TANDON SCHOOL OF ENGINEERING

CS-GY 6083 - Section B, Spring 2023

PROJECT PART II

Date of submission:
12/10/2023

Team Members:

Aditya Shyamsundar - as16410

Dhairya Shah – ds6969

Gunjan Dayani – gd2275

Table of contents

Execute Summary

The business case revolves around ZappRental, a car rental company akin to Kyte or Enterprise, aiming to offer a wide range of vehicles and rental options to customers. The objective is to design a centralized relational database to collect and manage relevant data for ZappRental and build a web-accessible frontend that will allow users to access the service via their browser's system. This database will facilitate efficient operations, improve data consistency, and aid in analyzing business performance.

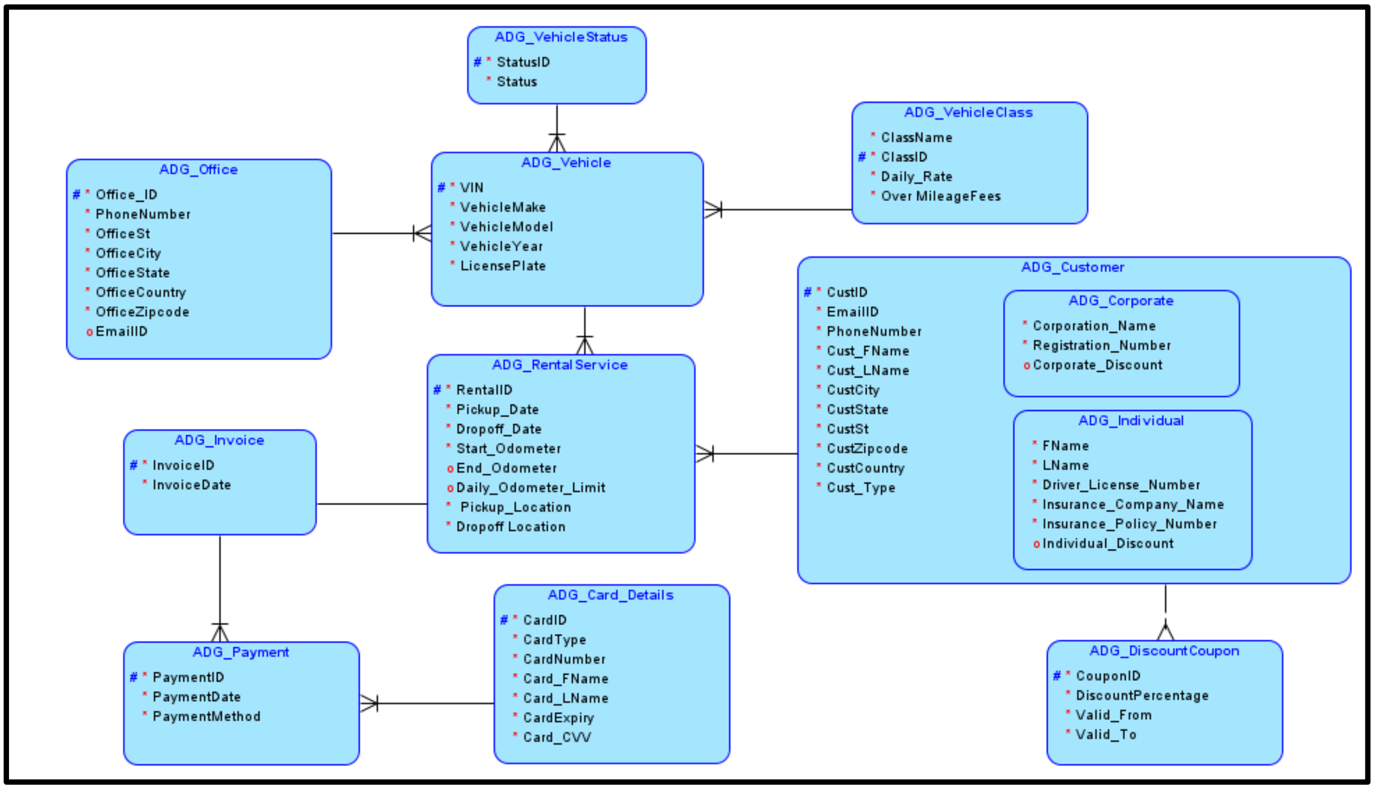
Approach:

- The first step was to analyze the business requirements for ZappRental, understanding the entities involved in the car rental process, their attributes, relationships, and the governing business rules. These entities include Customers, Vehicles, Rentals, Payments, Reservations, and Services.
- Each entity such as Customers, Vehicles, Rentals, Payments, Reservations, and Services has specific attributes and domain constraints. For instance, Vehicles include unique identifiers, make and model information, availability status, and rental rates.
- Relationships between entities were established, such as a Customer can have multiple Reservations, and each Reservation is linked to a specific Vehicle. These relationships ensure data integrity and consistency within the database.
- Data consistency was enforced using appropriate CHECK constraints on attributes, ensuring data adherence to specific rules and conditions, such as valid vehicle categories or reservation status options.
- Using tools like Oracle Data Modeler and MySQL Workbench, the schema for the centralized relational database system was designed. This schema details the structure and organization of the database, including tables, attributes, relationships, and constraints.
- A suitable database management system (MySQL and Oracle) was selected to implement the relational data model. The designed schema was implemented, tables were created, indexes defined, and data integrity ensured.
- A web-based user interface was implemented for the database schema, facilitating user interactions like registration, login, and managing rental-related activities. This interface caters to both customers and ZappRental employees, with tailored authorization levels for data access and modification.

