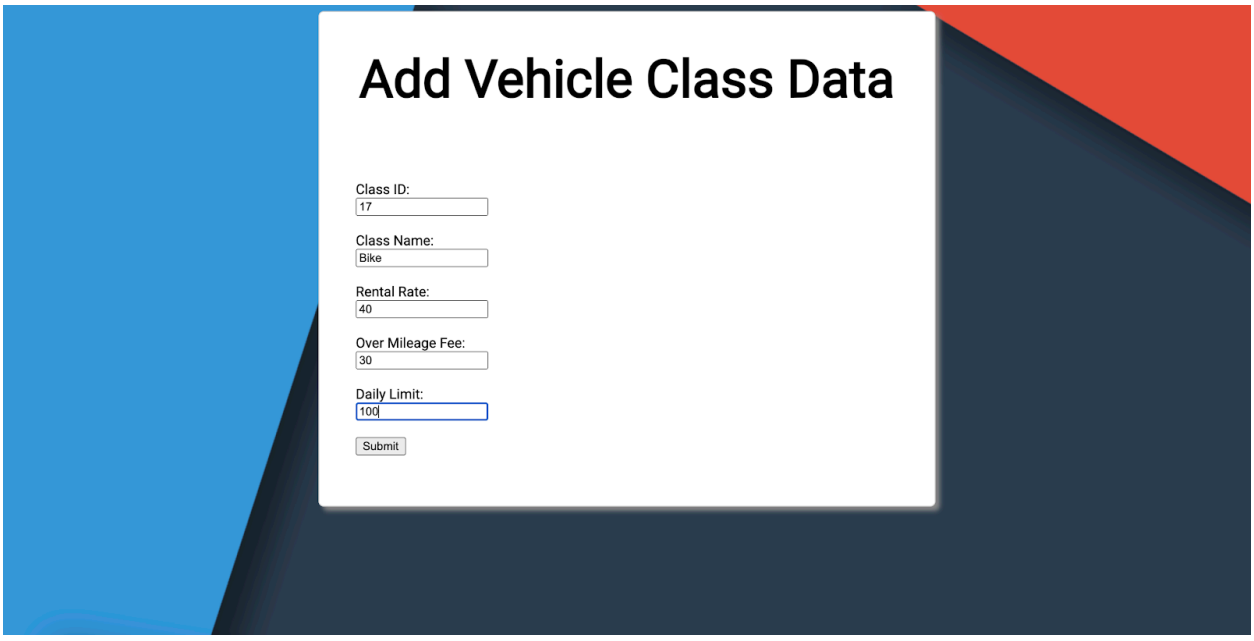
Valid until:
12/31/2020

Submit

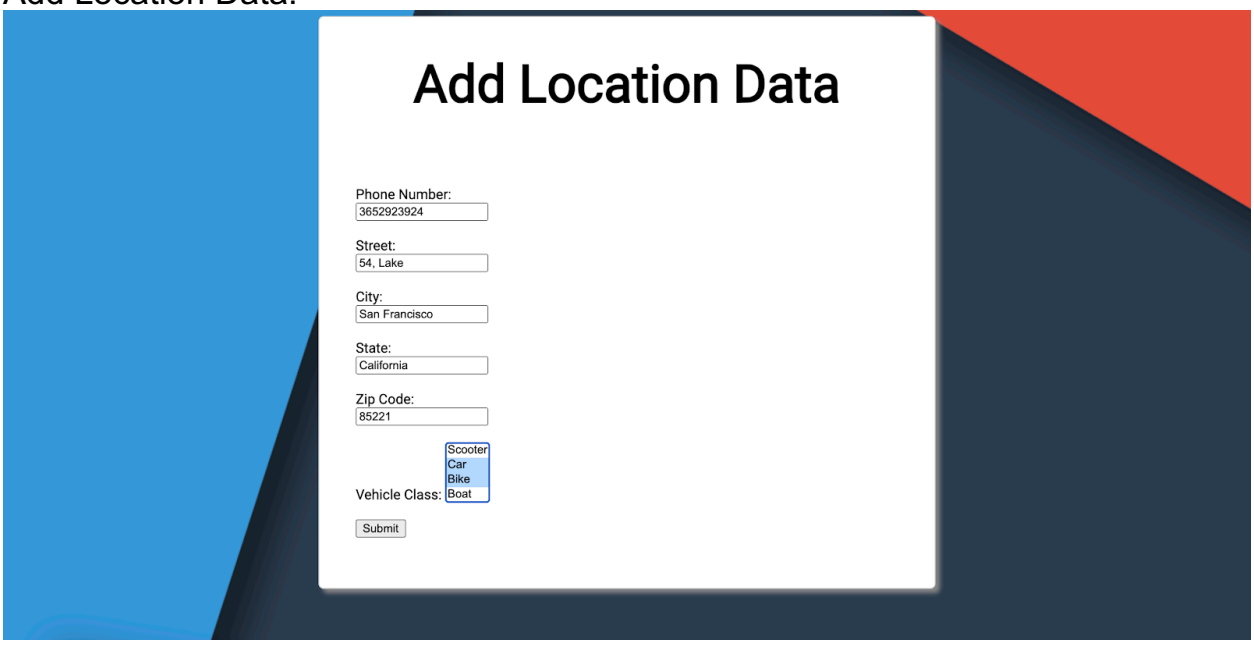
Add Vehicle Data:



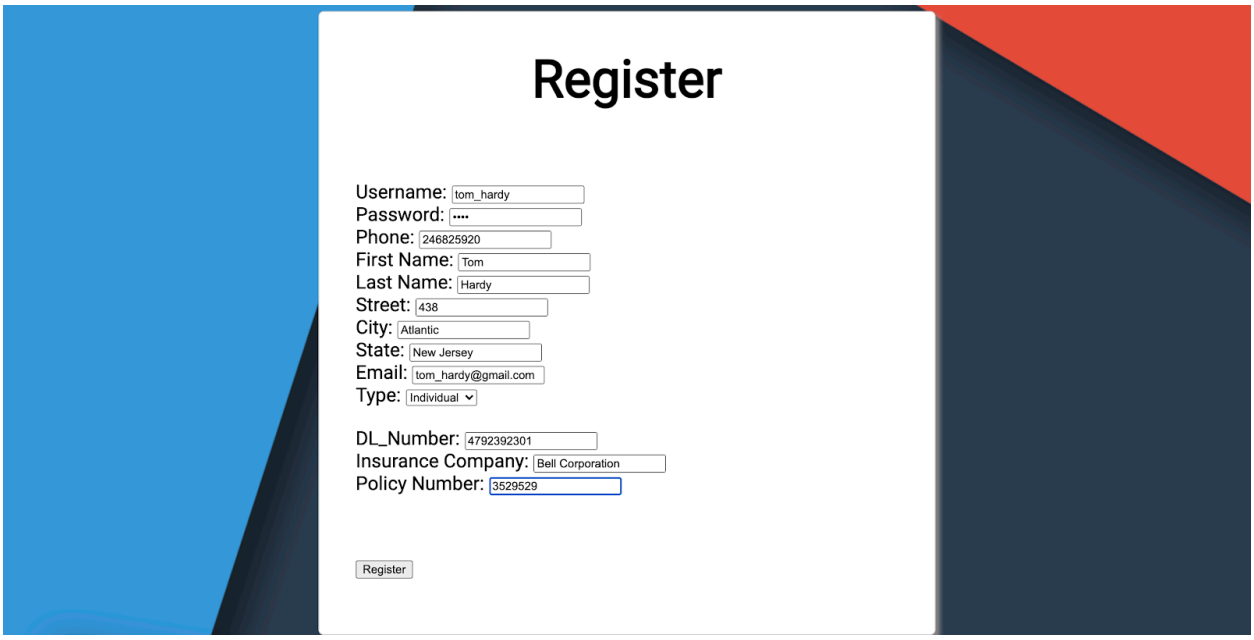
Add Vehicle Class Data:



