

KAR KIRAYA DOCUMENTATION CAR BUY AND RENTAL SYSTEM



TEAM MEMBERS:

**Narayana Mrunal
Devansh Srivastava
Hrithik Anilkumar Sharma
Chethan S
Varsha H P**

INDEX

- 1. INTRODUCTION**
- 2. GOALS OF THE PROJECT**
- 3. PROBLEM IN EXISTING SYSTEMS**
- 4. PROPOSED SYSTEM**
- 5. REQUIREMENTS**
 - 5.1 Functional Requirements**
 - 5.2 Non-Functional Requirements**
- 6. ENTITIES**
- 7. FUNCTIONAL DEPENDENCIES**
- 8. DB DESIGN AND UML DIAGRAM**
- 9. ER/EER DIAGRAM**
- 10. FINAL RELATIONAL SCHEMA**
- 11. SQL STATEMENTS**
- 12. BACKEND CODES**
- 13. FRONTEND CODES**
- 14. GLIMPSE OF THE APPLICATION**
- 15. CONCLUSION**

1. INTRODUCTION

We have chosen to produce a Car Buy and Rental application. Our customers can buy or rent a car according to their wish. There is a list of cars available with the model, top speed, price and other details. The customer can choose his favorite one from the list of cars available. In terms of renting the car, the customer can choose a car from the available ones and can rent it for whatever duration they want. Simultaneously, once a customer begins with the finalization process, entering his/her details, a database is created at our end and we can have a record of the information he gave while entering the details.



2. GOALS OF THE PROJECT

Our Application wants to give 'Online Car Buy and rental solutions' to our customers. Customers will have a variety of choices to select from, to either buy or rent their favorite car. It wants to give a complete solution for Buying/Renting services through the internet. Major goal is to make the application so efficient that it attracts more customers to come on our platform.

3. PROBLEM IN EXISTING SYSTEMS

- > There is no platform in which a user can Buy and Rent a car in one application only.
- > For buying a car, different companies have their own different webpages, but there is no centralized data of different cars in the same budget of different companies.
- > Updates and changes in details is a tedious task.

4. PROPOSED SYSTEM

The proposed system overcomes the problems in the existing system. In this system, data is centralized and is maintained electronically and it's easy to update the details, which has overcome the problem in the previous systems.

5. REQUIREMENTS

- a) Our application should have a collection of cars.
- b) Customer, based on his willingness, can buy or rent a car.
- c) Based on his preference, a list of cars available to buy or rent will be shown along with available date and time (from and to).
- d) Customers will select a car from the suggestions and should be able to reserve it for buy or rent.
- e) Simultaneously a database will be generated which will collect and store all the information of the user provided by him at the time of disclosure.
- f) Billing is generated when a car is returned.
- g) Once the car is returned it becomes available for the booking.

-> Functional Requirements:

- The system shall enable the admin to view, create, edit, and delete the cars.
- The system allows generation of reports for the order placed.
- The system shall allow the admin to update additional information for a given car.
- The system shall enable the customer to view the products menu , create an account , login to the system, and place an order.
- The system shall display the car ordered , the car buy/rental price and the total amount to be paid
- The system shall provide visual confirmation of the order placement.

-> Non- functional Requirements:



6. ENTITIES

a) **Customer:** Customer will be the one who is using our application for buying or reserving a car. Customer entities will store details like customer driving license number, email, address, name, and phone number.



b) **Car:** Car entities will have a list of cars available in the system. Each car will have attributes like make, model, mileage, top speed etc. Car will also have a separate flag to check the availability of the car.



c) **Booking**: Each car reservation will be monitored in the entity called booking. Booking will have attributes like booking id, from date and time of booking and due return date and time and actual return date and time of the booking, and booking status.



d) **Billing**: When a customer returns a car, a bill will be generated on the particular booking. Billing has attributes like Bill ID, bill date, bill status, total late fee, tax amount, and total amount.



7. FUNCTIONAL DEPENDENCIES

Model → make
Make, color → model
Make, msrp → model, year
Model, color → make
Model, year → make, msrp
Model, msrp → make, year
Color, msrp → make, model, year
Year, msrp → make, model

a) Customer_Details Relation: DL_number -> Fname, Mname, Lname, Phone_number, Email_id, Street, City, State, Zip Code Zipcode -> State, City

b) Car Relation: Registration_number -> Model, Make, Model_year, Mileage, Availability_flag Model -> Make

c) Car_Category Relation: Category_name -> No_of_person, Cost_per_day, Late_fee_per_hour

e) Booking_Details Relation: Booking_id -> From_dt_time, Ret_dt_time, Amount, Booking_status, Reg_num, DL_num, Act_ret_dt_time,

f) Billing_Details Relation: Bill_id -> Bill_date, Bill_status, Total_amt, Tax_amt, Booking_id, Total_late_fee

8. DB DESIGN AND UML DIAGRAM

1. Customer Details Relation

Customer
DL Number First Name Second Name Phone Number E-mail ID Street City State Zip Code

2. Buying Table:

Buying Table
ID Company Model Car color Car top speed Car mileage Total Cost Status

3. Rental Table:

Rental Table
ID Company Model Car color Car top speed Car mileage Rent cost per day Status

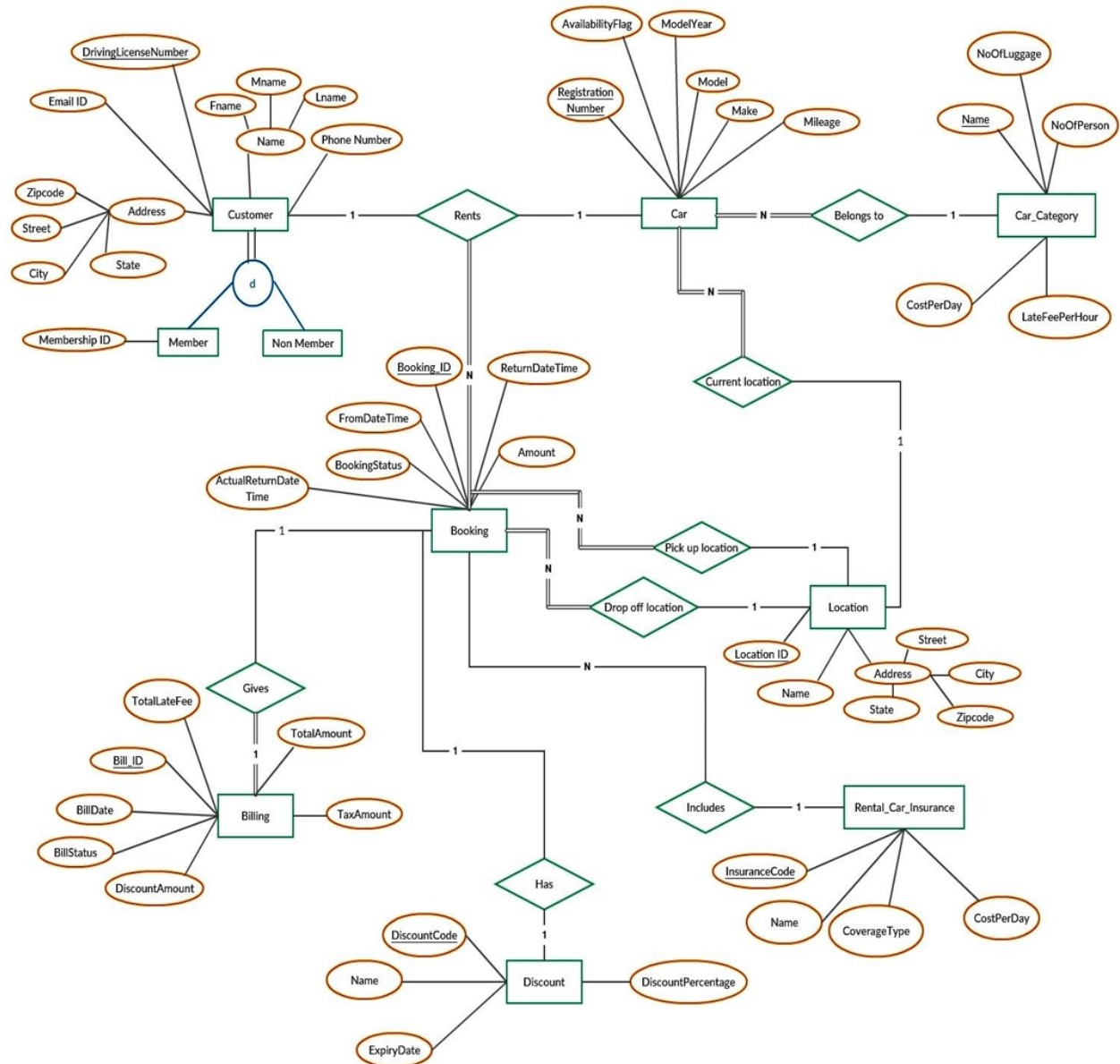
4. Booking Details:

Booking Details
Booking ID From date time Rent date time Amount Booking status Registration number DL number Act rent date time

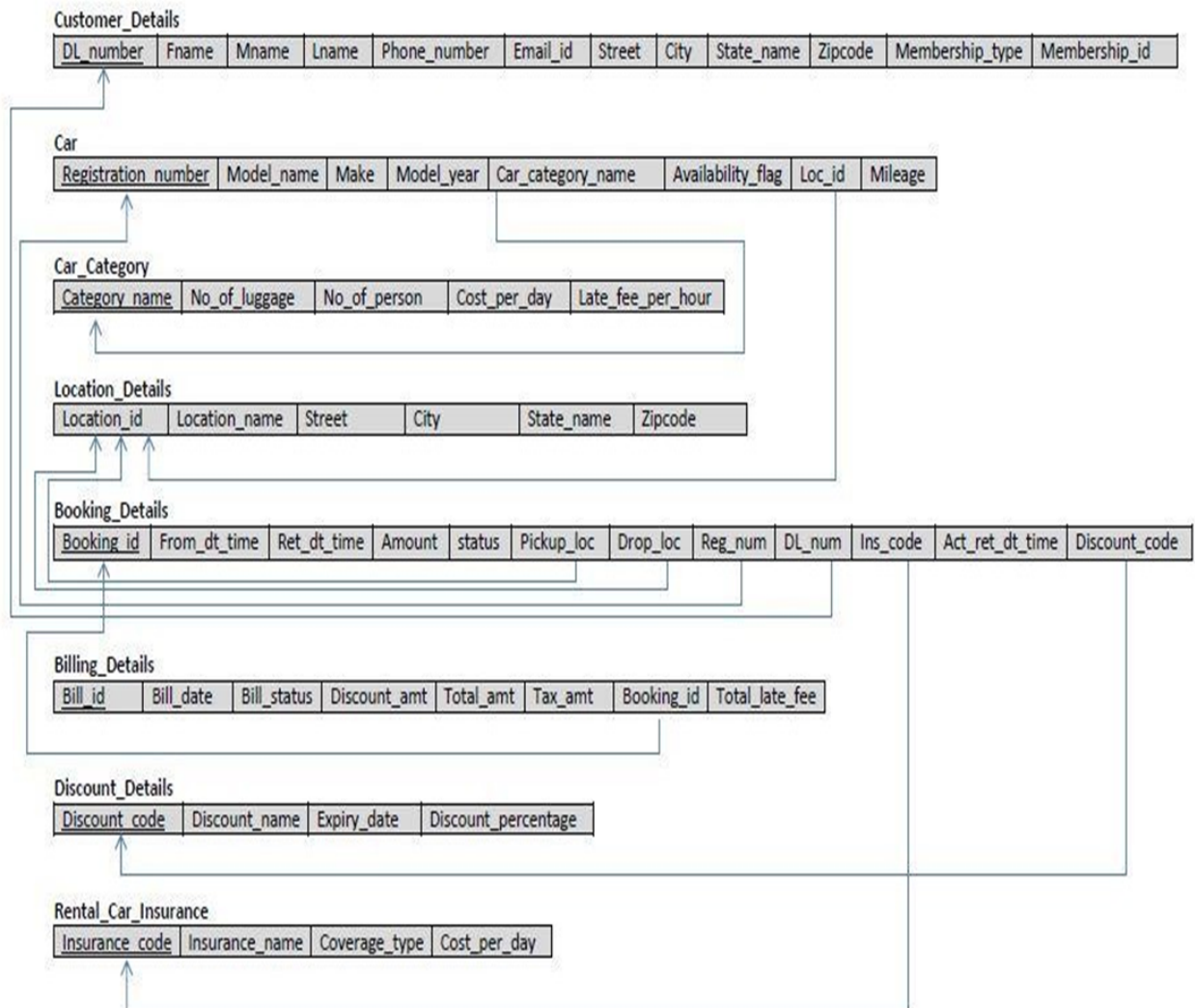
5. Billing Details:

Booking Details
Bill ID
Bill date
Bill status
Total tax amount
Booking ID
Total late fee

9. ER/EER DIAGRAM

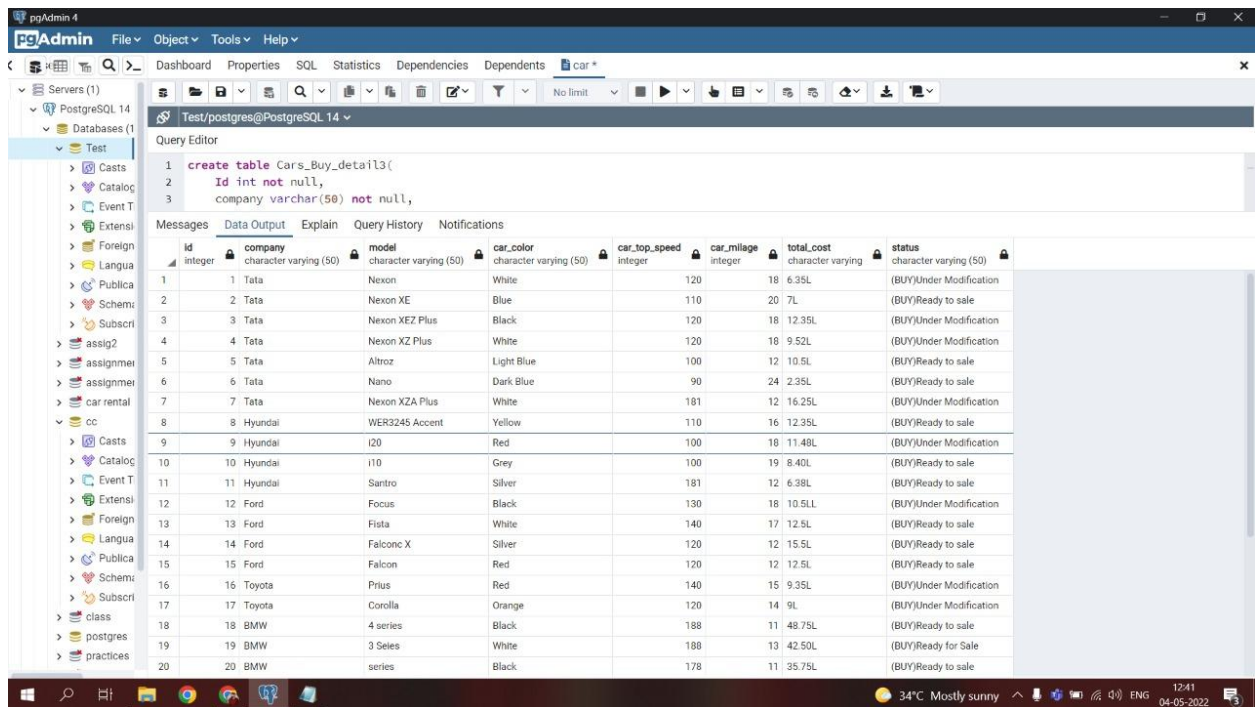


10. FINAL RELATIONAL SCHEMA



11. SQL STATEMENTS

For Buying the Car:



The screenshot shows the pgAdmin 4 interface. On the left, the 'Servers' tree shows 'PostgreSQL 14' with a database 'Test'. The 'Query Editor' is open, showing the SQL command to create the table 'Cars_Buy_detail3'. Below the editor, the 'Messages' tab shows the execution results, which are displayed as a table with 20 rows.

id	company	model	car_color	car_top_speed	car_milage	total_cost	status
1	Tata	Nexon	White	120	18	6.35L	(BUY)Under Modification
2	Tata	Nexon XE	Blue	110	20	7L	(BUY)Ready to sale
3	Tata	Nexon XEZ Plus	Black	120	18	12.35L	(BUY)Under Modification
4	Tata	Nexon XZ Plus	White	120	18	9.52L	(BUY)Under Modification
5	Tata	Altroz	Light Blue	100	12	10.5L	(BUY)Ready to sale
6	Tata	Nano	Dark Blue	90	24	2.35L	(BUY)Ready to sale
7	Tata	Nexon XZA Plus	White	181	12	16.25L	(BUY)Under Modification
8	Hyundai	WER3245 Accent	Yellow	110	16	12.35L	(BUY)Ready to sale
9	Hyundai	i20	Red	100	18	11.48L	(BUY)Under Modification
10	Hyundai	i10	Grey	100	19	8.40L	(BUY)Ready to sale
11	Hyundai	Santro	Silver	181	12	6.38L	(BUY)Ready to sale
12	Ford	Focus	Black	130	18	10.5LL	(BUY)Under Modification
13	Ford	Fista	White	140	17	12.5L	(BUY)Ready to sale
14	Ford	Falconc X	Silver	120	12	15.5L	(BUY)Ready to sale
15	Ford	Falcon	Red	120	12	12.5L	(BUY)Ready to sale
16	Toyota	Prius	Red	140	15	9.35L	(BUY)Under Modification
17	Toyota	Corolla	Orange	120	14	9L	(BUY)Under Modification
18	BMW	4 series	Black	188	11	48.75L	(BUY)Ready to sale
19	BMW	3 Seies	White	188	13	42.50L	(BUY)Ready for Sale
20	BMW	series	Black	178	11	35.75L	(BUY)Ready to sale

