AMRITA VISHWA VIDYAPEETHAM

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

19CSE204 – OBJECT ORIENTED PARADIGM

GROUP PROJECT

CAR RENTAL SYSTEM

Overview:

Car rental system is an online system where customers can rent different kinds of cars of their interest. The main aim of this project is to design and develop a database for the car rental company to maintain the records of different types of information such as customer details, different cars details and it would be easy to keep track of different cars. This system simplifies the assignment of different cars and also increases the efficiency as it attracts many new customers.

This system makes the customers work to search for cars easy as all the details about the availability of different cars are available. This system also helps the car rental company to find the customers easily.

GROUP MEMBERS:

ROLL NO.	NAME	EMAIL-ID
CB.EN.U4CSE19453	**R.ABHINAV**	*cb.en.u4cse19453@cb.students.amrita.edu*
CB.EN.U4CSE19459	**S.SHANTHAN**	*cb.en.u4cse19459@cb.students.amrita.edu*
CB.EN.U4CSE19449	**P.KOUSHIK**	*cb.en.u4cse19449@cb.students.amrita.edu*
CB.EN.U4CSE19405	**A.HEMANTH**	*cb.en.u4cse19405@cb.students.amrita.edu*

TEAM CONTRIBUTION:

RollNo/Name	Concept	Contribution	Justify your contribution	Program Name
CB.EN.U4CSE19459 **S.SHANTHAN**	Class	Main	Main class calls the login method from login class and display the user menu if the login class validates the login or display the error message when failed.	Main.java
CB.EN.U4CSE19459 **S.SHANTHAN**	Methods	main	Main method this method connects to the database does implement all of the database operations.	
CB.EN.U4CSE19459 **S.SHANTHAN**	Constructors	main	Initialises the all the variables in this class to null if the datatype is string and zero(0) if the datatype is int or double	
CB.EN.U4CSE19459 **S.SHANTHAN**	Static Concept	Loginbool, main	Main method itself is static and the login_bool is also a static variable which is present in login class and it is used to validate the login here in main method	
CB.EN.U4CSE19459 **S.SHANTHAN**	Inheritance —Type	Inherited from login	Class main is inherited from login class since it uses the method login which should restrict access to users	

CB.EN.U4CSE19459 **S.SHANTHAN**	Package	Com.amrita	All the classes present in the project are the part of the package com.amrita	
CB.EN.U4CSE19459 **S.SHANTHAN**	Exception handling	sqlexceptions	Expected SQLExceptions are mentioned in throws and try block captures the errors that might occur while the SQL statements getting executed.	