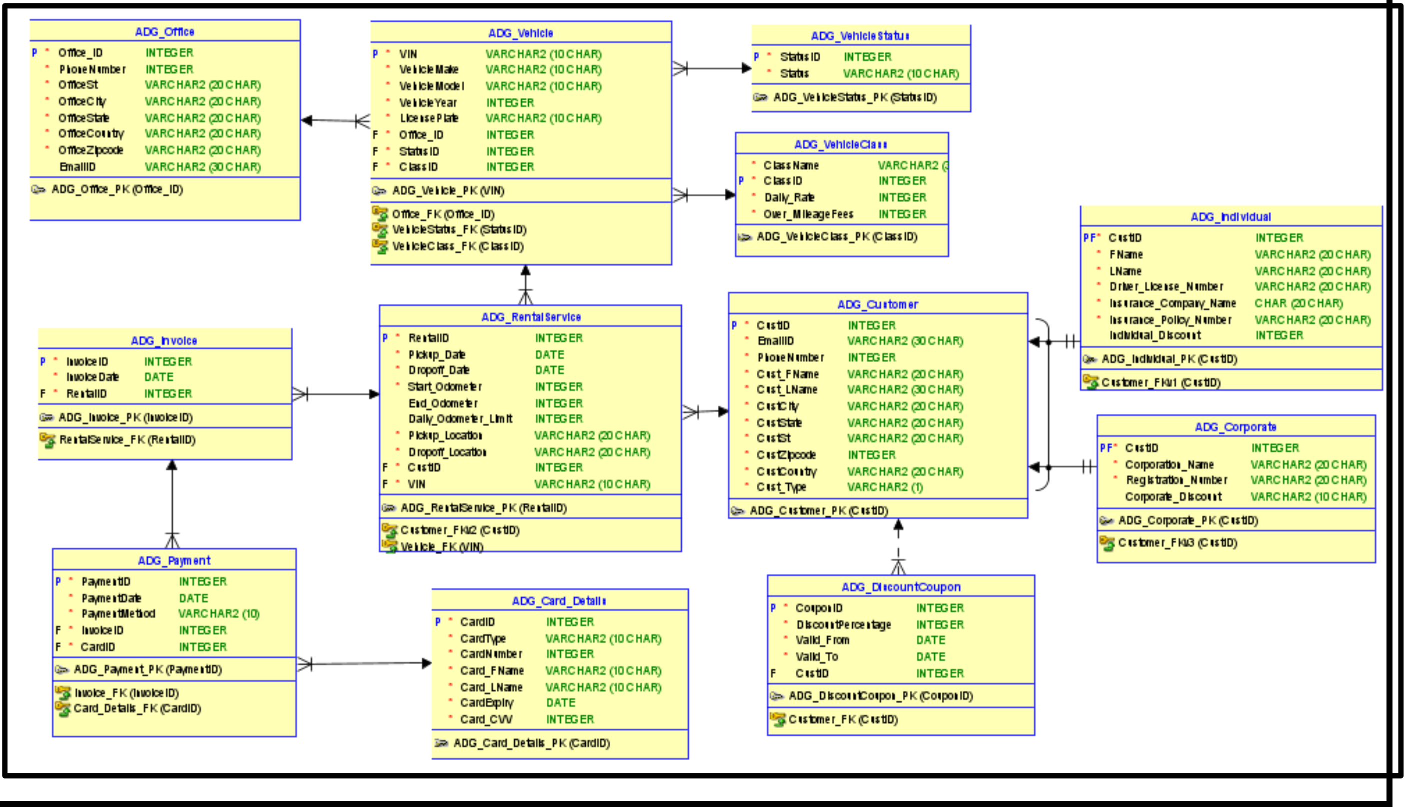
Business Performance Improvement:

- The centralized relational database system will significantly benefit ZappRental in several ways:
- Streamlining operations by providing a unified source for customer data, vehicle information, reservations, and payments, leading to efficient rental management.
- Enabling easy analysis and reporting on customer trends, vehicle utilization, and revenue streams, offering insights into customer behavior and vehicle performance.
- Facilitating effective financial management through comprehensive tracking of payments and rental transactions.
- Enhancing customer service with personalized rental experiences and promotions based on customer data, improving satisfaction and fostering loyalty.
- In conclusion, the centralized relational database system for ZappRental is a critical tool for enhancing business performance. It supports efficient operations, ensures data consistency, and provides valuable insights for strategic decision-making, ultimately driving growth and customer satisfaction in the car rental industry

Logical Model



Relational Model


ADG_Office
P * Office_ID INTEGER
* PhoneNumber INTEGER
* OfficeSt VARCHAR2 (20 CHAR)
* OfficeCity VARCHAR2 (20 CHAR)
* OfficeState VARCHAR2 (20 CHAR)
* OfficeCountry VARCHAR2 (20 CHAR)
* OfficeZipcode VARCHAR2 (20 CHAR)
EmailID VARCHAR2 (30 CHAR)
ADG_Office_PK (Office_ID)
ADG_Vehicle
P * VIN VARCHAR2 (10 CHAR)
* VehicleMake VARCHAR2 (10 CHAR)
* VehicleModel VARCHAR2 (10 CHAR)
* VehicleYear INTEGER
* LicensePlate VARCHAR2 (10 CHAR)
F * Office_ID INTEGER
F * StatusID INTEGER
F * ClassID INTEGER
ADG_Vehicle_PK (VIN)
Office_FK (Office_ID)
VehicleStatus_FK (StatusID)
VehicleClass_FK (ClassID)
ADG_VehicleStatus
P * StatusID INTEGER
* Status VARCHAR2 (10 CHAR)
ADG_VehicleStatus_PK (StatusID)
ADG_VehicleClass
* ClassName VARCHAR2 (
P * ClassID INTEGER
* Daily_Rate INTEGER
* Over_MileageFees INTEGER
ADG_VehicleClass_PK (ClassID)
ADG_Individual
PF* CustID INTEGER
* FName VARCHAR2 (20 CHAR)
* LName VARCHAR2 (20 CHAR)
* Driver_License_Number VARCHAR2 (20 CHAR)
* Insurance_Company_Name CHAR (20 CHAR)
* Insurance_Policy_Number VARCHAR2 (20 CHAR)
Individual_Discount INTEGER
ADG_Individual_PK (CustID)
Customer_FKv1 (CustID)
ADG_Invoice
P * InvoiceID INTEGER
* InvoiceDate DATE
F * RentalID INTEGER
ADG_Invoice_PK (InvoiceID)
RentalService_FK (RentalID)
ADG_RentalService
P * RentalID INTEGER
* Pickup_Date DATE
* Dropoff_Date DATE
* Start_Odometer INTEGER
End_Odometer INTEGER
Daily_Odometer_Limit INTEGER
* Pickup_Location VARCHAR2 (20 CHAR)
* Dropoff_Location VARCHAR2 (20 CHAR)
F * CustID INTEGER
F * VIN VARCHAR2 (10 CHAR)
ADG_RentalService_PK (RentalID)
Customer_FKv2 (CustID)
Vehicle_FK (VIN)
ADG_Customer
P * CustID INTEGER
* EmailID VARCHAR2 (30 CHAR)
* PhoneNumber INTEGER
* Cust_FName VARCHAR2 (20 CHAR)
* Cust_LName VARCHAR2 (30 CHAR)
* CustCity VARCHAR2 (20 CHAR)
* CustState VARCHAR2 (20 CHAR)
* CustSt VARCHAR2 (20 CHAR)
* CustZipcode INTEGER
* CustCountry VARCHAR2 (20 CHAR)
* Cust_Type VARCHAR2 (1)
ADG_Customer_PK (CustID)
ADG_Corporate
PF* CustID INTEGER
* Corporation_Name VARCHAR2 (20 CHAR)
* Registration_Number VARCHAR2 (20 CHAR)
Corporate_Discount VARCHAR2 (10 CHAR)
ADG_Corporate_PK (CustID)
Customer_FKv3 (CustID)
ADG_Payment
P * PaymentID INTEGER
* PaymentDate DATE
* PaymentMethod VARCHAR2 (10)
F * InvoiceID INTEGER
F * CardID INTEGER
ADG_Payment_PK (PaymentID)
Invoice_FK (InvoiceID)
Card_Details_FK (CardID)
ADG_Card_Details
P * CardID INTEGER
* CardType VARCHAR2 (10 CHAR)
* CardNumber INTEGER
* Card_FName VARCHAR2 (10 CHAR)
* Card_LName VARCHAR2 (10 CHAR)
* CardExpiry DATE
* Card_CVV INTEGER
ADG_Card_Details_PK (CardID)
ADG_DiscountCoupon
P * CouponID INTEGER
* DiscountPercentage INTEGER
* Valid_From DATE
* Valid_To DATE
F CustID INTEGER
ADG_DiscountCoupon_PK (CouponID)
Customer_FK (CustID)