Car Data:-

insert into Cars_Buy_detail
values

('1','Tata', 'Nexon','White','120','18','6.35L','(BUY)Under Modification'),
('2','Tata','Nexon XE','Blue','110','20','7L','(BUY)Ready to sale'),
('3','Tata','Nexon XEZ Plus','Black','120','18','12.35L','(BUY)Under Modification'),
('4','Tata','Nexon XZ Plus','White','120','18','9.52L','(BUY)Under Modification'),
('5','Tata','Altroz','Light Blue','100','12','10.5L','(BUY)Ready to sale'),
('6','Tata','Nano','Dark Blue','90','24','2.35L','(BUY)Ready to sale'),
('7','Tata','Nexon XZA Plus','White','181','12','16.25L','(BUY)Under Modification'),
('8','Hyundai','WER3245 Accent','Yellow','110','16','12.35L','(BUY)Ready to sale'),
('9','Hyundai','i20','Red','100','18','11.48L','(BUY)Under Modification'),

```

('10','Hyundai','i10','Gray','100','19','8.40L','(BUY)Ready to sale'),
('11','Hyundai','Santro','Silver','181','12','6.38L','(BUY)Ready to sale'),
('12','Ford','Focus','Black','130','18','10.5LL','(BUY)Under Modification'),
('13','Ford','Fista','White','140','17','12.5L','(BUY)Ready to sale'),
('14','Ford','Falcon X','Silver','120','12','15.5L','(BUY)Ready to sale'),
('15','Ford','Falcon','Red','120','12','12.5L','(BUY)Ready to sale'),
('16','Toyota','Prius','Red','140','15','9.35L','(BUY)Under Modification'),
('17','Toyota','Corolla','Orange','120','14','9L','(BUY)Under Modification'),
('18','BMW','4 series','Black','188','11','48.75L','(BUY)Ready to sale'),
('19','BMW','3 Series ','White','188','13','42.50L','(BUY)Ready for Sale'),
('20','BMW','series','Black','178','11','35.75L','(BUY)Ready to sale'),
('21','Maruti','Alto 800','Orange','100','21','5.03L','(BUY)Ready to sale'),
('22','Maruti','S-Pres','Light Orange','110','12','5.64L','(BUY)Under Modification'),
('23','MERCEDES BENZ','GLA','White','181','12','46.5L','(BUY)Under Modification'),
('24','Honda','Civic','Dark Blue','120','14','8.19L','(BUY)Ready to sale'),
('25','Renault','KWID','White','100','22','5.83L','(BUY)Ready to sale'),
('26','Datsun','Redi-GO','Blue','100','23','4.96L','(BUY)Ready to sale'),
('27','Skoda ','Kushaq','Orange','120','18','18.19L','(BUY)Ready to sale'),
('28','Dodge','Avenger','White','150','12','18.5L','(BUY)Under Modification'),
('29','Ferrari','2 series','Black','158','14','58.68L','(BUY)Ready for Sale'),
('30','Chevrolet','Sail Hatchback','Silver','120','18','10L','(BUY)Under Modification');

```

```
select * from Cars_Buy_detail;
```

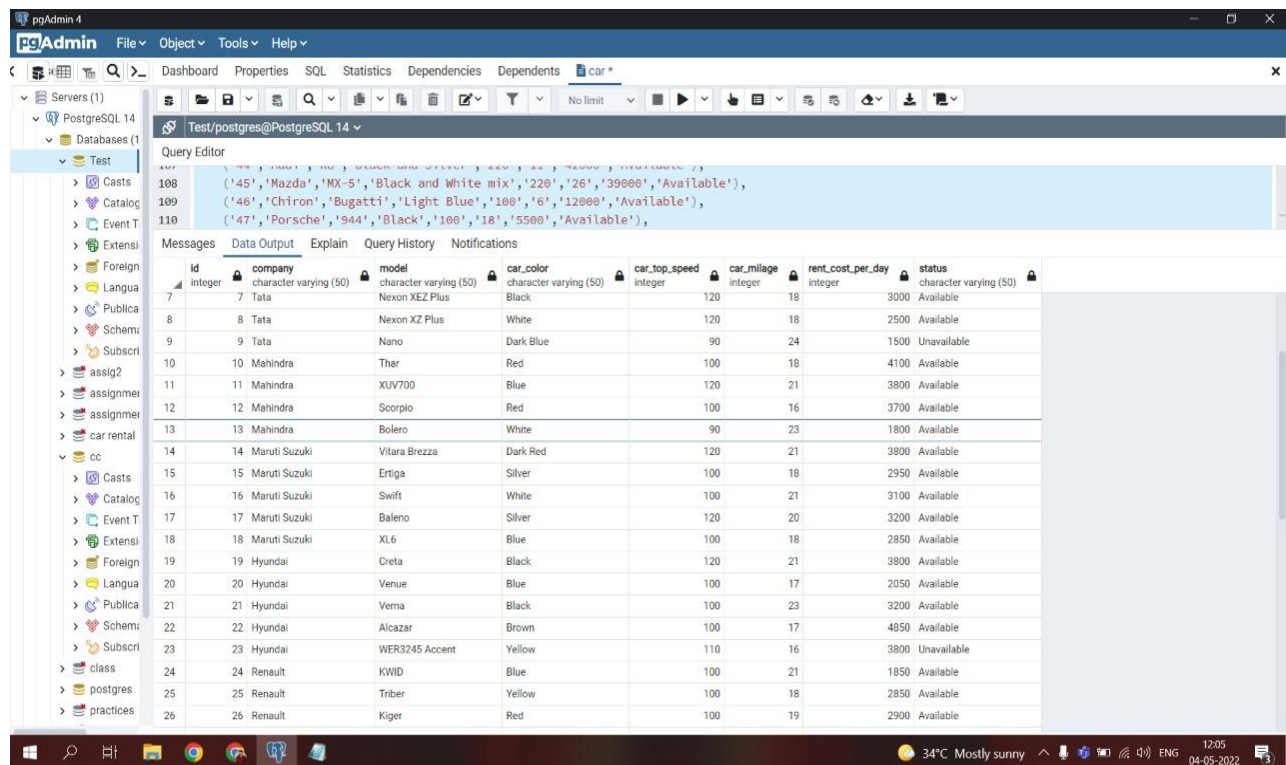
Car buy Query:-

```

create table Cars_Buy_detail(
  Id int not null,
    company varchar(50) not null,
    Model varchar(50) not null,
    car_color varchar(50) not null,
    car_top_speed int not null,
    car_milage int not null,
    Total_Cost varchar not null,
    status varchar(50) not null
);

```


For Renting the Car:



The screenshot shows the pgAdmin 4 interface with a PostgreSQL 14 database. The 'car' table is selected, and its data is displayed in the 'Data Output' tab. The table has 10 columns: id, company, model, car_color, car_top_speed, car_milage, rent_cost_per_day, and status. The data is as follows:

id	company	model	car_color	car_top_speed	car_milage	rent_cost_per_day	status
7	Tata	Nexon XEZ Plus	Black	120	18	3000	Available
8	Tata	Nexon XZ Plus	White	120	18	2500	Available
9	Tata	Nano	Dark Blue	90	24	1500	Unavailable
10	Mahindra	Thar	Red	100	18	4100	Available
11	Mahindra	XUV700	Blue	120	21	3800	Available
12	Mahindra	Scorpio	Red	100	16	3700	Available
13	Mahindra	Bolero	White	90	23	1800	Available
14	Maruti Suzuki	Vitara Brezza	Dark Red	120	21	3800	Available
15	Maruti Suzuki	Ertiga	Silver	100	18	2950	Available
16	Maruti Suzuki	Swift	White	100	21	3100	Available
17	Maruti Suzuki	Baleno	Silver	120	20	3200	Available
18	Maruti Suzuki	XL6	Blue	100	18	2850	Available
19	Hyundai	Creta	Black	120	21	3800	Available
20	Hyundai	Venue	Blue	100	17	2050	Available
21	Hyundai	Verna	Black	100	23	3200	Available
22	Hyundai	Alcazar	Brown	100	17	4850	Available
23	Hyundai	WER3245 Accent	Yellow	110	16	3800	Unavailable
24	Renault	KWID	Blue	100	21	1850	Available
25	Renault	Triber	Yellow	100	18	2850	Available
26	Renault	Kiger	Red	100	19	2900	Available