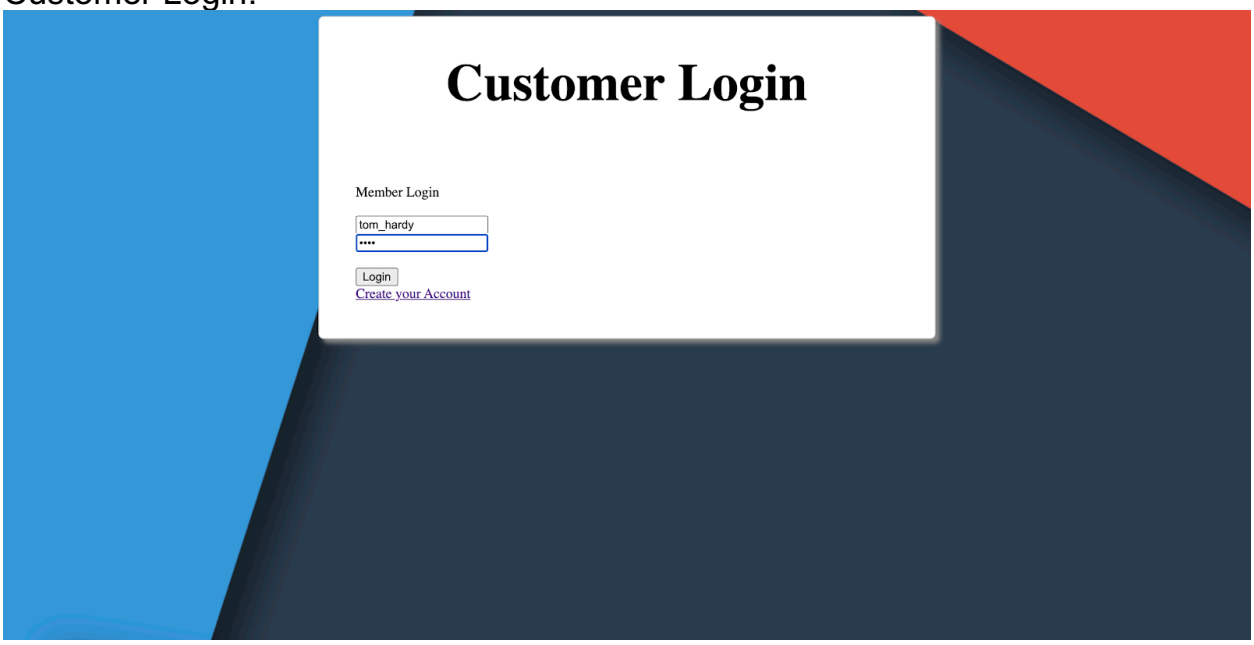
Add Location Data:



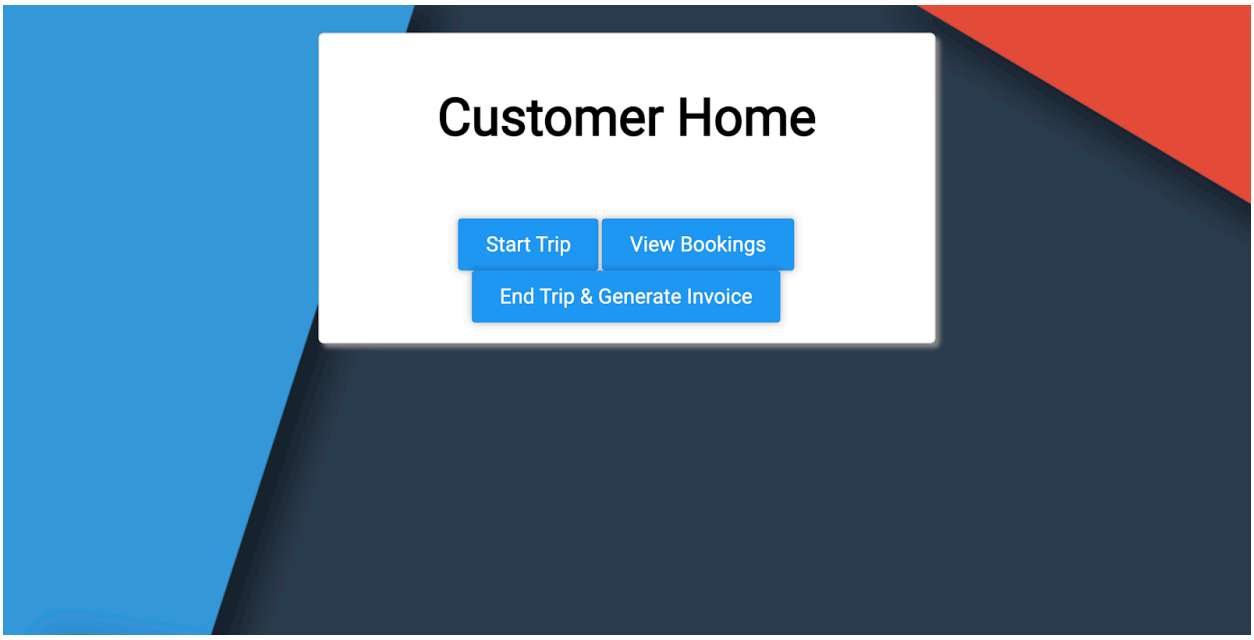
Customer Registration:



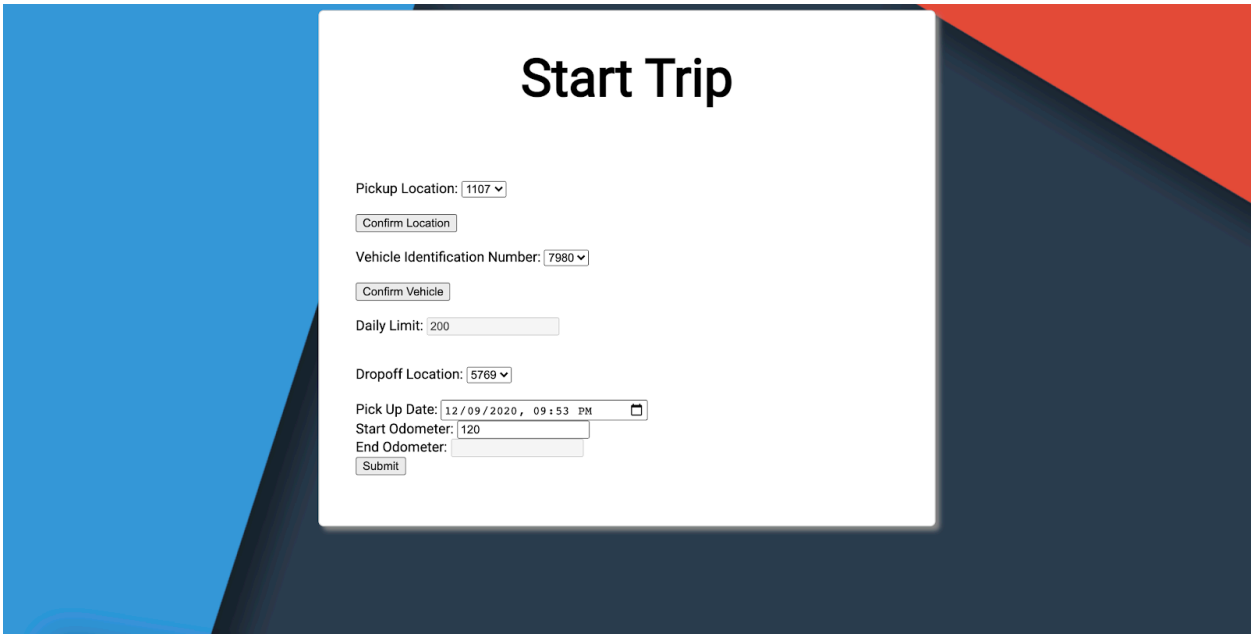
Customer Login:



Customer Portal:



Start Trip:

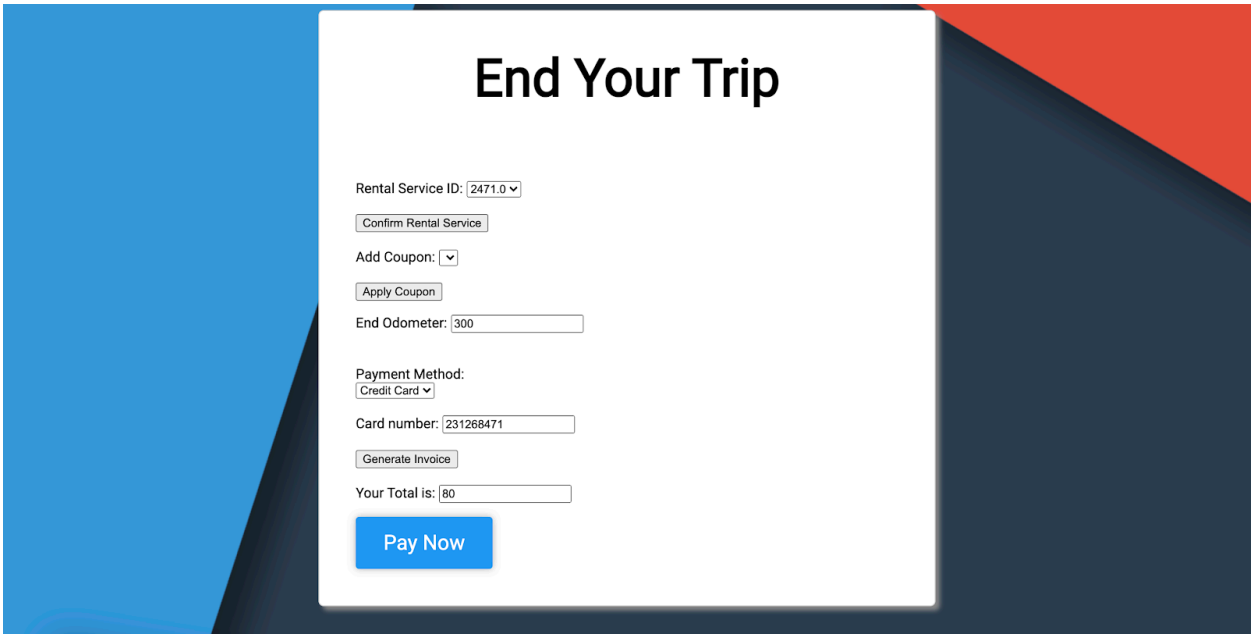


View Bookings:

All Bookings

Rental Service ID	Customer ID	Pickup Location	Dropoff Location	Pickup Date	Dropoff Date	Start Odometer	End Odometer	Vehicle Id
2439.0	1	5769	5769	Dec. 3, 2020, 1 p.m.	Dec. 18, 2020, 4 a.m.	100	200	2501
3540.0	1	5863	5769	Dec. 2, 2020, 11:58 p.m.	Dec. 18, 2020, 3:59 a.m.	100	300	2501
4349.0	1	5863	1107	Dec. 8, 2020, midnight	Dec. 18, 2020, 2:45 a.m.	243	914	2501
4484.0	1	5863	5769	Dec. 1, 2020, 12:48 p.m.	Dec. 18, 2020, 3:50 a.m.	400	700	2501
4568.0	1	5863	8228	Dec. 13, 2020, midnight	Dec. 14, 2020, 5:09 a.m.	50	421	2501
5540.0	1	5863	5863	Dec. 5, 2020, 11:49 p.m.	Dec. 18, 2020, 4:49 a.m.	100	200	2501
6807.0	1	5769	5769	Dec. 9, 2020, 2:54 p.m.	Dec. 18, 2020, 3:55 a.m.	599	899	2501
7021.0	1	1107	1107	Dec. 3, 2020, midnight	Dec. 14, 2020, 4:48 a.m.	31	868	7980
7123.0	1	1107	8102	Dec. 2, 2020, 11:13 p.m.	Dec. 18, 2020, 3:15 a.m.	500	600	5690
7901.0	1	5769	8228	Dec. 3, 2020, 2:53 p.m.	Dec. 18, 2020, 3:53 a.m.	500	700	2501
8615.0	1	5863	5863	Dec. 3, 2020, 3:59 a.m.	Dec. 18, 2020, 2:53 a.m.	321	540	2501
9243.0	1	5863	5863	Dec. 1, 2020, midnight	Dec. 18, 2020, 3:05 a.m.	321	890	2501
9465.0	1	5863	6412	Dec. 11, 2020, 12:56 p.m.	Dec. 18, 2020, 3:56 a.m.	300	500	2501

End Trip:



Security and Additional features

We have implemented the following strategies in order to improve security of our web application:

1. Weaker passwords are not allowed. The user needs to enter a minimum of 8 digits with a mixture of characters, alphabets and numbers.
2. The passwords are hashed and then stored in the database so that a fraudulent activity cannot occur.
3. We ensured that no sensitive information was displayed in any part of the web application.
4. We made our web application robust and secure against SQL attacks.
5. Cross Site Scripting (XSS) or, to give it its proper name, JavaScript injection, can be used for session hijacking, site defacement, network scanning, undermining CSRF defences, site redirection/phishing, load of remotely hosted scripts, data theft and keystroke logging. Attackers are also using XSS more and more frequently. We made sure that no cross-scripting attacks could occur on our application. We used csrf-token in our code for this.

Overall experience

Overall, it was a very fruitful experience. The project helped us understand the database concepts really well. Initially, we got a very good exposure of how the web application should be designed in order to fit the database architecture that we had developed in part 1. The brainstorming sessions to discuss the overall flow of the application in order to be logically sound helped us a lot in understanding and thinking clearly. The practical experience of how an actual web application uses databases to store the data and how it is connected and used was gained in this project. There were a few time constraints due to the COVID situation and it was a bit difficult to manage time and develop the web application in a very short period of time, but we were able to develop and deliver in the assigned time. It helped us learn how to manage time. Finally, we would like to point out that it was a very good learning experience and it was worthwhile to work on this project.

Business Analysis

i) SQL QUERIES

Q1) TABLE JOINS

We intend to find out how much was the amount for the trip for each customer and how many payment methods they used for the billing

```
SELECT
  KPRC_INVOICE.INVOICE_AMOUNT AS AMOUNT,
  KPRC_INVOICE.RENTAL_SERVICE_ID AS SERVICE_ID,
  KPRC_RENTAL_SERVICE.CUST_ID AS CUSTOMER
FROM KPRC_INVOICE
JOIN KPRC_RENTAL_SERVICE
  ON KPRC_INVOICE.RENTAL_SERVICE_ID =
KPRC_RENTAL_SERVICE.RENTAL_SERVICE_ID
JOIN KPRC_PAYMENT
  ON KPRC_PAYMENT.INVOICE_ID = KPRC_INVOICE.INVOICE_ID
ORDER BY KPRC_RENTAL_SERVICE.CUST_ID;
```