Assumptions

- Rental rates and over mileage fees are consistent across the same class of vehicle regardless of location.

- Discount coupons for individual customers are unique and cannot be reused.

- For the payment methods, the card number is stored as a string to accommodate different formats.

- The odometer reading is always captured in miles.

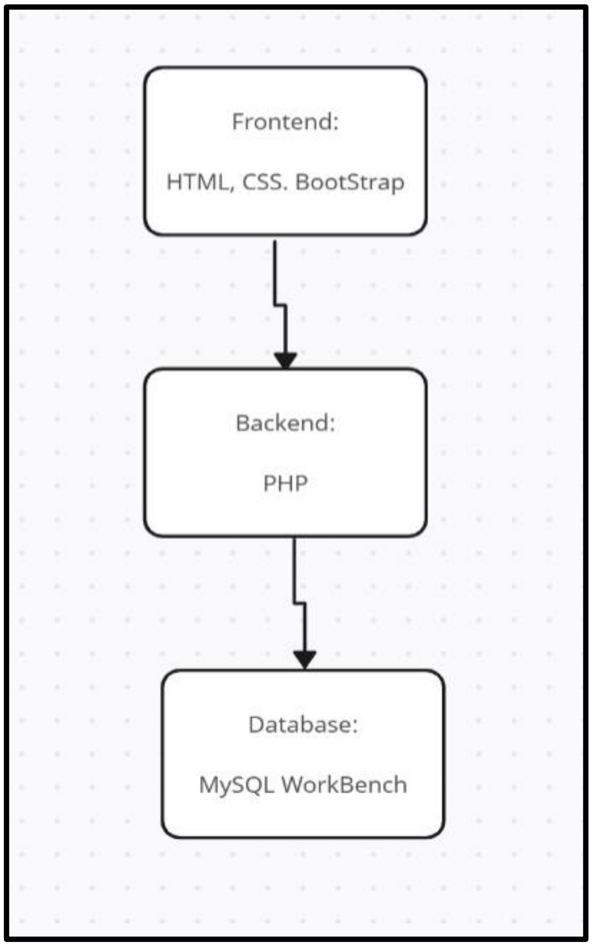
- A vehicle can only be rented out to one customer at a time.

- A vehicle once rented cannot be rented again until returned.

- A customer can rent multiple vehicles but with separate rental service records.

- All payments are processed at the time of vehicle return and there is no partial payment or deposit handling specified

Brief details of technologies used:



HTML and CSS are essential tools in web development. HTML is used for structuring web page content and defining elements like headings, paragraphs, and images. It provides the foundation for organizing information. CSS, on the other hand, controls the visual appearance of a webpage, including layout, colors, and fonts.

PHP is a popular server-side scripting language integral to backend web development, renowned for its ease of use and flexibility. Its straightforward syntax and ability to embed directly into HTML make it ideal for creating dynamic, data-driven web applications. PHP is compatible with various database systems and offers extensive support through its vast community, frameworks, and libraries. While historically critiqued for security and performance issues, ongoing improvements and updates, such as Just-In-Time compilation in PHP 8, have significantly enhanced its efficiency and robustness. PHP's widespread adoption is also bolstered by its extensive hosting support, making it a versatile and enduring choice in web development.

MySQL is a popular open-source relational database management system that offers excellent performance and scalability, allowing for efficient handling of large datasets and high-volume transactions. MySQL also boasts a robust feature set, including comprehensive support for SQL queries, ACID (Atomicity, Consistency, Isolation, Durability) compliance, and various indexing techniques for optimizing query execution. MySQL offers strong data security measures, including user authentication, access control, and encryption options. Moreover, it integrates seamlessly with various programming languages and frameworks, making it a versatile choice for developers across different platforms. Overall, MySQL's combination of performance, features, security, and community support establishes it as a reliable and widely adopted database solution.