PROGRAM :

```
package com.amrita;

import com.sun.scenario.effect.impl.sw.sse.SSEBlend_SRC_OUTPeer;
import java.sql.SQLException;
import java.util.Scanner;
import java.lang.String;


public class Main extends login {
    public static void main(String[] args) throws SQLException {
        login x = new login();
        x.login();
        if (login_bool) {
            Scanner user_input = new Scanner(System.in);
            DB_OPERATIONS d = new DB_OPERATIONS();
            String query;
            System.out.println("==============car rental system
console==========================");
            System.out.println("1. Insert");
            System.out.println("2. Update");
            System.out.println("3. Select");
            System.out.println("4. Delete");
            System.out.println("====================Please Select an option
======================");
            int option = user_input.nextInt();
            System.out.println("(CUSTOMER || CAR || BOOKING || BILLING ||
PAYMENT)");
            if (option == 1) {
                String table_name;
                System.out.println("Please enter the table name : ");
                user_input.nextLine();
                table_name = user_input.nextLine();
                if (table_name.equals("CUSTOMER")) {
```

```
                    System.out.println("Enter ID : ");
                    String CUSTOMER_ID = user_input.nextLine();
                    user_input.nextLine();
                    System.out.println("Enter Phone number : ");
                    String PHONE_NUMBER = user_input.nextLine();
                    user_input.nextLine();

                    System.out.println("Enter Email");
                    String EMAIL = user_input.nextLine();
                    user_input.nextLine();

                    System.out.println("Enter Street");
                    String STREET = user_input.nextLine();

                    System.out.println("Enter City");
                    String CITY = user_input.nextLine();

                    System.out.println("Enter Zipcode");
                    String ZIPCODE = user_input.nextLine();

                    System.out.println("Enter Aadhaar Id");
                    String AADHAAR = user_input.nextLine();

                    query = "INSERT INTO " + table_name + " VALUES('" +
CUSTOMER_ID + "','" + PHONE_NUMBER + "','" + EMAIL + "','" + STREET + "','"
+ CITY + "','" + ZIPCODE + "','" + AADHAAR + "')";
                    d.Insert(query);
                } else if (table_name.equals("CAR")) {

                    System.out.println("Enter CAR_ID : ");
                    String CAR_ID = user_input.nextLine();
                    System.out.println("Enter car model : ");
                    String CAR_MODEL = user_input.nextLine();


                    System.out.println("Enter customer id");
                    String CUSTOMER_ID = user_input.nextLine();


                    System.out.println("Enter location");
                    String LOCATION = user_input.nextLine();

                    System.out.println("Enter registration number   ");
                    String REG_NO = user_input.nextLine();

                    query = "insert into " + table_name + " values('" +
CAR_ID + "','" + CAR_MODEL + "','" + CUSTOMER_ID + "','" + LOCATION + "','"
+ REG_NO + "')";
                    d.Insert(query);
                } else if (table_name.equals("BOOKING")) {

                    System.out.println("Enter BOOKING ID : ");
                    String BOOKING_ID = user_input.nextLine();

                    System.out.println("Enter car id");
                    String CAR_ID = user_input.nextLine();

                    System.out.println("Enter customer id");
                    String CUSTOMER_ID = user_input.nextLine();

                    System.out.println("Enter FROM DATE   ");
```

```
                    String FROM_DATE = user_input.nextLine();

                    System.out.println("Enter TO DATE   ");
                    String TO_DATE = user_input.nextLine();

                    query = "insert into " + table_name + " values('" +
BOOKING_ID + "','" + CAR_ID + "','" + CUSTOMER_ID + "','" + FROM_DATE +
"','" + TO_DATE + "')";
                    d.Insert(query);
                } else if (table_name.equals("BILLING")) {
                    System.out.println("Enter BILLING ID : ");
                    String BILLING_ID = user_input.nextLine();
                    System.out.println("Enter BOOKING ID");
                    String BOOKING_ID = user_input.nextLine();
                    System.out.println("Enter customer id");
                    String CUSTOMER_ID = user_input.nextLine();
                    System.out.println("Enter CAR ID   ");
                    String CAR_ID = user_input.nextLine();

                    System.out.println("Enter BILLING DATE ");
                    String BILL_DATE = user_input.nextLine();

                    query = "insert into " + table_name + " values('" +
BILLING_ID + "','" + BOOKING_ID + "','" + CUSTOMER_ID + "','" + CAR_ID +
"','" + BILL_DATE + "')";
                    d.Insert(query);
                } else if (table_name.equals("PAYMENT")) {
                    System.out.println("Enter PAYMENT METHOD : ");
                    String PAYMENT_METHOD = user_input.nextLine();
                    System.out.println("Enter PAYMENT STATUS");
                    String PAYMENT_STATUS = user_input.nextLine();
                    System.out.println("Enter BILLING ID");
                    String BILLING_ID = user_input.nextLine();
                    System.out.println("Enter TOTAL AMOUNT   ");
                    String TOTAL_AMOUNT = user_input.nextLine();
                    query = "insert into " + table_name + " values('" +
PAYMENT_METHOD + "','" + PAYMENT_STATUS + "','" + BILLING_ID + "','" +
TOTAL_AMOUNT + "')";
                    d.Insert(query);
                }
            } else if (option == 2) {
                String table_name;
                System.out.println("Please enter the table name : ");
                user_input.nextLine();
                table_name = user_input.nextLine();
                if (table_name.equals("CUSTOMER")) {
                    System.out.println("enter column ");
                    String UpdateCol = user_input.nextLine();
                    System.out.println("enter value: ");
                    String Value = user_input.nextLine();
                    System.out.println("enter  corresponding CUSTOMER_ID:
");
                    int changes = user_input.nextInt();
                    query = "UPDATE " + table_name + " SET " + UpdateCol +
"='" + Value + "' WHERE CUSTOMER_ID = " + changes;
                    d.Update(query);
                } else if (table_name.equals("CAR")) {
                    System.out.println("enter column: ");
                    String UpdateCol = user_input.nextLine();
                    System.out.println("enter value: ");
                    String Value = user_input.nextLine();
```

```
                    System.out.println("enter corresponding CAR_ID : ");
                    int changes = user_input.nextInt();
                    query = "UPDATE " + table_name + " SET " + UpdateCol +
"='" + Value + "' WHERE CAR_ID = " + changes;
                    d.Update(query);
                } else if (table_name.equals("BOOKING")) {
                    System.out.println("enter column: ");
                    String UpdateCol = user_input.nextLine();
                    System.out.println("enter value: ");
                    String Value = user_input.nextLine();
                    System.out.println("enter corresponding BOOKING_ID: ");
                    int changes = user_input.nextInt();
                    query = "UPDATE " + table_name + " SET " + UpdateCol +
"='" + Value + "' WHERE BOOKING_ID = " + changes;
                    d.Update(query);
                } else if (table_name.equals("BILLING")) {
                    System.out.println("enter column: ");
                    String UpdateCol = user_input.nextLine();
                    System.out.println("enter value: ");
                    String Value = user_input.nextLine();
                    System.out.println("enter corresponding BILLING_ID: ");
                    int changes = user_input.nextInt();
                    query = "UPDATE " + table_name + " SET " + UpdateCol +
"='" + Value + "' WHERE BILLING_ID = " + changes;
                    d.Update(query);
                } else if (table_name.equals("PAYMENT")) {
                    System.out.println("enter column ");
                    String UpdateCol = user_input.nextLine();
                    System.out.println("enter value: ");
                    String Value = user_input.nextLine();
                    System.out.println("enter corresponding PAYMENT_METHOD
: ");
                    String changes = user_input.nextLine();
                    query = "UPDATE " + table_name + " SET " + UpdateCol +
"='" + Value + "' WHERE PAYMENT_METHOD = " + changes;
                    d.Update(query);
                }
            } else if (option == 3) {
                String table_name;
                System.out.println("Please enter the table name : ");
                user_input.nextLine();
                table_name = user_input.nextLine();
                query = "SELECT * FROM " + table_name;
                d.Select(query);
            } else if (option == 4) {
                String table_name;
                System.out.println("Please enter the table name : ");
                user_input.nextLine();
                table_name = user_input.nextLine();
                System.out.println("enter id to delete : ");
                int DeleteEntry = user_input.nextInt();
                query = "DELETE FROM " + table_name + " WHERE BILLING_ID ="
+ DeleteEntry;
                d.Delete(query);
            }
        }
    }
}
```