SQL Worksheet

Clear Find Actions Save Run

```sql
SELECT
  KPRC_INVOICE.INVOICE_AMOUNT AS AMOUNT,
  KPRC_INVOICE.RENTAL_SERVICE_ID AS SERVICE_ID,
  KPRC_RENTAL_SERVICE.CUST_ID AS CUSTOMER
FROM KPRC_INVOICE
JOIN KPRC_RENTAL_SERVICE
  ON KPRC_INVOICE.RENTAL_SERVICE_ID = KPRC_RENTAL_SERVICE.RENTAL_SERVICE_ID
JOIN KPRC_PAYMENT
  ON KPRC_PAYMENT.INVOICE_ID = KPRC_INVOICE.INVOICE_ID
ORDER BY KPRC_RENTAL_SERVICE.CUST_ID;
```

60	313	3
60	303	3
25	314	4
75	304	4
210	305	5
210	315	5
210	315	5
210	315	5
25	306	6
240	307	7
60	308	8
120	309	9
35	310	10

Download CSV

18 rows selected.

SQL Worksheet

Clear Find Actions Save Run

```sql
SELECT
  KPRC_INVOICE.INVOICE_AMOUNT AS AMOUNT,
  KPRC_INVOICE.RENTAL_SERVICE_ID AS SERVICE_ID,
  KPRC_RENTAL_SERVICE.CUST_ID AS CUSTOMER
FROM KPRC_INVOICE
JOIN KPRC_RENTAL_SERVICE
  ON KPRC_INVOICE.RENTAL_SERVICE_ID = KPRC_RENTAL_SERVICE.RENTAL_SERVICE_ID
JOIN KPRC_PAYMENT
  ON KPRC_PAYMENT.INVOICE_ID = KPRC_INVOICE.INVOICE_ID
ORDER BY KPRC_RENTAL_SERVICE.CUST_ID;
```

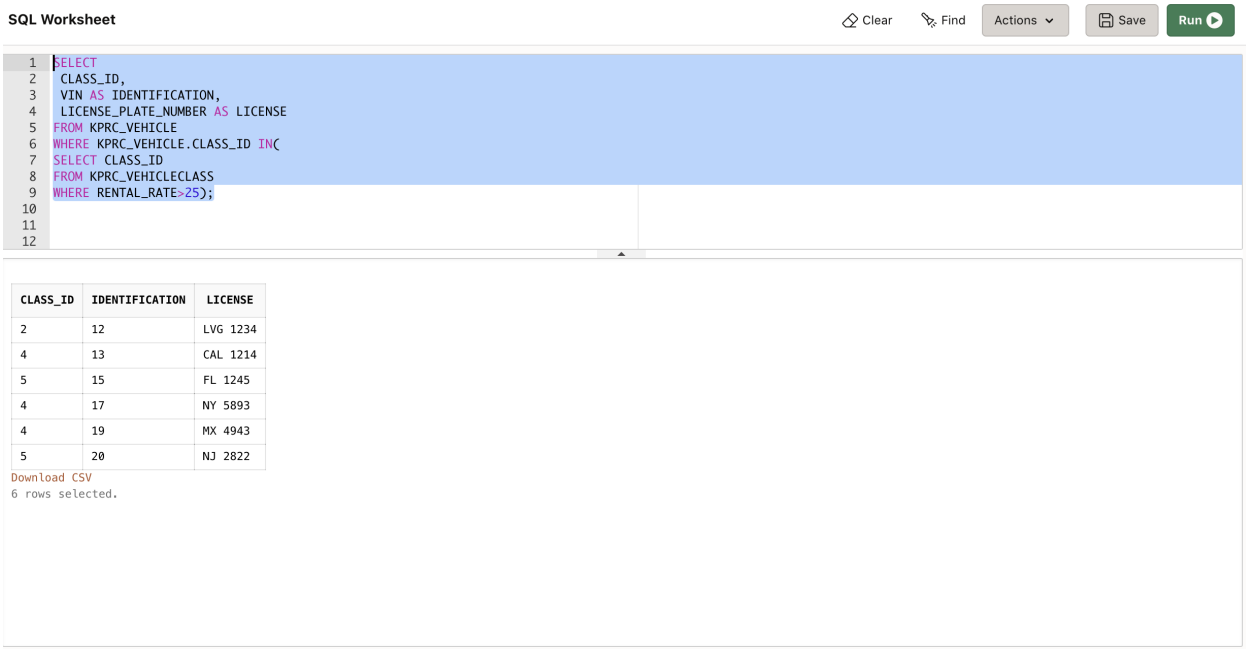
AMOUNT	SERVICE_ID	CUSTOMER
20	301	1
20	301	1
60	311	1
80	302	2
120	312	2
60	313	3
60	303	3
25	314	4
75	304	4
210	305	5
210	315	5
210	315	5
210	315	5
25	306	6