Car Data:-

insert into Rental_cars_detail
values

('1','Tata','Nexon','Orange','100','22','2800','Available'),
('2','Tata','Harrier','Red','100','21','3500','Available'),
('3','Tata','Punch','Blue','120','17','2400','Available'),
('4','Tata','Safari','Navy Blue','110','21','5200','Available'),
('5','Tata','Nexon','White','120','18','2850','Unavailable'),
('6','Tata','Nexon XE','Blue','110','20','2200','Available'),
('7','Tata','Nexon XEZ Plus','Black','120','18','3000','Available'),
('8','Tata','Nexon XZ Plus','White','120','18','2500','Available'),
('9','Tata','Nano','Dark Blue','90','24','1500','Unavailable'),
('10','Mahindra','Thar','Red','100','18','4100','Available'),
('11','Mahindra','XUV700','Blue','120','21','3800','Available'),

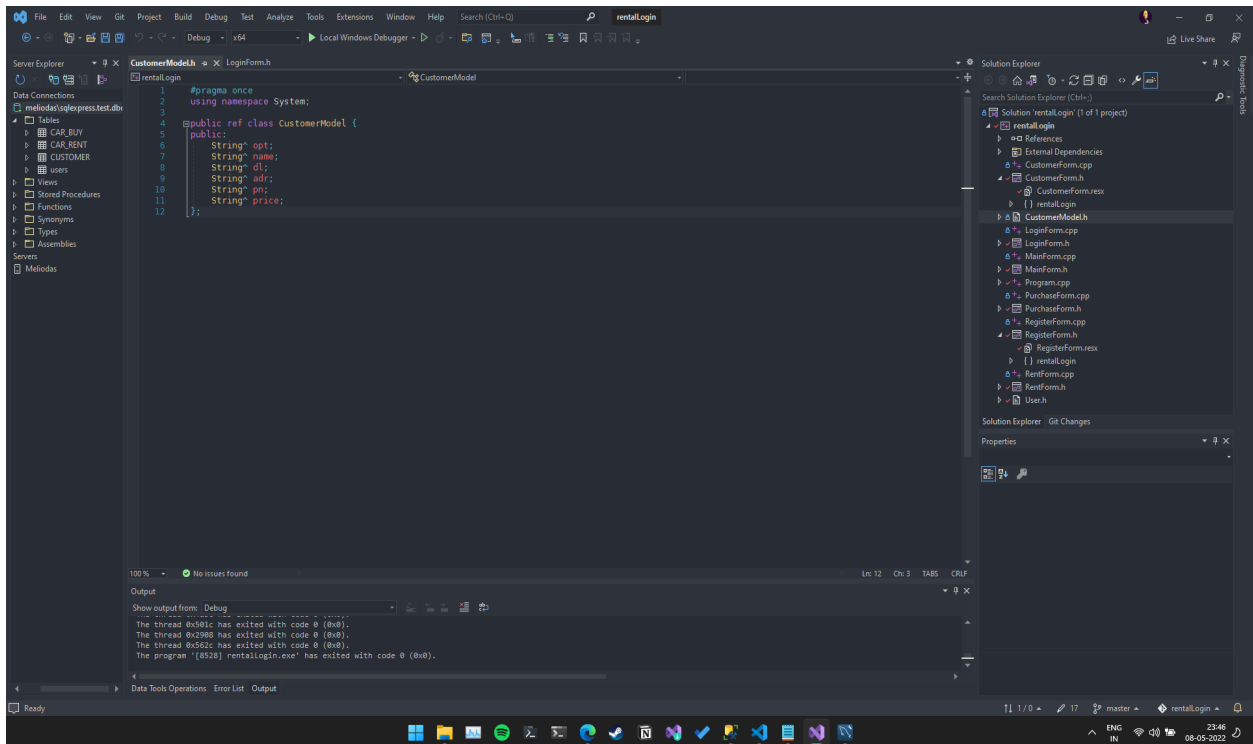
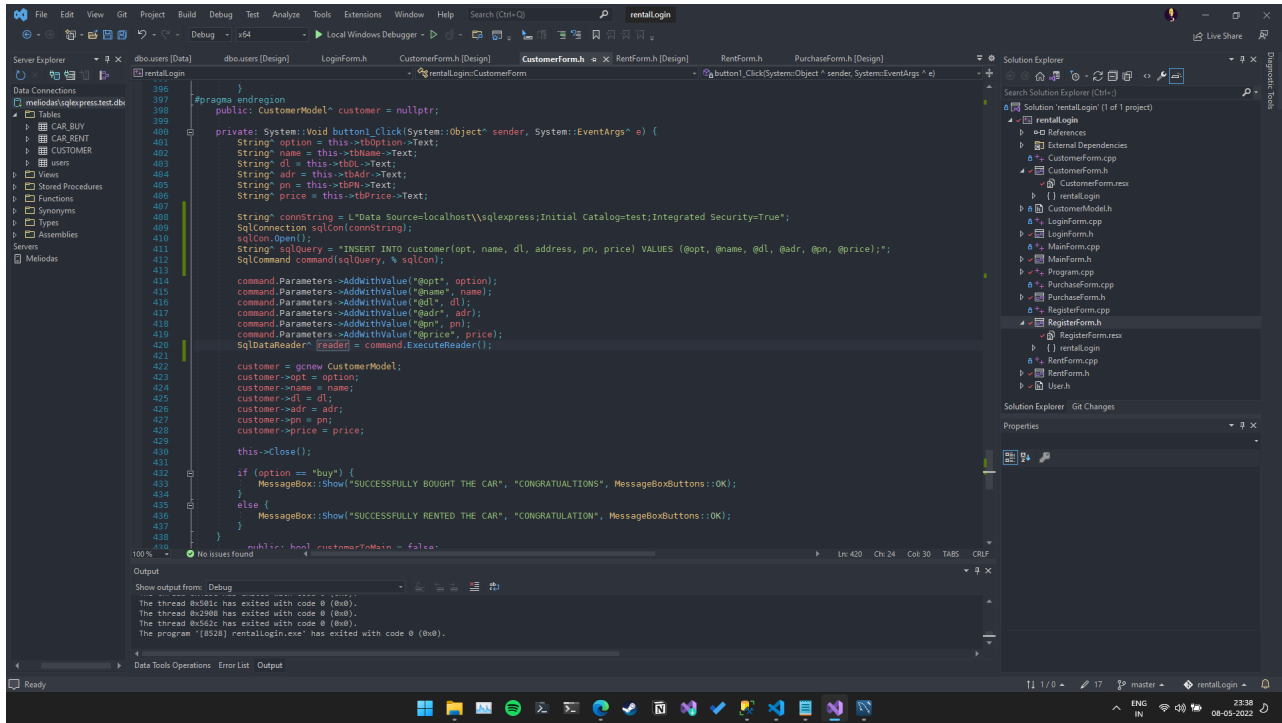
('12','Mahindra','Scorpio','Red','100','16','3700','Available'),
 ('13','Mahindra','Bolero','White','90','23','1800','Available'),
 ('14','Maruti Suzuki','Vitara Brezza','Dark Red','120','21','3800','Available'),
 ('15','Maruti Suzuki','Ertiga','Silver','100','18','2950','Available'),
 ('16','Maruti Suzuki','Swift','White','100','21','3100','Available'),
 ('17','Maruti Suzuki','Baleno','Silver','120','20','3200','Available'),
 ('18','Maruti Suzuki','XL6','Blue','100','18','2850','Available'),
 ('19','Hyundai','Creta','Black','120','21','3800','Available'),
 ('20','Hyundai','Venue','Blue','100','17','2050','Available'),
 ('21','Hyundai','Verna','Black','100','23','3200','Available'),
 ('22','Hyundai','Alcazar','Brown','100','17','4850','Available'),
 ('23','Hyundai','WER3245 Accent','Yellow','110','16','3800','Unavailable'),
 ('24','Renault','KWID','Blue','100','21','1850','Available'),
 ('25','Renault','Triber','Yellow','100','18','2850','Available'),
 ('26','Renault','Kiger','Red','100','19','2900','Available'),
 ('27','BMW','3 Series ','White','188','13','25000','Available'),
 ('28','BMW','4 series','Black','188','11','20500','Available'),
 ('29','Ferrari','2 series','Black','158','14','28500','Available'),
 ('30','Ferrari','Lexus LC','Yellow','250','9','44500','Available'),
 ('31','Chevrolet','Sail Hatchback','Silver','120','18','150','Available'),
 ('32','Chevrolet','Corvette Z06','Yellow','210','14','40000','Unavailable'),
 ('33','Ford','Focus','Black','130','18','4000','Available'),
 ('34','Ford','Falcon','Red','120','12','4500','Available'),
 ('35','Ford','Ford Focus SVT','Red','100','21','3500','Available'),
 ('36','Toyota','Prius','Red','140','15','4200','Available'),
 ('37','Toyota','Toyota MR2','Blue','100','17','4800','Available'),
 ('38','Toyota','Celica','Red','100','17','3500','Unavailable'),
 ('39','Jaguar','Ford Mustang','Red','120','13','15000','Unavailable'),
 ('40','Jaguar','F-TYPE','Blue','210','10','44000','Available'),
 ('41','Dodge','Avenger','White','150','12','3720','Available'),
 ('42','MERCEDES BENZ','GLA','White','181','12','29000','Unavailable'),
 ('43','Lamborghini','Huracán','Black','220','10','45500','Available'),
 ('44','Audi','R8','Black and Silver','220','11','42000','Available'),
 ('45','Mazda','MX-5','Black and White mix','220','26','39000','Available'),
 ('46','Chiron','Bugatti','Light Blue','100','6','12000','Available'),
 ('47','Porsche','944','Black','100','18','5500','Available'),
 ('48','Subaru Impreza','2.5RS','Orange','100','14','4500','Available'),
 ('49','Honda','Civic Si','Yellow','100','18','4500','Unavailable'),
 ('50','Nissan','350Z','Silver','100','18','2500','Available');