Rollo/Name	Concept	Contribution	Justify your contribution	Program Name
CB.EN.U4CSE19453 R.ABHINAV	Class	signup	Sign up class gets username and password from the users and asks to validate by re-entering the password. If it validates user credentials gets added to the database.	signup.java
CB.EN.U4CSE19453 **R.ABHINAV**	Methods	signup	signup method here all the interactions with the user are done using CLI and validations and data is queried to and froth the Mysql server which is the backend.	signup.java
CB.EN.U4CSE19453 **R.ABHINAV**	Constructor	signup	Default constructor to invoke be invoked on creation of object of signup class.	signup.java
CB.EN.U4CSE19453 **R.ABHINAV**	Inheritance	Inherited from Database Operations	Class signup is inherited from Database Operations class since it uses the method Insert which stores credentials of the user once he signed up	signup.java
CB.EN.U4CSE19453 **R.ABHINAV**	Package	com.amrita	Since to facilitate inheritance all the classes for this project are placed under this package.	signup.java
CB.EN.U4CSE19453 **R.ABHINAV**	Exception handling	try, throws	Expected SQLExceptions are mentioned in throws and try block captures the errors that might occur while the SQL statements getting executed.	signup.java

PROGRAM :

```
package com.amrita;

import java.sql.*;
import java.util.Scanner;

public class Signup extends DB_OPERATIONS {
    //Default constuctor
    public Signup() {}
    //connection variables
    String url = "jdbc:mysql://localhost:3306/CARRENTALSYSTEM";;
    String pass = "root";
    String user = "root";
    public void signup() throws SQLException {
        try (
                //establishing connection with the database
                Connection connection = DriverManager.getConnection(url,
user, pass);

                //Creating a statement
                Statement statement = connection.createStatement();
        ) {
            Scanner input = new Scanner(System.in);
            System.out.println("==========Welcome to the Signup
Page==============");
            System.out.print("Username : ");
            String username = input.next();
            System.out.print("\nPassword : ");
            String password = input.next();
            System.out.print("\nConfirm Password : ");
            String Confirm_Password = input.next();

            //checking if the password and the confirm password is same
else return
            if (password.equals(Confirm_Password)) {
                String query = "select username from credentials";

                //storing the result of the select query in a result set
object
                ResultSet resultset = statement.executeQuery(query);

                //initialzing a count variable to count how many times the
username is in the credentials relation
                int count = 0;

                //here we iterate through the object and check if the user
name is already taken
                while(resultset.next()){
                    String username_db = resultset.getString("username");
                    if(username.equals(username_db)){
                        count = count + 1;
                    }
                }
                if(count == 0){
                    String exeQuery = "INSERT INTO credentials VALUES
('"+username+"','"+password+"')";
```

```
                    Insert(exeQuery);
                    System.out.println("SignUp Sucessfull!! You can now
login");
                }


            } else {
                return;
            }
        }
    }
}
```

RollNo/Name	Concept	Contribution	Justify your contribution	Program Name
CB.EN.U4CSE19449 **P.KOUSHIK**	Class	login	Login class performs all the operations such as showing the menu and then executing the code based on user's choice	login.java
CB.EN.U4CSE19449 **P.KOUSHIK**	Methods	login	Login method here displaying the menu and program in favour of user choice is done. If user chooses login it authenticates user using his username and password. If user doesn't have an account, he can choose signup to get signed up	login.java
CB.EN.U4CSE19449 **P.KOUSHIK**	Construc tor	login	Default constructor to invoke be invoked on creation of object of login class.	login.java
CB.EN.U4CSE19449 **P.KOUSHIK**	Inheritan ce	Inherited from signup	Class login is inherited from signup class since it uses the method signup in case if the user isn't signed up before	login.java

CB.EN.U4CSE19449 **P.KOUSHIK**	Package	com.amrita	Since to facilitate inheritance all the classes for this project are placed under this package.	login.java
CB.EN.U4CSE19449 **P.KOUSHIK**	Exception handling	try, throws	Expected SQLExceptions are mentioned in throws and try block captures the errors that might occur while the SQL statements getting executed.	login.java