Additional Features

- CRUD (Create, Read, Update, and Delete) Functionalities for Employees
- Service Management (Separate login for Employees to manage the services offered)
- SQL Injection Protection (Procedures)
- Triggers
- Password Encryption

DDL Code (Oracle)

```
-- Generated by Oracle SQL Developer Data Modeler 23.1.0.087.0806
--   at:        2023-11-12 00:31:30 EST
--   site:      Oracle Database 21c
--   type:      Oracle Database 21c



-- predefined type, no DDL - MDSYS.SDO_GEOMETRY

-- predefined type, no DDL - XMLTYPE

CREATE TABLE adg_card_details (
    cardid     INTEGER NOT NULL,
    cardtype   VARCHAR2(10 CHAR) NOT NULL,
    cardnumber INTEGER NOT NULL,
    card_fname VARCHAR2(10 CHAR) NOT NULL,
    card_lname VARCHAR2(10 CHAR) NOT NULL,
    cardexpiry DATE NOT NULL,
    card_cvv   INTEGER NOT NULL
);

COMMENT ON COLUMN adg_card_details.cardid IS
    'Card ID';
```

```
COMMENT ON COLUMN adg_card_details.cardtype IS

    'Card Type';

COMMENT ON COLUMN adg_card_details.cardnumber IS
    'Card Number';

COMMENT ON COLUMN adg_card_details.card_fname IS
    'Card First Name';

COMMENT ON COLUMN adg_card_details.card_lname IS
    'Card Last Name';

COMMENT ON COLUMN adg_card_details.cardexpiry IS
    'Card Expiration Date';

COMMENT ON COLUMN adg_card_details.card_cvv IS
    'Card CVV';

ALTER TABLE adg_card_details ADD CONSTRAINT adg_card_details_pk PRIMARY KEY (
cardid );

CREATE TABLE adg_corporate (
    custid              INTEGER NOT NULL,
    corporation_name    VARCHAR2(20 CHAR) NOT NULL,
    registration_number VARCHAR2(20 CHAR) NOT NULL,
    corporate_discount  VARCHAR2(10 CHAR)
);

COMMENT ON COLUMN adg_corporate.custid IS
    'Customer ID';

COMMENT ON COLUMN adg_corporate.corporation_name IS
    'Corporation_Name';

COMMENT ON COLUMN adg_corporate.registration_number IS
    'Registration_Number';

COMMENT ON COLUMN adg_corporate.corporate_discount IS
    'Corporate_Discount';
```

```
ALTER TABLE adg_corporate ADD CONSTRAINT adg_corporate_pk PRIMARY KEY (
custid );

CREATE TABLE adg_customer (
    custid      INTEGER NOT NULL,
    emailid     VARCHAR2(30 CHAR) NOT NULL,
    phonenumber INTEGER NOT NULL,
    cust_fname  VARCHAR2(20 CHAR) NOT NULL,
    cust_lname  VARCHAR2(30 CHAR) NOT NULL,
    custcity    VARCHAR2(20 CHAR) NOT NULL,
    custstate   VARCHAR2(20 CHAR) NOT NULL,
    custst      VARCHAR2(20 CHAR) NOT NULL,
    custzipcode INTEGER NOT NULL,
    custcountry VARCHAR2(20 CHAR) NOT NULL,
    cust_type   VARCHAR2(1) NOT NULL
);

ALTER TABLE adg_customer
    ADD CONSTRAINT ch_inh_adg_customer CHECK ( cust_type IN ( 'C', 'I' ) );

COMMENT ON COLUMN adg_customer.custid IS
    'Customer ID';

COMMENT ON COLUMN adg_customer.emailid IS
    'Email ID';

COMMENT ON COLUMN adg_customer.phonenumber IS
    'Phone Number';

COMMENT ON COLUMN adg_customer.cust_fname IS
    'First Name of Customer';

COMMENT ON COLUMN adg_customer.cust_lname IS
    'Last Name of Customer';

COMMENT ON COLUMN adg_customer.custcity IS
    'Customer City Address';

COMMENT ON COLUMN adg_customer.custstate IS
    'Customer State Address';
```

```
COMMENT ON COLUMN adg_customer.custst IS
    'Customer Street Address';

COMMENT ON COLUMN adg_customer.custzipcode IS
    'Cust Zipcode';

COMMENT ON COLUMN adg_customer.custcountry IS
    'Customer Country';

COMMENT ON COLUMN adg_customer.cust_type IS
    'Customer Type';

ALTER TABLE adg_customer ADD CONSTRAINT adg_customer_pk PRIMARY KEY (
custid );

CREATE TABLE adg_discountcoupon (
    couponid           INTEGER NOT NULL,
    discountpercentage INTEGER NOT NULL,
    valid_from         DATE NOT NULL,
    valid_to           DATE NOT NULL,
    custid             INTEGER
);

ALTER TABLE adg_discountcoupon
    ADD CONSTRAINT dsicount_percentage CHECK ( discountpercentage BETWEEN 0 AND
100 );

COMMENT ON COLUMN adg_discountcoupon.couponid IS
    'CouponID';

COMMENT ON COLUMN adg_discountcoupon.discountpercentage IS
    'Discount Percentage';

COMMENT ON COLUMN adg_discountcoupon.valid_from IS
    'Start Date of Discount';

COMMENT ON COLUMN adg_discountcoupon.valid_to IS
    'End Date of Discount';

ALTER TABLE adg_discountcoupon ADD CONSTRAINT adg_discountcoupon_pk
PRIMARY KEY ( couponid );
```

```
CREATE TABLE adg_individual (
    custid                  INTEGER NOT NULL,
    fname                   VARCHAR2(20 CHAR) NOT NULL,
    lname                   VARCHAR2(20 CHAR) NOT NULL,
    driver_license_number   VARCHAR2(20 CHAR) NOT NULL,
    insurance_company_name  CHAR(20 CHAR) NOT NULL,
    insurance_policy_number VARCHAR2(20 CHAR) NOT NULL,
    individual_discount     INTEGER
);

COMMENT ON COLUMN adg_individual.custid IS
    'Customer ID';

COMMENT ON COLUMN adg_individual.fname IS
    'First Name ';

COMMENT ON COLUMN adg_individual.lname IS
    'Last Name of Individual';