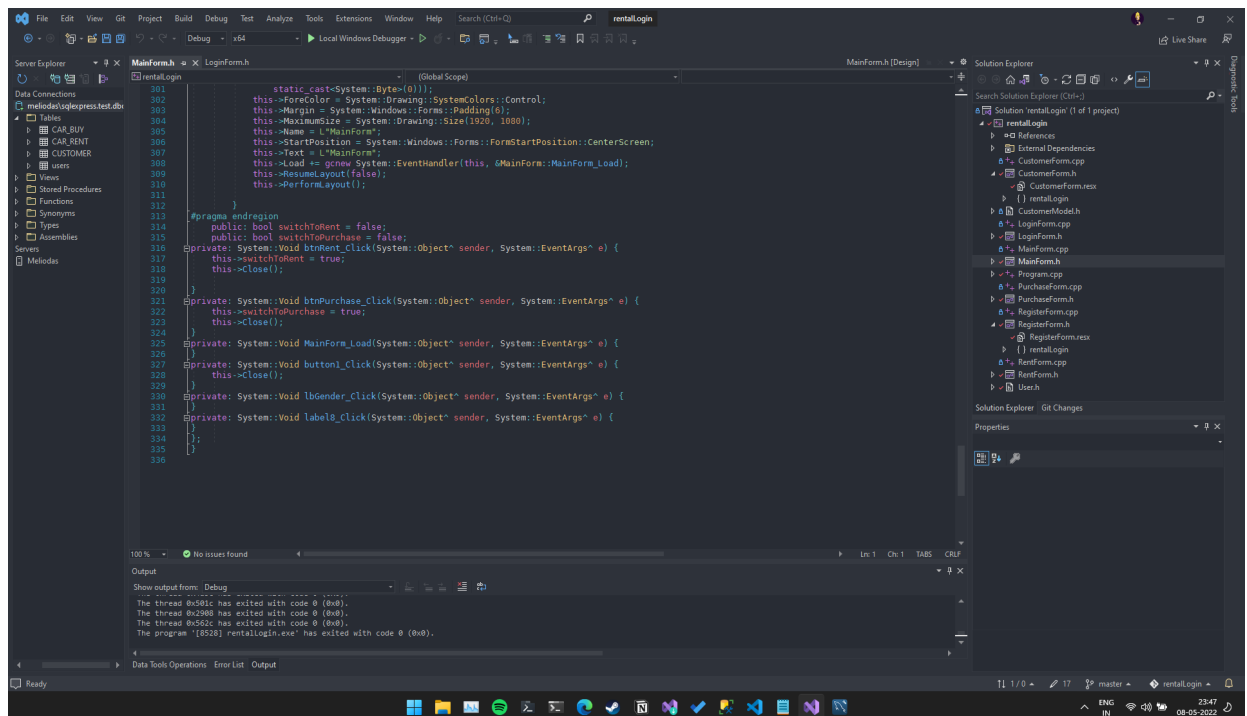
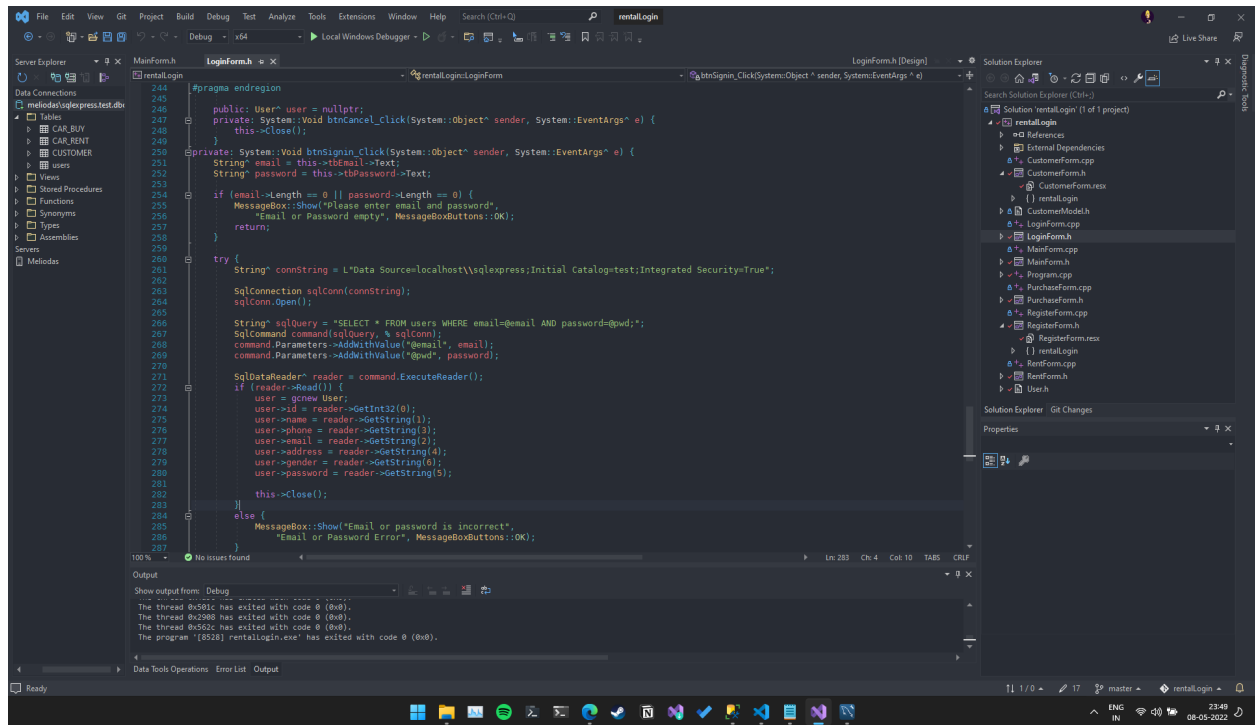
select * from Rental_cars_detail;

Car Rent Query:-

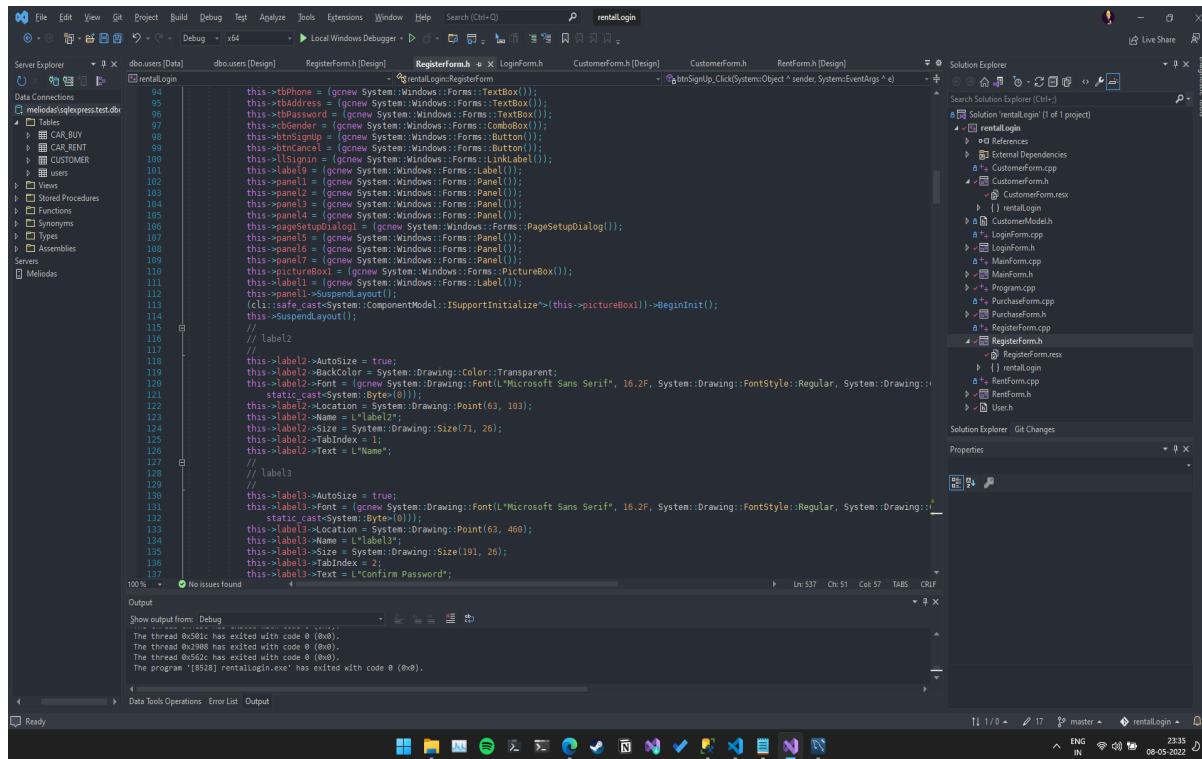
```
create table Rental_cars_detail(  
  Id int not null,  
    company varchar(50) not null,  
  Model varchar(50) not null,  
    car_color varchar(50) not null,  
    car_top_speed int not null,  
    car_milage int not null,  
    rent_cost_per_day int not null,  
  status varchar(50) not null  
);
```