PROGRAM :

```
package com.amrita;

import java.sql.*;
import java.util.Scanner;

public class login extends Signup{
    public static boolean login_bool;

    //Default Constructors
    public login(){ }

    public boolean login() throws SQLException{

        //step-1 : creating a database connection - in the project
        try (
                Connection connection = DriverManager.getConnection(url,
user, pass);

                //creating a statement and executing a query
                Statement statement = connection.createStatement();

        ) {
            //user selects an option here to sign  up or login


            System.out.println("===========Menu==================");

            System.out.println("1. Login");
            System.out.println("2. Sign Up");
            System.out.println("============END==================");
            //creating a scanner object to take the inputs
            Scanner input = new Scanner(System.in);
            int option = input.nextInt();
            if(option == 1){
```

```
                // generating the select query
                String query = "select * from credentials";

                //executing the query and storing the result set in an
object
                ResultSet resultset = statement.executeQuery(query);

                //using the information that is stored in result set to
validate the login
                String username_input;
                String password_input;
                //input username
                System.out.print("Username : ");
                username_input = input.next();
                System.out.print("\n");
                //input password
                System.out.print("Password : ");

                       password_input = input.next();
                          //checking for the user int the database;
                                  while(resultset.next()) {
                          String Username = resultset.getString("username");

                      String Password = resultset.getString("password");

                      if(Username.equals(username_input)){
                          if(Password.equals(password_input)){
                              //if username and password both are found in
the database then login_bool i set to true
                              login_bool = true;
                              return login_bool;
                          }
                          else{
                              //else set to false
                              login_bool = false;
                          }
                      }
                  }
              }
              else if(option == 2){
                  signup();
                  login();
              }
          }
          return login_bool;
      }
}
```

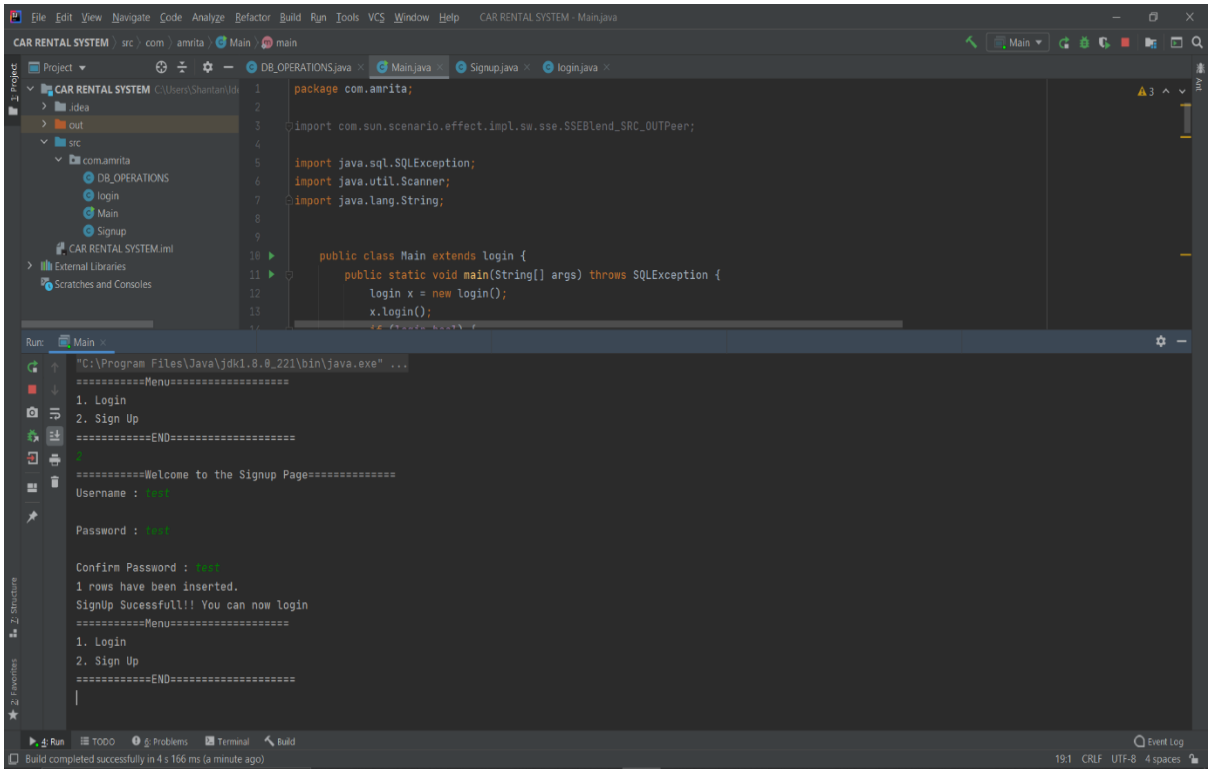
Rollo/Name	Concept	Contribution	Justify your contribution
CB.EN.U4CSE19405 **A.HEMANTH**	Class	DB Operations	DB operations helps in connecting frontend with the database
CB.EN.U4CSE19405 **A.HEMANTH**	Methods	Update, select insert, delete	1.Update:update the database through the query. 2.select:displays the data from database 3.insert:adds data into the database 4.Delete:deletes the data from database
CB.EN.U4CSE19405 **A.HEMANTH**	Constructor	DB operations	Default constructor to invoke be invoked on creation of object of DB operations class.
CB.EN.U4CSE19405 **A.HEMANTH**	Inheritance	Base class for inheritance	It is the base class for inheritance of main class,signup class.
CB.EN.U4CSE19405 **A.HEMANTH**	Package	com.amrita	Since to facilitate inheritance all the classes for this project are placed under this package.
CB.EN.U4CSE19405 **A.HEMANTH**	Exception handling	try, throws	Expected SQLExceptions are mentioned in throws and try block captures the errors that might occur while the SQL statements getting executed.