COMMENT ON COLUMN adg_individual.driver_license_number IS
    'Driver_License_Number';

COMMENT ON COLUMN adg_individual.insurance_company_name IS
    'Insurance_Company_Name';

COMMENT ON COLUMN adg_individual.insurance_policy_number IS
    'Insurance_Policy_Number';

COMMENT ON COLUMN adg_individual.individual_discount IS
    'Individual_Discount';

ALTER TABLE adg_individual ADD CONSTRAINT adg_individual_pk PRIMARY KEY (
custid );

CREATE TABLE adg_invoice (
    invoiceid   INTEGER NOT NULL,
    invoicedate DATE NOT NULL,
    rentalid    INTEGER NOT NULL
);

COMMENT ON COLUMN adg_invoice.invoiceid IS
    'InvoiceID';
```

```
COMMENT ON COLUMN adg_invoice.invoicedate IS
    'Invoice Date';

ALTER TABLE adg_invoice ADD CONSTRAINT adg_invoice_pk PRIMARY KEY (
invoiceid );

CREATE TABLE adg_office (
    office_id      INTEGER NOT NULL,
    phonenumber    INTEGER NOT NULL,
    officest       VARCHAR2(20 CHAR) NOT NULL,
    officecity     VARCHAR2(20 CHAR) NOT NULL,
    officestate    VARCHAR2(20 CHAR) NOT NULL,
    officecountry VARCHAR2(20 CHAR) NOT NULL,
    officezipcode VARCHAR2(20 CHAR) NOT NULL,
    emailid        VARCHAR2(30 CHAR)
);

COMMENT ON COLUMN adg_office.office_id IS
    'Office ID';

COMMENT ON COLUMN adg_office.phonenumber IS
    'Phone Number of Office';

COMMENT ON COLUMN adg_office.officest IS
    'Office Street Address';

COMMENT ON COLUMN adg_office.officecity IS
    'Office City Address';

COMMENT ON COLUMN adg_office.officestate IS
    'Office State Address';

COMMENT ON COLUMN adg_office.officecountry IS
    'Office Country Address';

COMMENT ON COLUMN adg_office.officezipcode IS
    'Office Zipcode';

COMMENT ON COLUMN adg_office.emailid IS
    'Email Address of Office';
```

```
ALTER TABLE adg_office ADD CONSTRAINT adg_office_pk PRIMARY KEY ( office_id );

CREATE TABLE adg_payment (
    paymentid     INTEGER NOT NULL,
    paymentdate   DATE NOT NULL,
    paymentmethod VARCHAR2(10) NOT NULL,
    invoiceid     INTEGER NOT NULL,
    cardid        INTEGER NOT NULL
);

ALTER TABLE adg_payment
    ADD CONSTRAINT payment CHECK ( paymentmethod IN ( 'credit', 'debit', 'gift' ) );

COMMENT ON COLUMN adg_payment.paymentid IS
    'Payment ID';

COMMENT ON COLUMN adg_payment.paymentdate IS
    'Payment Date';

COMMENT ON COLUMN adg_payment.paymentmethod IS
    'Payment Method';

ALTER TABLE adg_payment ADD CONSTRAINT adg_payment_pk PRIMARY KEY (
paymentid );

CREATE TABLE adg_rentalservice (
    rentalid             INTEGER NOT NULL,
    pickup_date          DATE NOT NULL,
    dropoff_date         DATE NOT NULL,
    start_odometer       INTEGER NOT NULL,
    end_odometer         INTEGER,
    daily_odometer_limit INTEGER,
    pickup_location      VARCHAR2(20 CHAR) NOT NULL,
    dropoff_location     VARCHAR2(20 CHAR) NOT NULL,
    custid               INTEGER NOT NULL,
    vin                  VARCHAR2(10 CHAR) NOT NULL
);

COMMENT ON COLUMN adg_rentalservice.rentalid IS
    'Rental ID';
```

```
COMMENT ON COLUMN adg_rentalservice.pickup_date IS
    'Pickup Date';

COMMENT ON COLUMN adg_rentalservice.dropoff_date IS
    'Dropoff_Date';

COMMENT ON COLUMN adg_rentalservice.start_odometer IS
    'Start_Odometer';

COMMENT ON COLUMN adg_rentalservice.end_odometer IS
    'End Odometer';

COMMENT ON COLUMN adg_rentalservice.daily_odometer_limit IS
    'Daily_Odometer_Limit';

COMMENT ON COLUMN adg_rentalservice.pickup_location IS
    ' Pickup Location';

COMMENT ON COLUMN adg_rentalservice.dropoff_location IS
    'Dropoff Location';

ALTER TABLE adg_rentalservice ADD CONSTRAINT adg_rentalservice_pk PRIMARY KEY
( rentalid );

CREATE TABLE adg_vehicle (
    vin          VARCHAR2(10 CHAR) NOT NULL,
    vehiclemake  VARCHAR2(10 CHAR) NOT NULL,
    vehiclemodel VARCHAR2(10 CHAR) NOT NULL,
    vehicleyear  INTEGER NOT NULL,
    licenseplate VARCHAR2(10 CHAR) NOT NULL,
    office_id    INTEGER NOT NULL,
    statusid     INTEGER NOT NULL,
    classid      INTEGER NOT NULL
);

COMMENT ON COLUMN adg_vehicle.vin IS
    'Vehicle Identification Number';

COMMENT ON COLUMN adg_vehicle.vehiclemake IS
    'Vehcle Make';

COMMENT ON COLUMN adg_vehicle.vehiclemodel IS
```

```
    'Model of vehicle';

COMMENT ON COLUMN adg_vehicle.vehicleyear IS
    'Vehicle Manufacturing Year ';

COMMENT ON COLUMN adg_vehicle.licenseplate IS
    'License Plate of Vehicle';

ALTER TABLE adg_vehicle ADD CONSTRAINT adg_vehicle_pk PRIMARY KEY ( vin );

CREATE TABLE adg_vehicleclass (
    classname        VARCHAR2(30 CHAR) NOT NULL,
    classid          INTEGER NOT NULL,
    daily_rate       INTEGER NOT NULL,
    over_mileagefees INTEGER NOT NULL
);

ALTER TABLE adg_vehicleclass
    ADD CONSTRAINT class_name CHECK ( classname IN ( 'Mini Van', 'Premium SUV',
'SUV', 'Station Wagon', 'luxury car',
                                                     'mid-size car', 'small car' ) );