Q2) MULTI-ROW SUBQUERY

We intend to find out the Vehicles whose rental rate is greater than 25

```
SELECT
 CLASS_ID,
 VIN AS IDENTIFICATION,
 LICENSE_PLATE_NUMBER AS LICENSE
```

```
FROM KPRC_VEHICLE
WHERE KPRC_VEHICLE.CLASS_ID IN(
SELECT CLASS_ID
FROM KPRC_VEHICLECLASS
WHERE RENTAL_RATE>25);
```



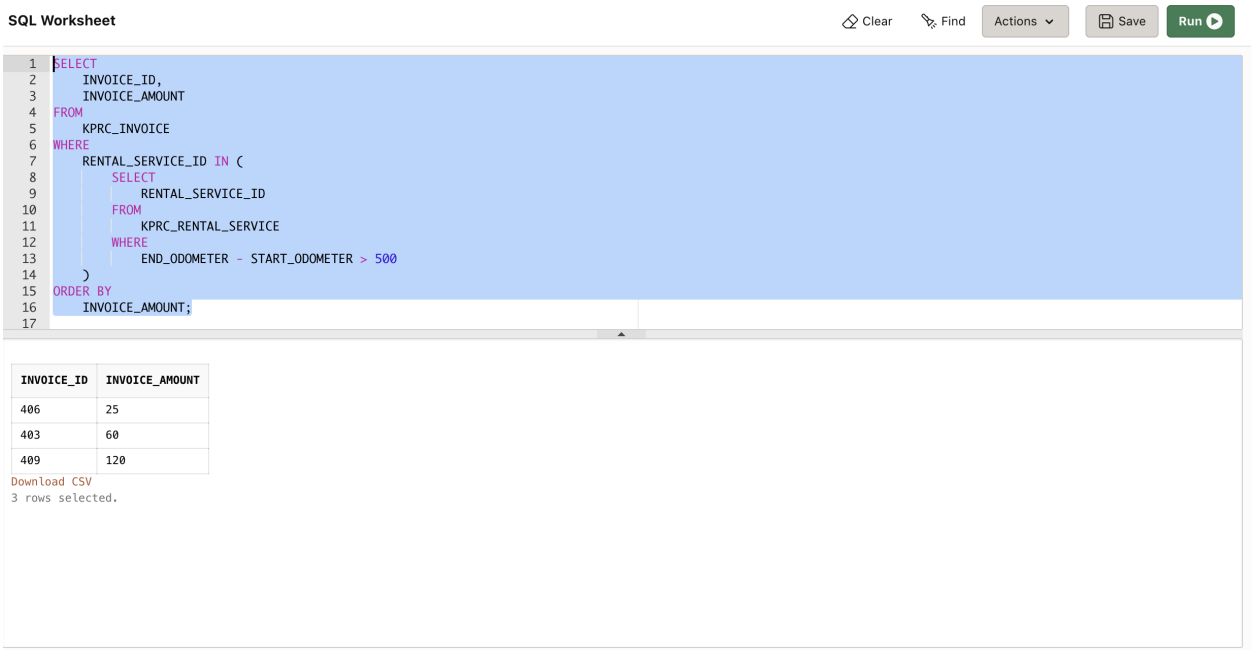
CLASS_ID	IDENTIFICATION	LICENSE
2	12	LVG 1234
4	13	CAL 1214
5	15	FL 1245
4	17	NY 5893
4	19	MX 4943
5	20	NJ 2822

Q3) CORRELATED SUBQUERY

We intend to find out the INVOICE_ID and the AMOUNT for the trips where the odometer reading difference was greater than 500 i.e the trips where the vehicles went a distance of more than 500 miles

```
SELECT
   INVOICE_ID,
   INVOICE_AMOUNT,
FROM
   KPRC_INVOICE
WHERE
   RENTAL_SERVICE_ID IN (
      SELECT
         RENTAL_SERVICE_ID
      FROM
         KPRC_RENTAL_SERVICE
      WHERE
         END_ODOMETER - START_ODOMETER > 500
   )
```