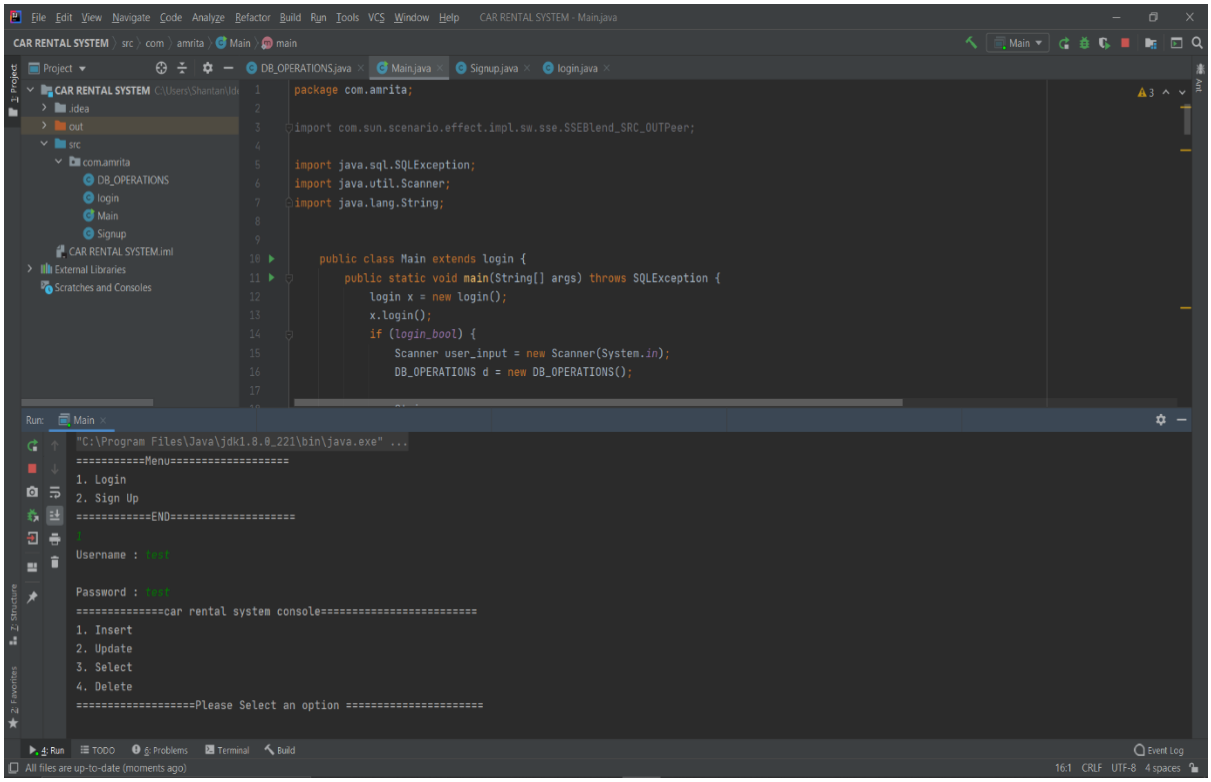
PROGRAM :