COMMENT ON COLUMN adg_vehicleclass.classname IS
    'Vehicle Class Name';

COMMENT ON COLUMN adg_vehicleclass.classid IS
    'Class ID';

COMMENT ON COLUMN adg_vehicleclass.daily_rate IS
    'Daily Rate of Car';

COMMENT ON COLUMN adg_vehicleclass.over_mileagefees IS
    'Over Mileage Fees';

ALTER TABLE adg_vehicleclass ADD CONSTRAINT adg_vehicleclass_pk PRIMARY KEY (
classid );

CREATE TABLE adg_vehiclestatus (
    statusid INTEGER NOT NULL,
    status   VARCHAR2(10 CHAR) NOT NULL
);
```

```
ALTER TABLE adg_vehiclestatus
   ADD CONSTRAINT "Status of Vehicle" CHECK ( status IN ( 'Available', 'Rented', 'Under
Maintenance' ) );

COMMENT ON COLUMN adg_vehiclestatus.statusid IS
   'Status ID of Vehicle';

COMMENT ON COLUMN adg_vehiclestatus.status IS
   'Status of vehicle';

ALTER TABLE adg_vehiclestatus ADD CONSTRAINT adg_vehiclestatus_pk PRIMARY
KEY ( statusid );

ALTER TABLE adg_payment
   ADD CONSTRAINT card_details_fk FOREIGN KEY ( cardid )
      REFERENCES adg_card_details ( cardid );

ALTER TABLE adg_discountcoupon
   ADD CONSTRAINT customer_fk FOREIGN KEY ( custid )
      REFERENCES adg_customer ( custid );

ALTER TABLE adg_individual
   ADD CONSTRAINT customer_fkv1 FOREIGN KEY ( custid )
      REFERENCES adg_customer ( custid );

ALTER TABLE adg_rentalservice
   ADD CONSTRAINT customer_fkv2 FOREIGN KEY ( custid )
      REFERENCES adg_customer ( custid );

ALTER TABLE adg_corporate
   ADD CONSTRAINT customer_fkv3 FOREIGN KEY ( custid )
      REFERENCES adg_customer ( custid );

ALTER TABLE adg_payment
   ADD CONSTRAINT invoice_fk FOREIGN KEY ( invoiceid )
      REFERENCES adg_invoice ( invoiceid );

ALTER TABLE adg_vehicle
   ADD CONSTRAINT office_fk FOREIGN KEY ( office_id )
      REFERENCES adg_office ( office_id );

ALTER TABLE adg_invoice
```

```
   ADD CONSTRAINT rentalservice_fk FOREIGN KEY ( rentalid )
      REFERENCES adg_rentalservice ( rentalid );

ALTER TABLE adg_rentalservice
   ADD CONSTRAINT vehicle_fk FOREIGN KEY ( vin )
      REFERENCES adg_vehicle ( vin );

ALTER TABLE adg_vehicle
   ADD CONSTRAINT vehicleclass_fk FOREIGN KEY ( classid )
      REFERENCES adg_vehicleclass ( classid );

ALTER TABLE adg_vehicle
   ADD CONSTRAINT vehiclestatus_fk FOREIGN KEY ( statusid )
      REFERENCES adg_vehiclestatus ( statusid );

CREATE OR REPLACE TRIGGER arc_fkarc_4_adg_individual BEFORE
   INSERT OR UPDATE OF custid ON adg_individual
   FOR EACH ROW
DECLARE
   d VARCHAR2(1);
BEGIN
   SELECT
      a.cust_type
   INTO d
   FROM
      adg_customer a
   WHERE
      a.custid = :new.custid;

   IF ( d IS NULL OR d <> 'I' ) THEN
      raise_application_error(-20223, 'FK Customer_FKv1 in Table ADG_Individual violates
Arc constraint on Table ADG_Customer - discriminator column Cust_Type doesn''t have value
''I'''
      );
   END IF;

EXCEPTION
   WHEN no_data_found THEN
      NULL;
   WHEN OTHERS THEN
      RAISE;
END;
```

```
/

CREATE OR REPLACE TRIGGER arc_fkarc_4_adg_corporate BEFORE
   INSERT OR UPDATE OF custid ON adg_corporate
   FOR EACH ROW
DECLARE
   d VARCHAR2(1);
BEGIN
   SELECT
      a.cust_type
   INTO d
   FROM
      adg_customer a
   WHERE
      a.custid = :new.custid;

   IF ( d IS NULL OR d <> 'C' ) THEN
      raise_application_error(-20223, 'FK Customer_FKv3 in Table ADG_Corporate violates Arc constraint on Table ADG_Customer - discriminator column Cust_Type doesn"t have value "C"'
      );
   END IF;

EXCEPTION
   WHEN no_data_found THEN
      NULL;
   WHEN OTHERS THEN
      RAISE;
END;
/



-- Oracle SQL Developer Data Modeler Summary Report:
--
-- CREATE TABLE                            12
-- CREATE INDEX                             0
-- ALTER TABLE                             28
-- CREATE VIEW                              0
-- ALTER VIEW                               0
-- CREATE PACKAGE                           0
-- CREATE PACKAGE BODY                      0
-- CREATE PROCEDURE                         0
-- CREATE FUNCTION            0
-- CREATE TRIGGER            2
-- ALTER TRIGGER            0
-- CREATE COLLECTION TYPE        0
-- CREATE STRUCTURED TYPE        0
-- CREATE STRUCTURED TYPE BODY      0
-- CREATE CLUSTER            0
-- CREATE CONTEXT            0
-- CREATE DATABASE            0
-- CREATE DIMENSION           0
-- CREATE DIRECTORY           0
-- CREATE DISK GROUP          0
-- CREATE ROLE             0
-- CREATE ROLLBACK SEGMENT         0
-- CREATE SEQUENCE            0
-- CREATE MATERIALIZED VIEW        0
-- CREATE MATERIALIZED VIEW LOG      0
-- CREATE SYNONYM             0
-- CREATE TABLESPACE           0
-- CREATE USER             0
--
-- DROP TABLESPACE            0
-- DROP DATABASE             0
--
-- REDACTION POLICY           0
--
-- ORDS DROP SCHEMA            0
-- ORDS ENABLE SCHEMA           0
-- ORDS ENABLE OBJECT          0
--
-- ERRORS               0
-- WARNINGS               0
```