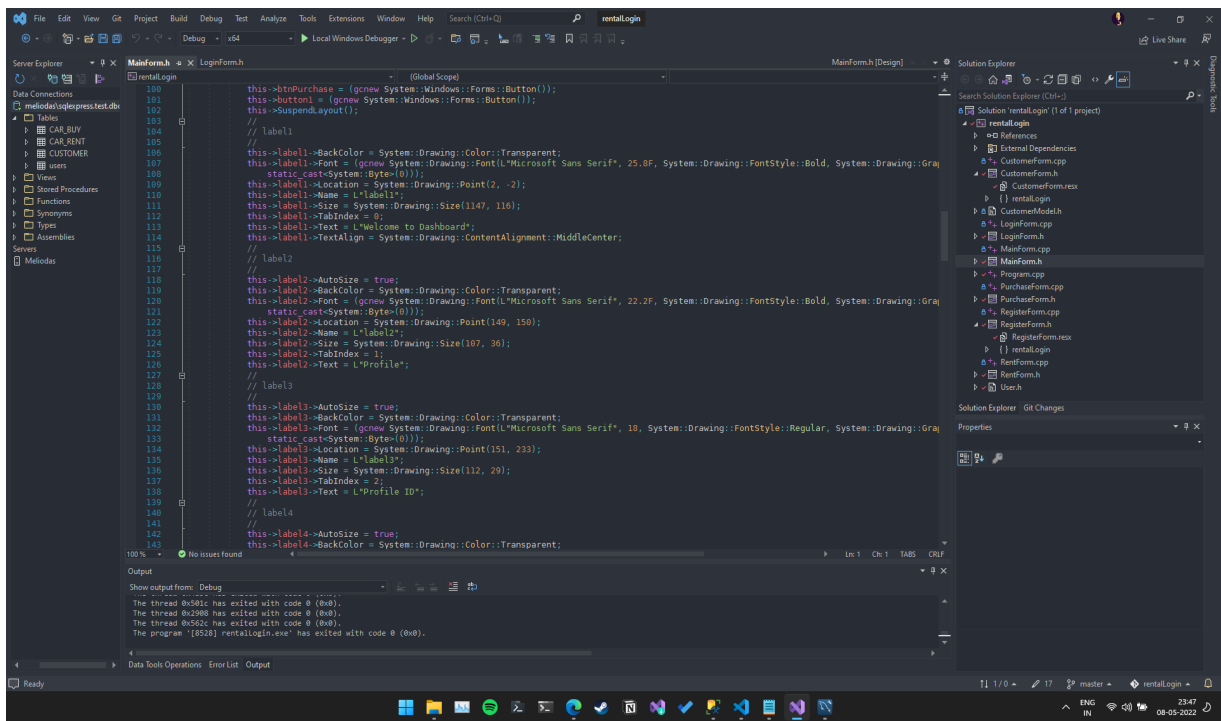
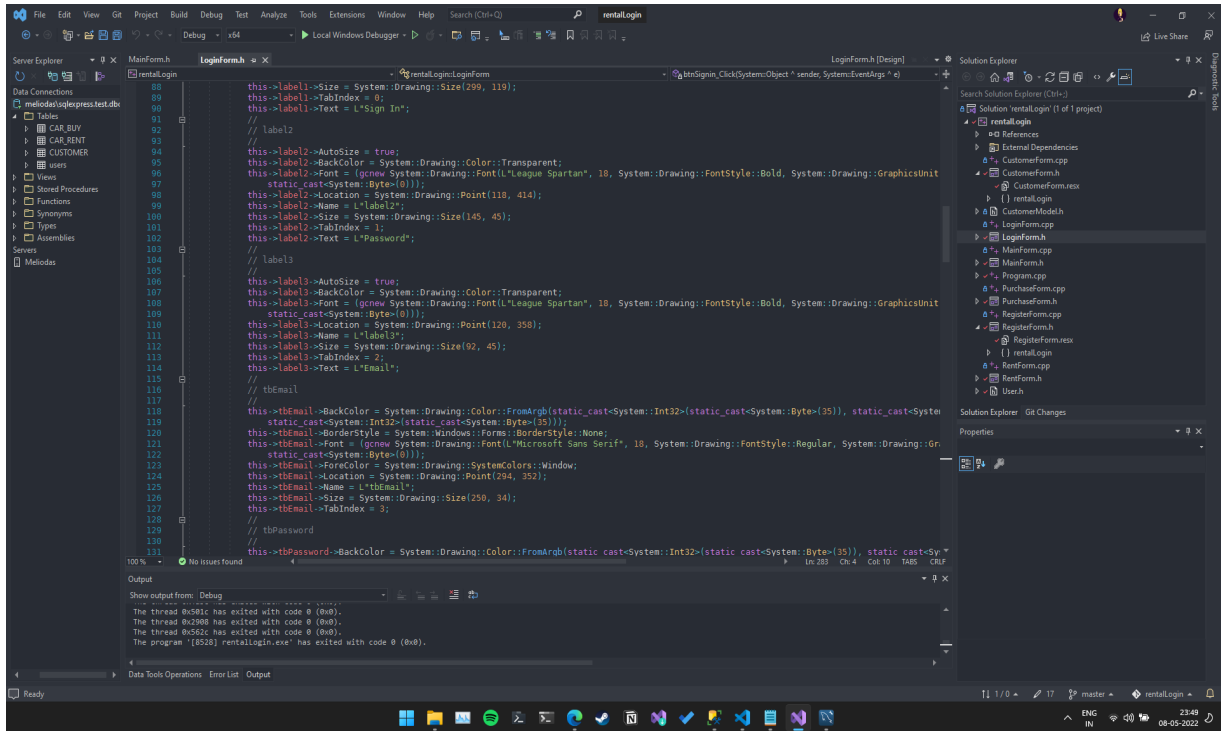
12. BACKEND CODES:

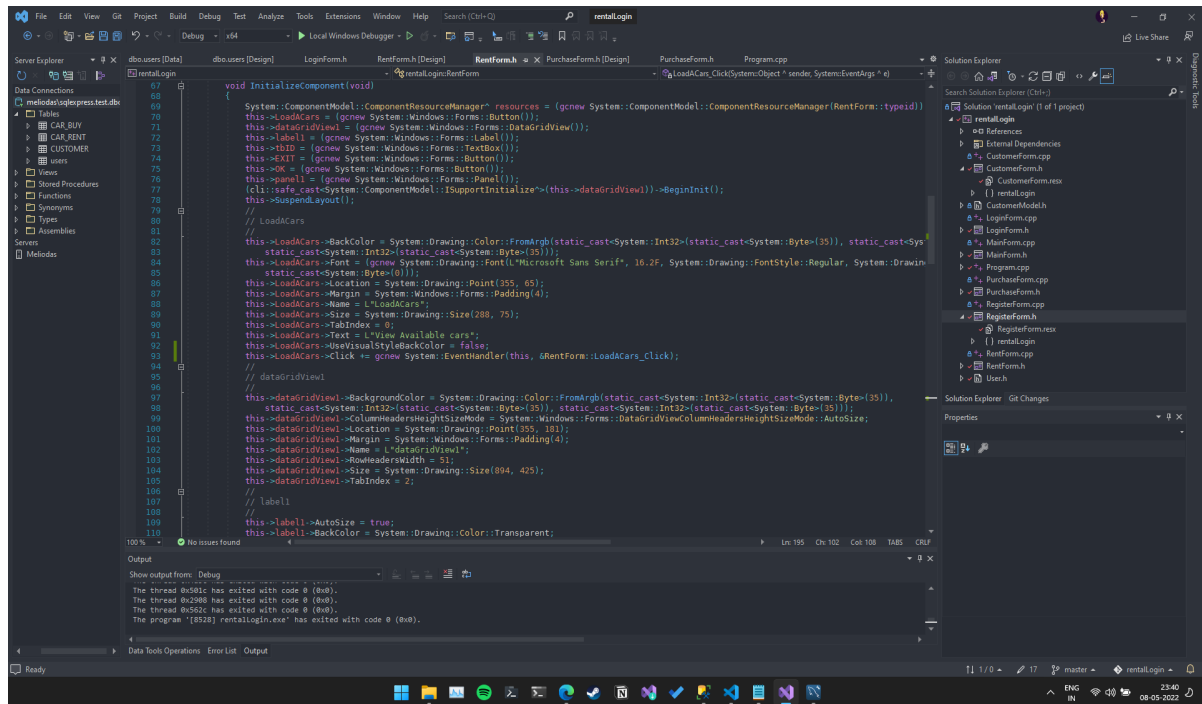




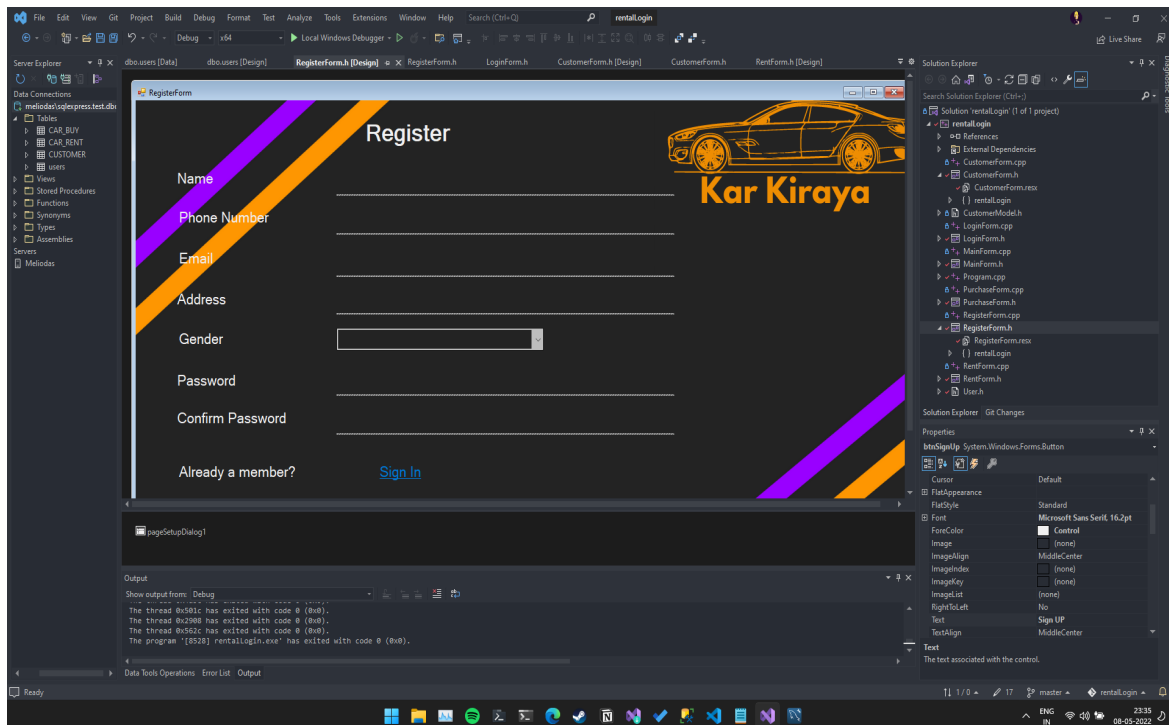
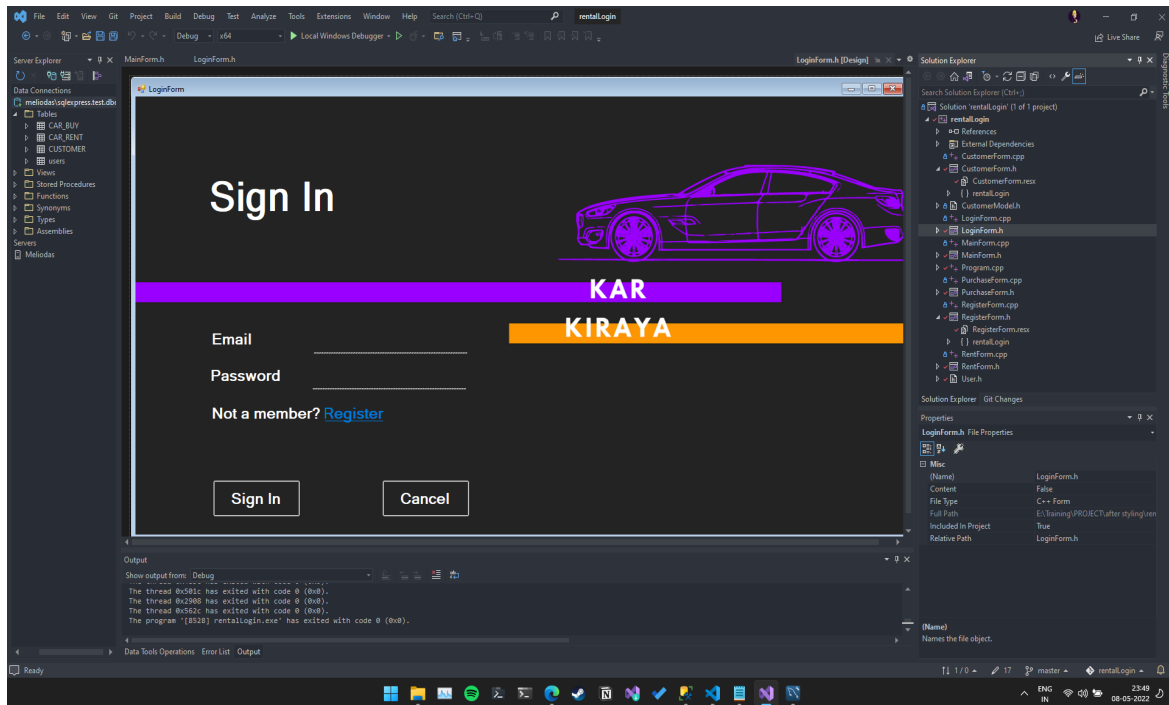
13. FRONTEND CODES:

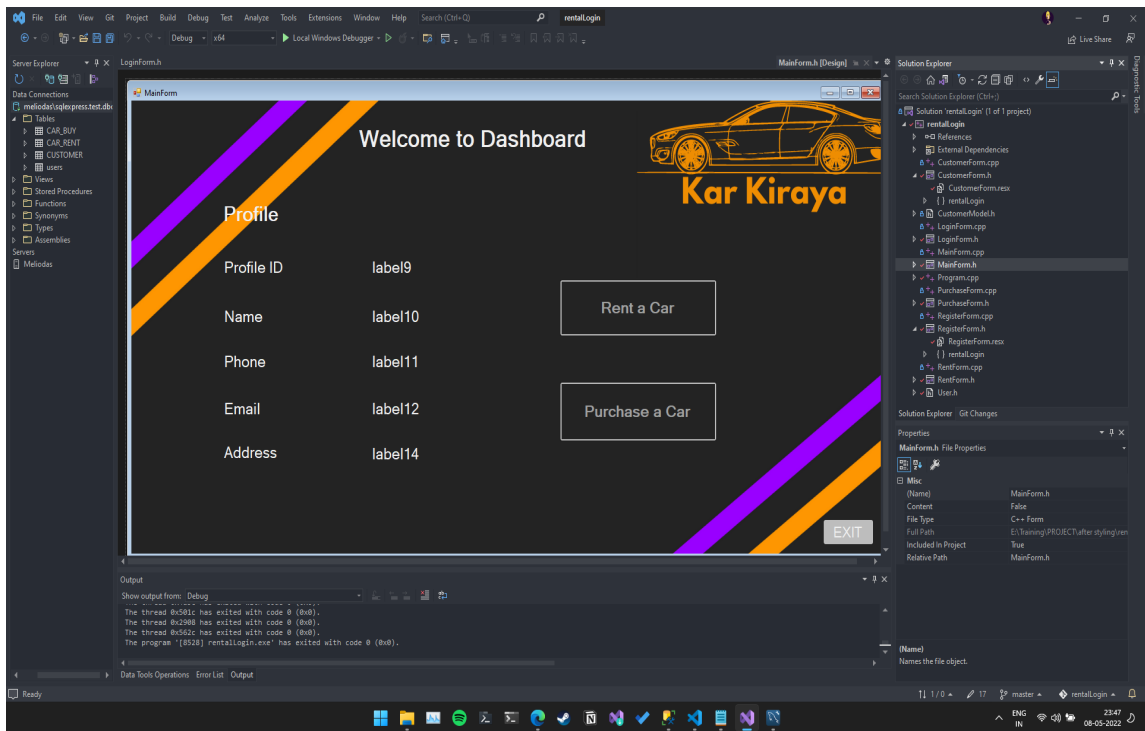
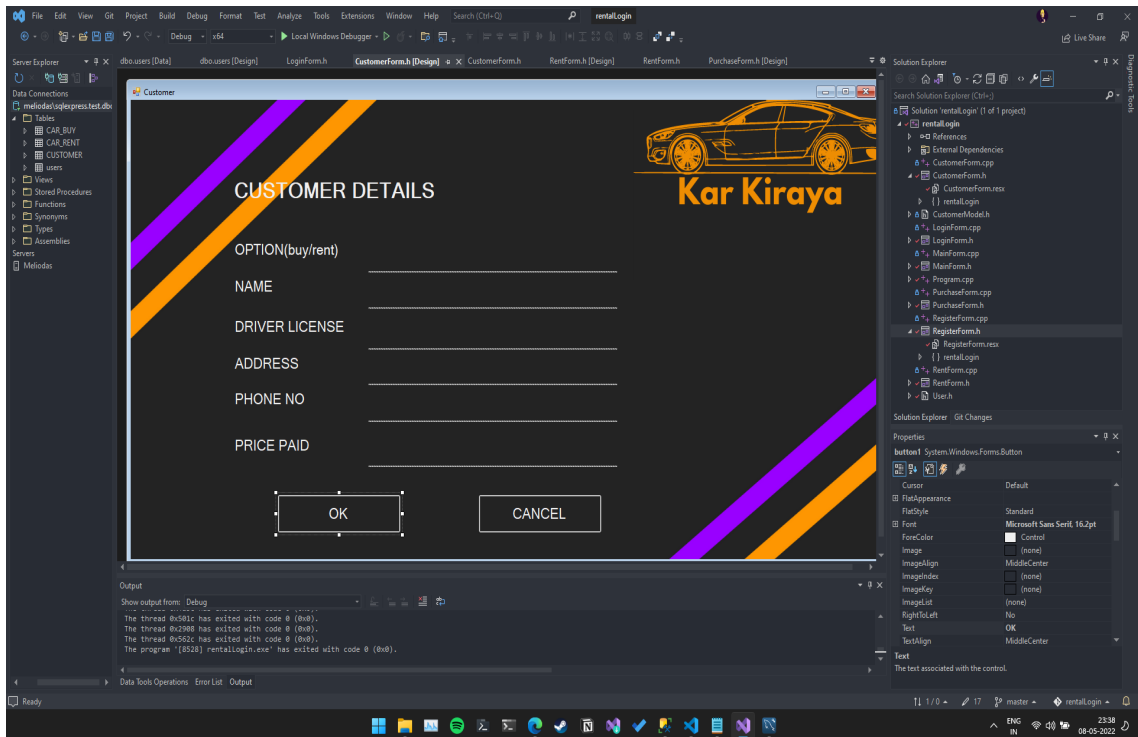


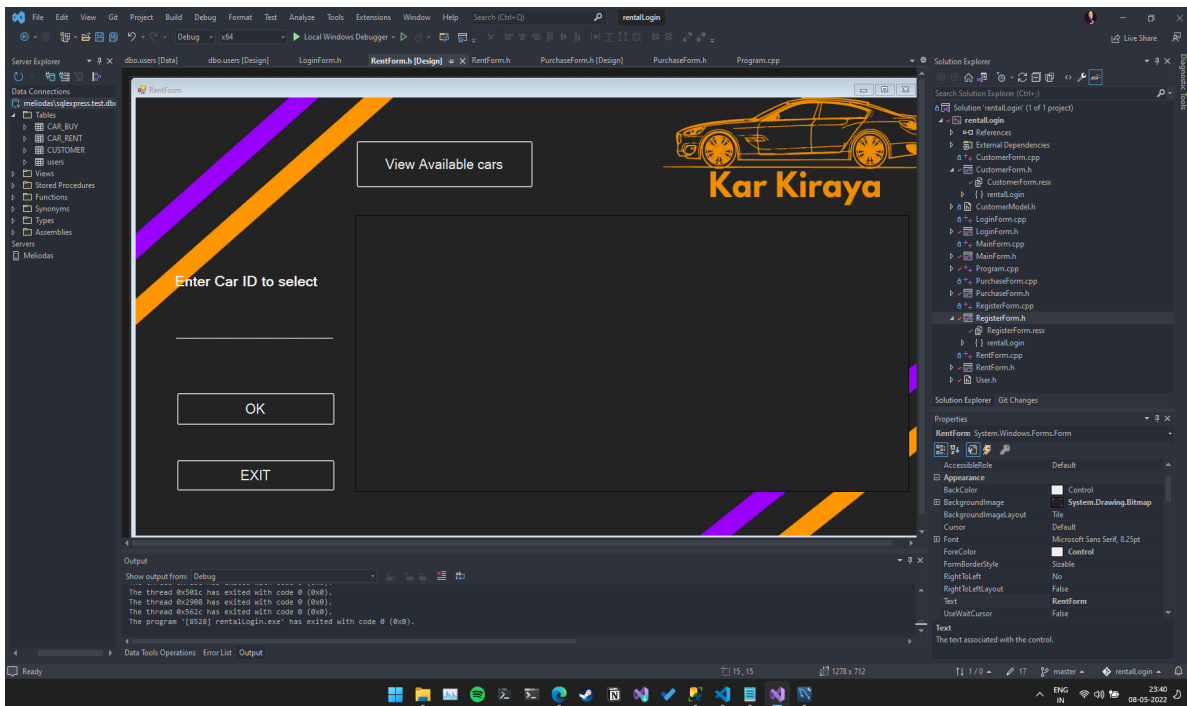
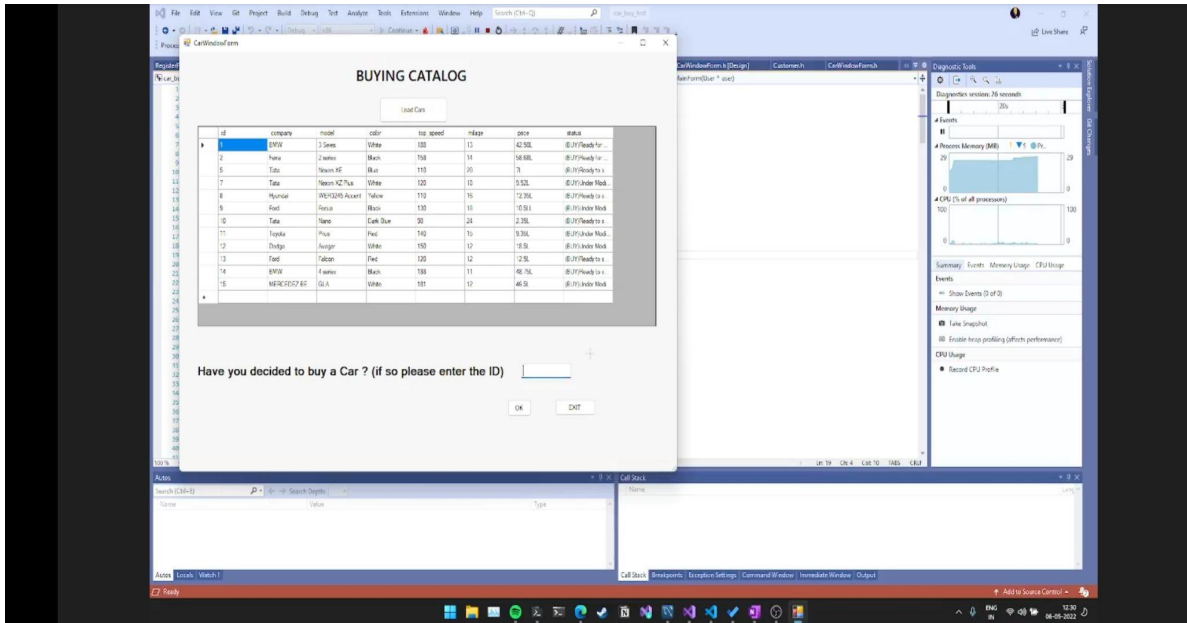




14. APPLICATION GLIMPSE:







15. CONCLUSION

During the course of this project, we learnt about practices that go into creating a database, and good knowledge about the front end part. We learnt how to design an application from a Database perspective and how to efficiently store and manipulate data.