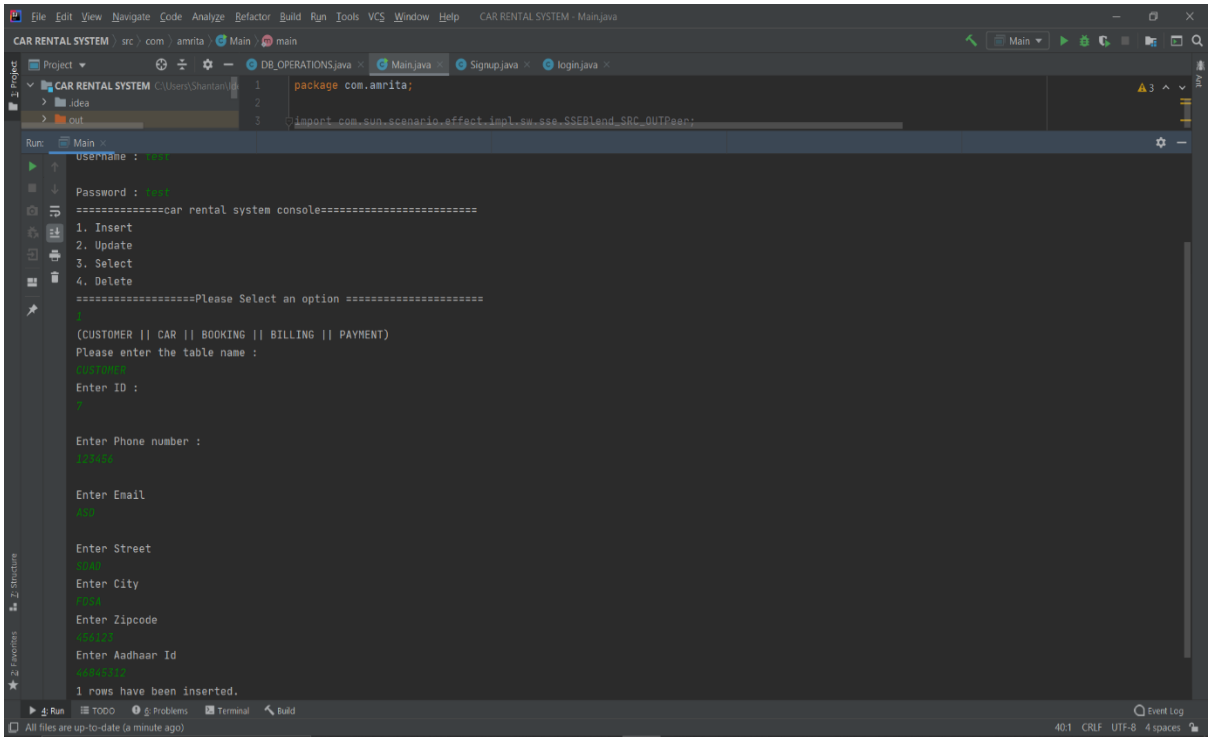
```
package com.amrita;

import java.sql.*;

public class DB_OPERATIONS {
    public String username = "root";
    public String password = "root";
    public String Dburl = "jdbc:mysql://localhost:3306/CARRENTALSYSTEM";

    public void Insert(String query) throws SQLException {
        try{
            int rows_affected = 0;
            //creating the connection to the database
            Connection connect =
DriverManager.getConnection(Dburl,username,password);

            //creating a statement && executing a query
            Statement statement = connect.createStatement();

            // excecuting a query
            rows_affected = statement.executeUpdate(query);
            //prinring the result
            System.out.println(rows_affected + " rows have been
inserted.");
            //closing the connection
            connect.close();
        } catch (SQLException throwables) {
            throwables.printStackTrace();
        }
    }
    public void Update(String query) throws SQLException{
        try{
            int rows = 0;

            //creating the connection to the database
            Connection connection =
DriverManager.getConnection(Dburl,username,password);

            // creating a statement && executing the query here
            Statement statement = connection.createStatement();
            //executing a query
            rows = statement.executeUpdate(query);
            //printing the rows affected
            System.out.println(rows + " rows have been updated.");
            //closing the connection
            connection.close();
        } catch (SQLException throwables) {
            throwables.printStackTrace();
        }
    }
    public void Delete(String query) throws SQLException{
        try{
            int rows_effected = 0;
            //creating a connection to the database
            Connection connection =
```