List of tables, and total number of records of each table

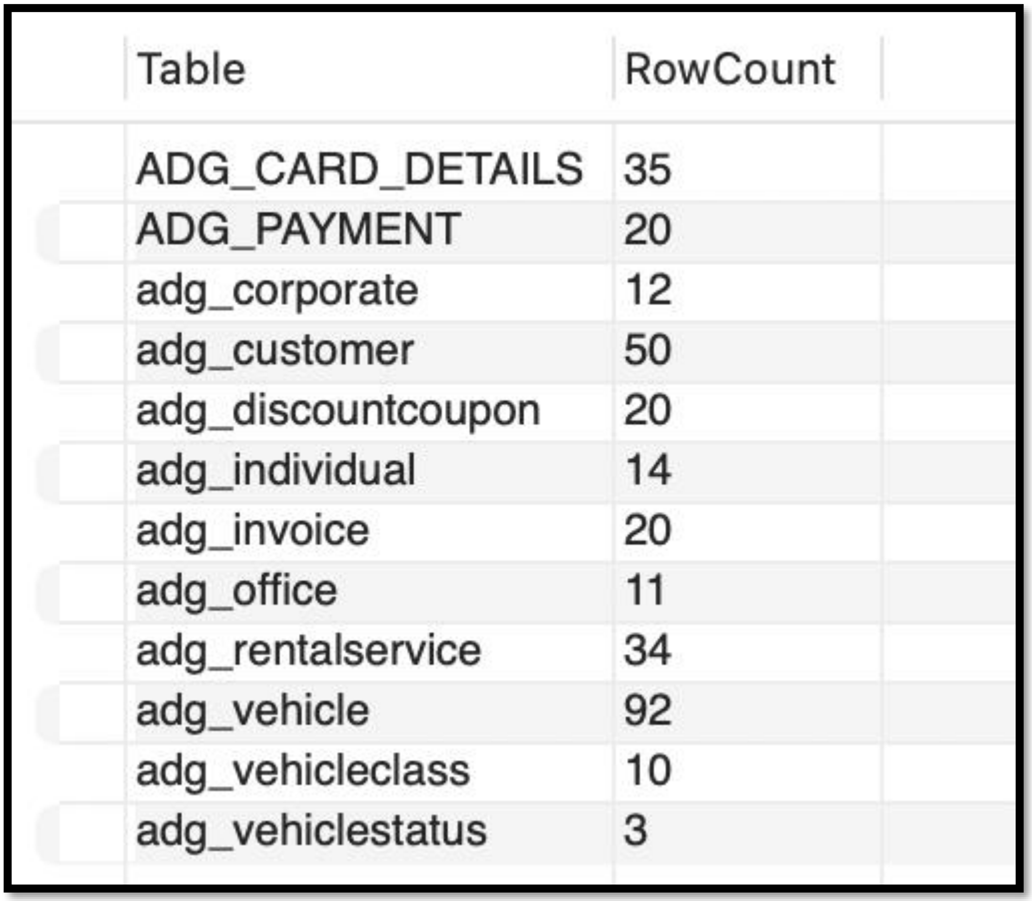
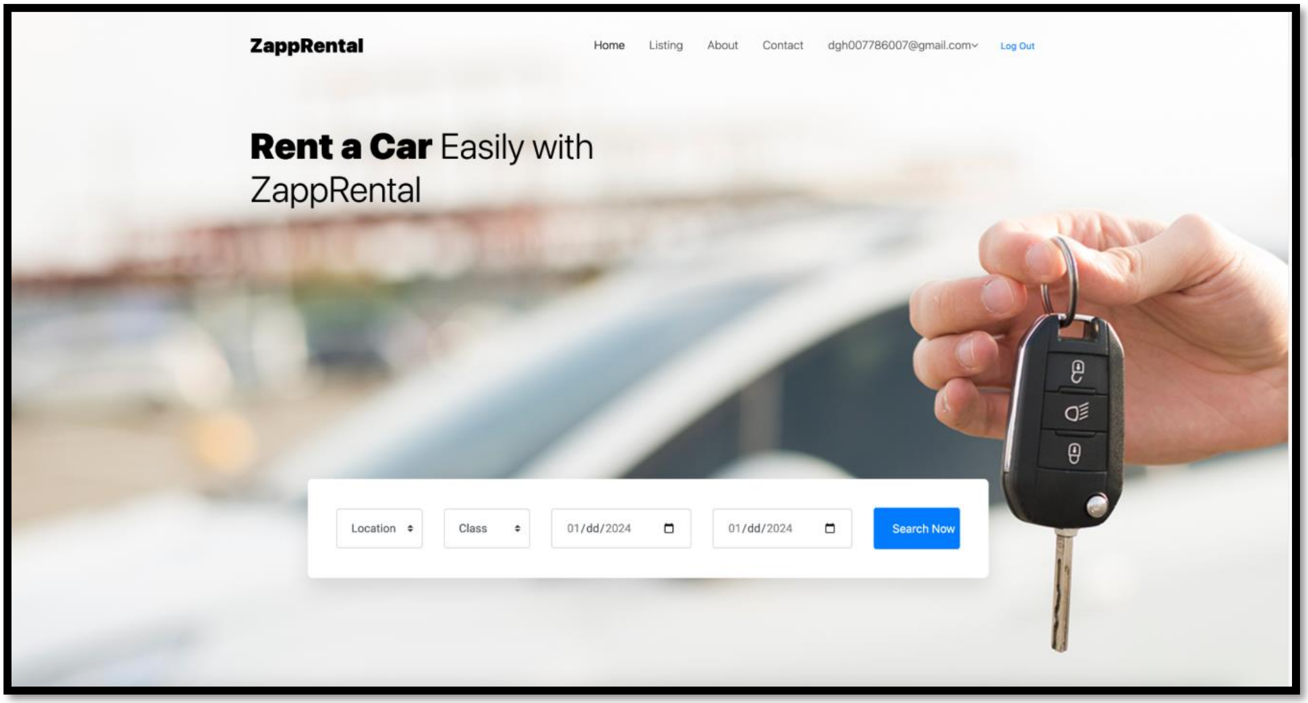