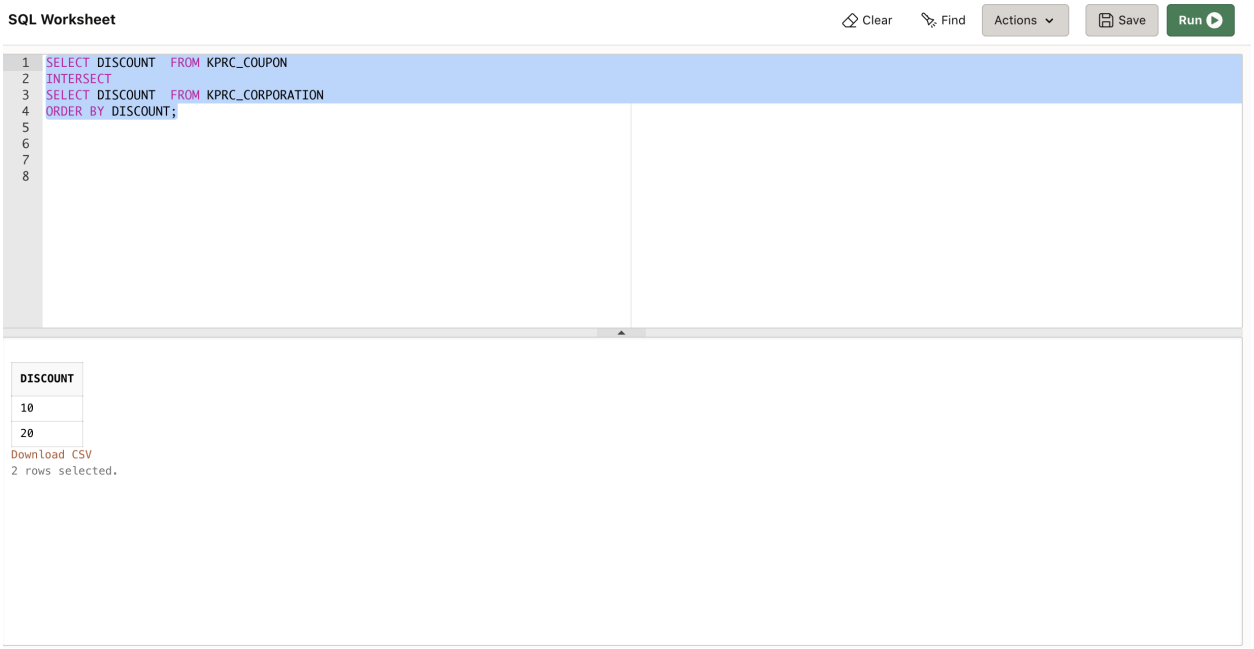
ORDER BY
INVOICE_AMOUNT



Q4) SET OPERATOR QUERY

We intend to find out what are the common discount percentages between corporate and individual customers

SELECT DISCOUNT FROM KPRC_COUPON
INTERSECT
SELECT DISCOUNT FROM KPRC_CORPORATION
ORDER BY DISCOUNT;



Q5) WITH CLAUSE QUERY

We intend to find out the invoice details which have the amount greater than the average invoice amounts in general

```
WITH AVERAGE_TABLE(average_amount) as
   (SELECT avg(INVOICE_AMOUNT)
   from KPRC_INVOICE)
      SELECT *
      FROM KPRC_INVOICE, AVERAGE_TABLE
      WHERE KPRC_INVOICE.INVOICE_AMOUNT > AVERAGE_TABLE.average_amount;
```

SQL Worksheet

Clear Find Actions Save Run

```sql
WITH AVERAGE_TABLE(average_amount) as
    (SELECT avg(INVOICE_AMOUNT)
    from KPRC_INVOICE)
        SELECT *
        FROM KPRC_INVOICE, AVERAGE_TABLE
        WHERE KPRC_INVOICE.INVOICE_AMOUNT > AVERAGE_TABLE.average_amount;
```

INVOICE_ID	INVOICE_DATE	INVOICE_AMOUNT	RENTAL_SERVICE_ID	AVERAGE_AMOUNT
405	15-AUG-20	210	305	93.333333333333333333333333333333333333
407	19-OCT-20	240	307	93.333333333333333333333333333333333333
409	17-DEC-20	120	309	93.333333333333333333333333333333333333
412	19-MAR-20	120	312	93.333333333333333333333333333333333333
415	25-JUN-20	210	315	93.333333333333333333333333333333333333

Download CSV

5 rows selected.

Conclusion

During the course of this project, we learnt a lot of the work and best practices that go into creating a database, the rules to construct a good ER diagram, how to come up with relational schema mapping from the ER diagram, deriving the functional dependencies and how to normalize the relational schema. We learnt on how to design a system from Database perspective and how to efficiently store and manipulate data. In the second part of the project, we learnt how to implement an actual web application that interacts with the database and fetches and stores the results.