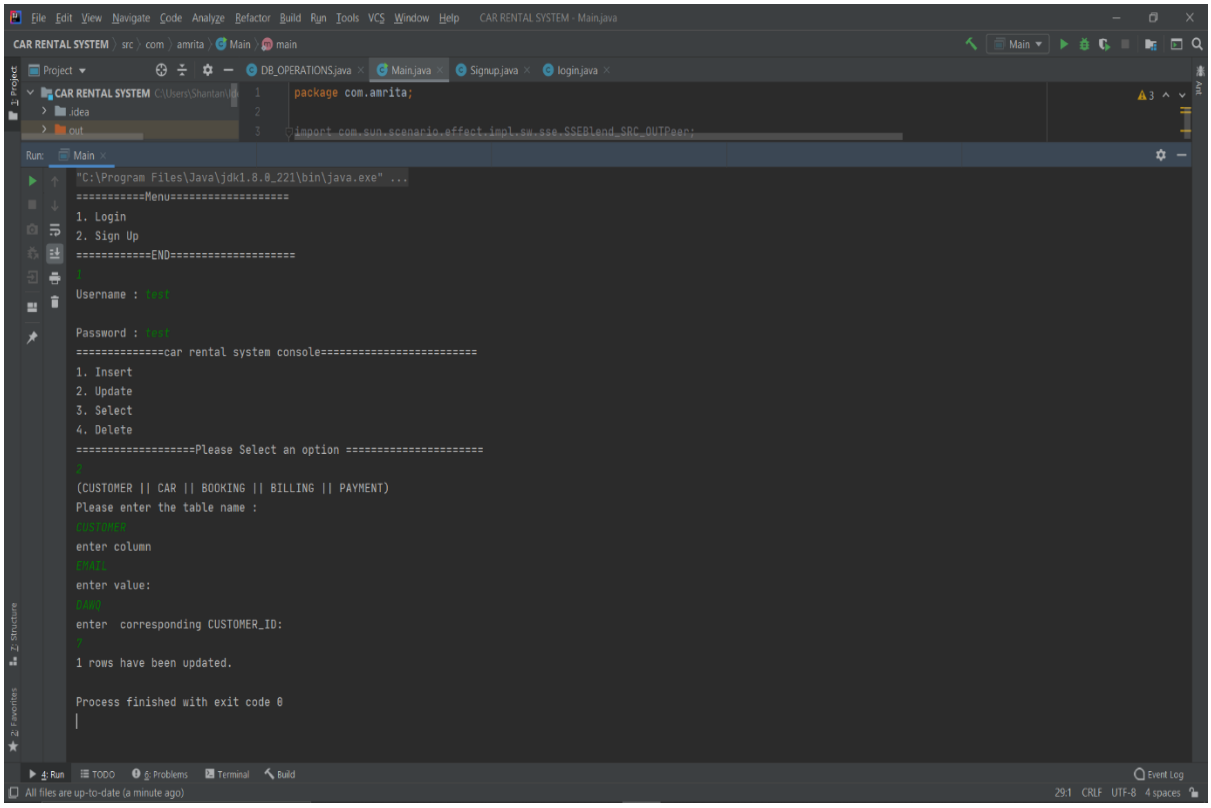
```
DriverManager.getConnection(Dburl,username,password);
            //creating a statement
            Statement statement = connection.createStatement();
            //executing a query
            rows_effected = statement.executeUpdate(query);
            //printing the rows effected statement;
            System.out.println(rows_effected + " have been updated");
            connection.close();
        } catch (SQLException throwables){
            throwables.printStackTrace();
        }
    }

    public void Select(String query) throws SQLException{
        try{
            int count = 0;
            //creating a connection to the database
            Connection connection =
DriverManager.getConnection(Dburl,username,password);
            //creating a statement
            Statement statement = connection.createStatement();
            //storing the result of executed query in a result set object
            ResultSet set = statement.executeQuery(query);
            while(set.next()){
                String id = set.getString("CUSTOMER_ID");
                String name = ((ResultSet) set).getString("PHONE_NUMBER");
                String phno = set.getString("EMAIL");
                String street = ((ResultSet) set).getString("STREET");
                String city = set.getString("CITY");
                String zipcode = ((ResultSet) set).getString("ZIPCODE");
                String aadhaar = set.getString("AADHAAR");


                System.out.println(id + " "+ name + "  "+ phno + "  "+
street + "  "+ city + " "+zipcode+ " "+ aadhaar );
                count++;
            }
            System.out.println(count + " rows are Selected");
        } catch (SQLException throwables) {
            throwables.printStackTrace();
        }
    }
}
```

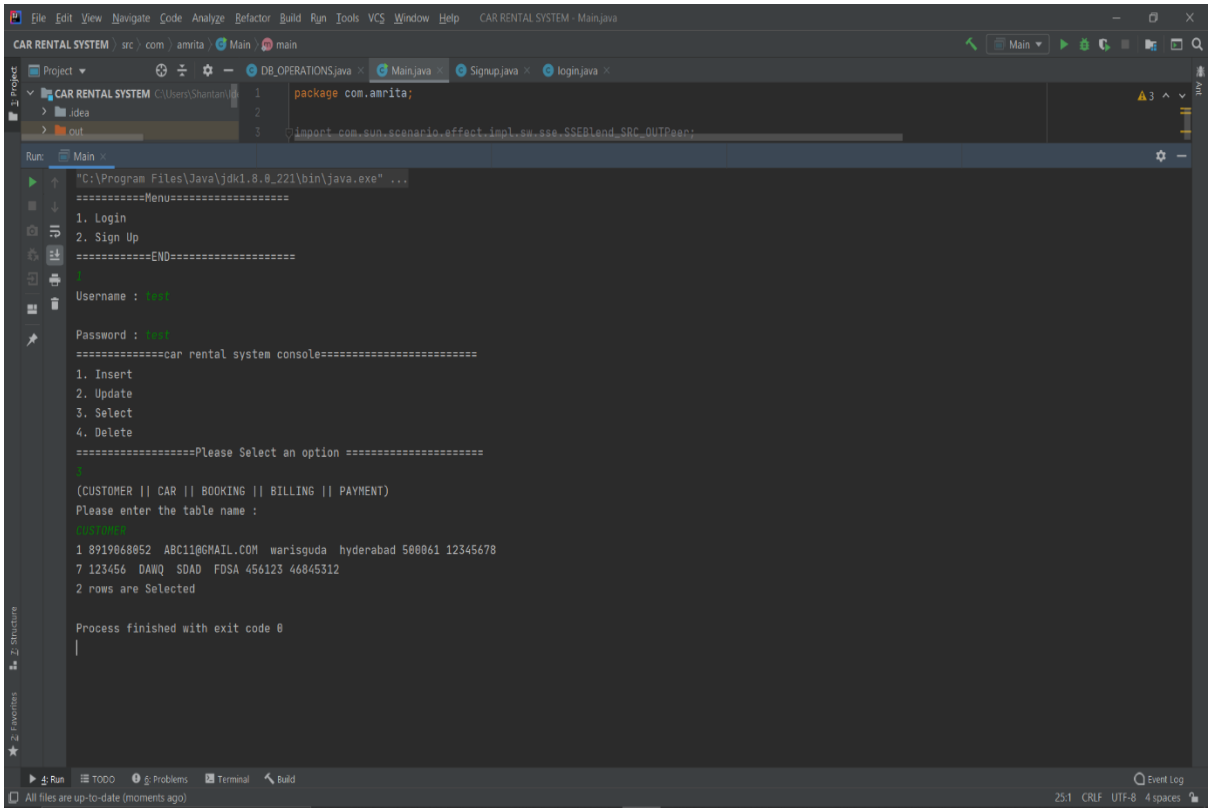
OUTPUT


CAR RENTAL SYSTEM - Main.java
package com.amrita;
import com.sun.scenario.effect.impl.sw.sse.SSEBlend_SRC_OUTPeer;
import java.sql.SQLException;
import java.util.Scanner;
import java.lang.String;
public class Main extends login {
public static void main(String[] args) throws SQLException {
login x = new login();
x.login();
"C:\Program Files\Java\jdk1.8.0_221\bin\java.exe" ...
==========Menu==================
1. Login
2. Sign Up
============END==================
2
==========Welcome to the Signup Page==========
Username : test
Password : test
Confirm Password : test
1 rows have been inserted.
SignUp Sucessfull!! You can now login
==========Menu==================
1. Login
2. Sign Up
============END==================
Build completed successfully in 4 s 166 ms (a minute ago)


CAR RENTAL SYSTEM - Main.java
package com.amrita;
import com.sun.scenario.effect.impl.sw.sse.SSEBlend_SRC_OUTPeer;
import java.sql.SQLException;
import java.util.Scanner;
import java.lang.String;
public class Main extends login {
public static void main(String[] args) throws SQLException {
login x = new login();
x.login();
if (login_bool) {
Scanner user_input = new Scanner(System.in);
DB_OPERATIONS d = new DB_OPERATIONS();
"C:\Program Files\Java\jdk1.8.0_221\bin\java.exe" ...
==========Menu==================
1. Login
2. Sign Up
============END==================
1
Username : test
Password : test
==========car rental system console==========
1. Insert
2. Update
3. Select
4. Delete
==========Please Select an option ==========
All files are up-to-date (moments ago)


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
CAR RENTAL SYSTEM - Main.java
CAR RENTAL SYSTEM
DB_OPERATIONS.java
Main.java
Signup.java
login.java
package com.amrita;
Run: Main
Username : test
Password : test
==========car rental system console==========
1. Insert
2. Update
3. Select
4. Delete
==========Please Select an option ==========
1
(CUSTOMER || CAR || BOOKING || BILLING || PAYMENT)
Please enter the table name :
CUSTOMER
Enter ID :
7
Enter Phone number :
123456
Enter Email
ASD
Enter Street
ROAD
Enter City
FOSA
Enter Zipcode
456123
Enter Aadhaar Id
46845312
1 rows have been inserted.
4: Run
TODO
Problems
Terminal
Build
Event Log
All files are up-to-date (a minute ago)
40:1
CRLF
UTF-8
4 spaces


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
CAR RENTAL SYSTEM - Main.java
CAR RENTAL SYSTEM
DB_OPERATIONS.java
Main.java
Signup.java
login.java
package com.amrita;
Run: Main
"C:\Program Files\Java\jdk1.8.0_221\bin\java.exe" ...
==========Menu==========
1. Login
2. Sign Up
==========END==========
1
Username : test
Password : test
==========car rental system console==========
1. Insert
2. Update
3. Select
4. Delete
==========Please Select an option ==========
2
(CUSTOMER || CAR || BOOKING || BILLING || PAYMENT)
Please enter the table name :
CUSTOMER
enter column
EMAIL
enter value:
DAMO
enter corresponding CUSTOMER_ID:
7
1 rows have been updated.
Process finished with exit code 0
4: Run
TODO
Problems
Terminal
Build
Event Log
All files are up-to-date (a minute ago)
29:1
CRLF
UTF-8
4 spaces



PROJECT SOURCE CODE:

Drive Link :

https://drive.google.com/drive/folders/14PuLks3LWJMAUDDfUaDR9V6E4JhfxVCZ?usp=sharing

----------THANK YOU----------