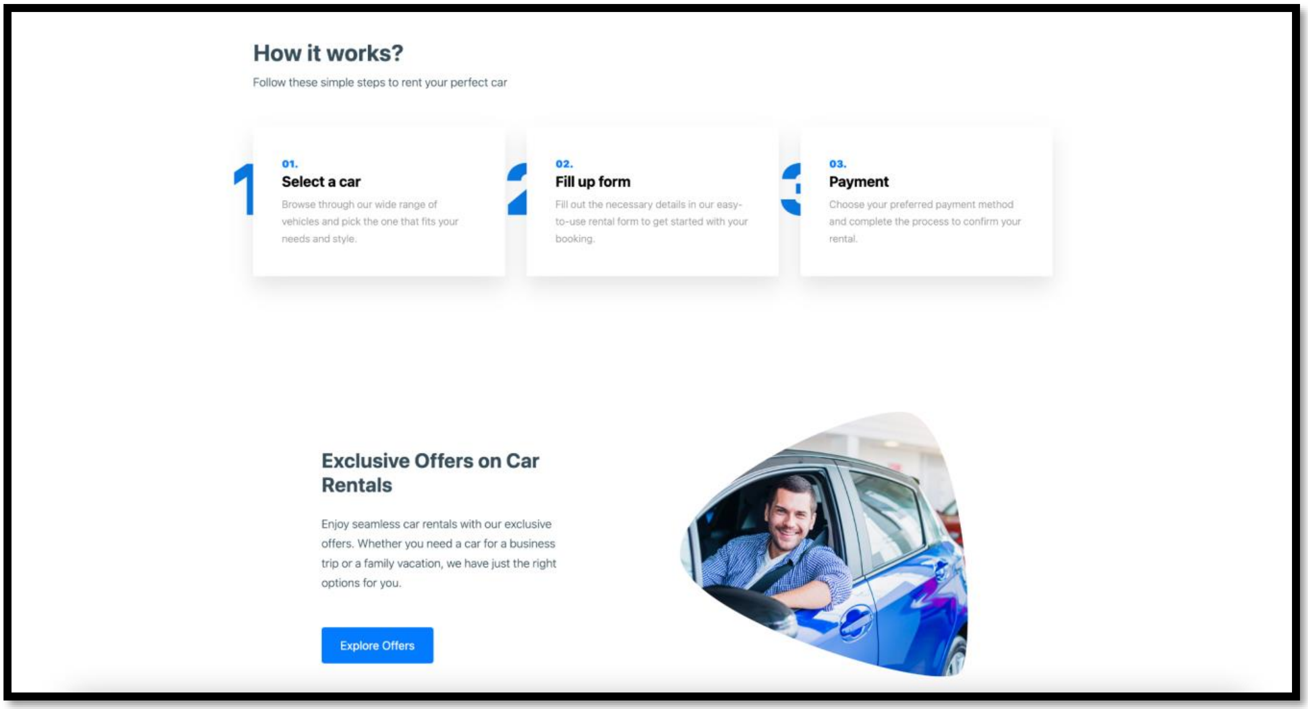
Table	RowCount
ADG_CARD_DETAILS	35
ADG_PAYMENT	20
adg_corporate	12
adg_customer	50
adg_discountcoupon	20
adg_individual	14
adg_invoice	20
adg_office	11
adg_rentalservice	34
adg_vehicle	92
adg_vehicleclass	10
adg_vehiclestatus	3

Screenshots of web application

Home page:



Rent a Car Easily with ZappRental



How it works?

Follow these simple steps to rent your perfect car

1 **01.**

Select a car

Browse through our wide range of vehicles and pick the one that fits your needs and style.

2 **02.**

Fill up form

Fill out the necessary details in our easy-to-use rental form to get started with your booking.

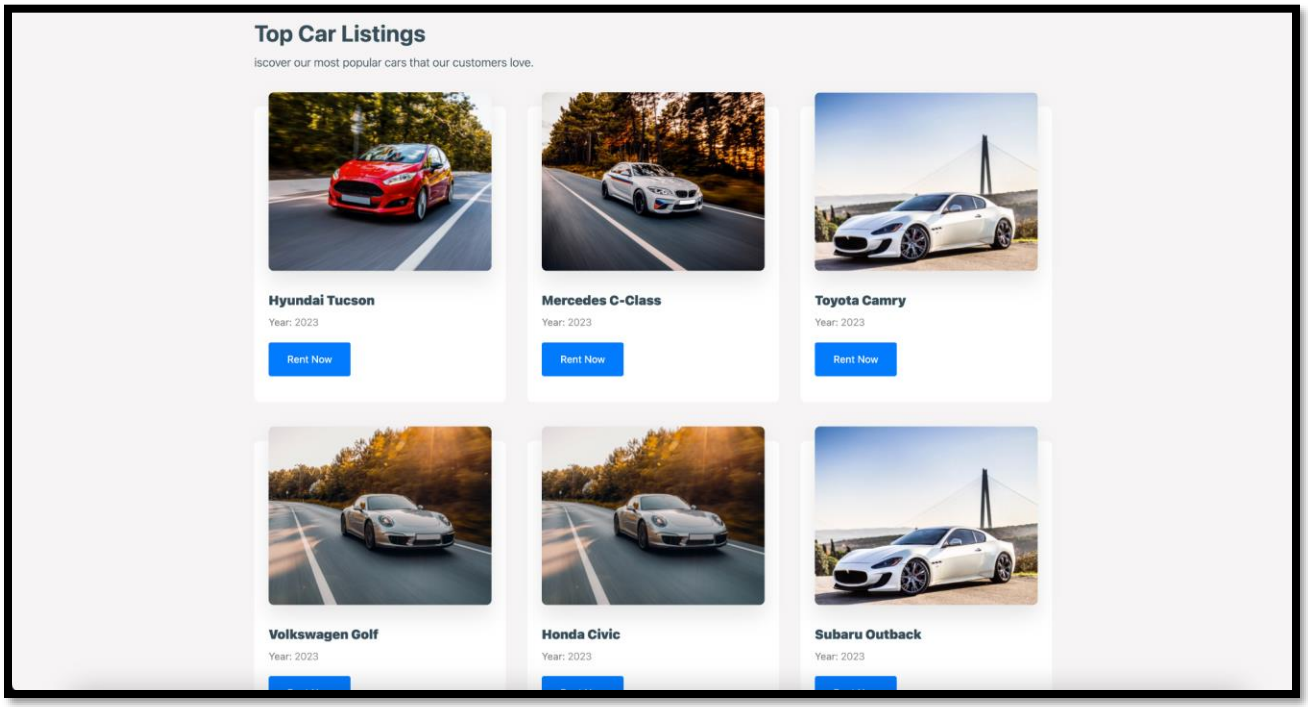
3 **03.**

Payment

Choose your preferred payment method and complete the process to confirm your rental.

Exclusive Offers on Car Rentals

Enjoy seamless car rentals with our exclusive offers. Whether you need a car for a business trip or a family vacation, we have just the right options for you.



Top Car Listings

iscover our most popular cars that our customers love.

Hyundai Tucson

Year: 2023

Mercedes C-Class

Year: 2023

Toyota Camry

Year: 2023

Volkswagen Golf

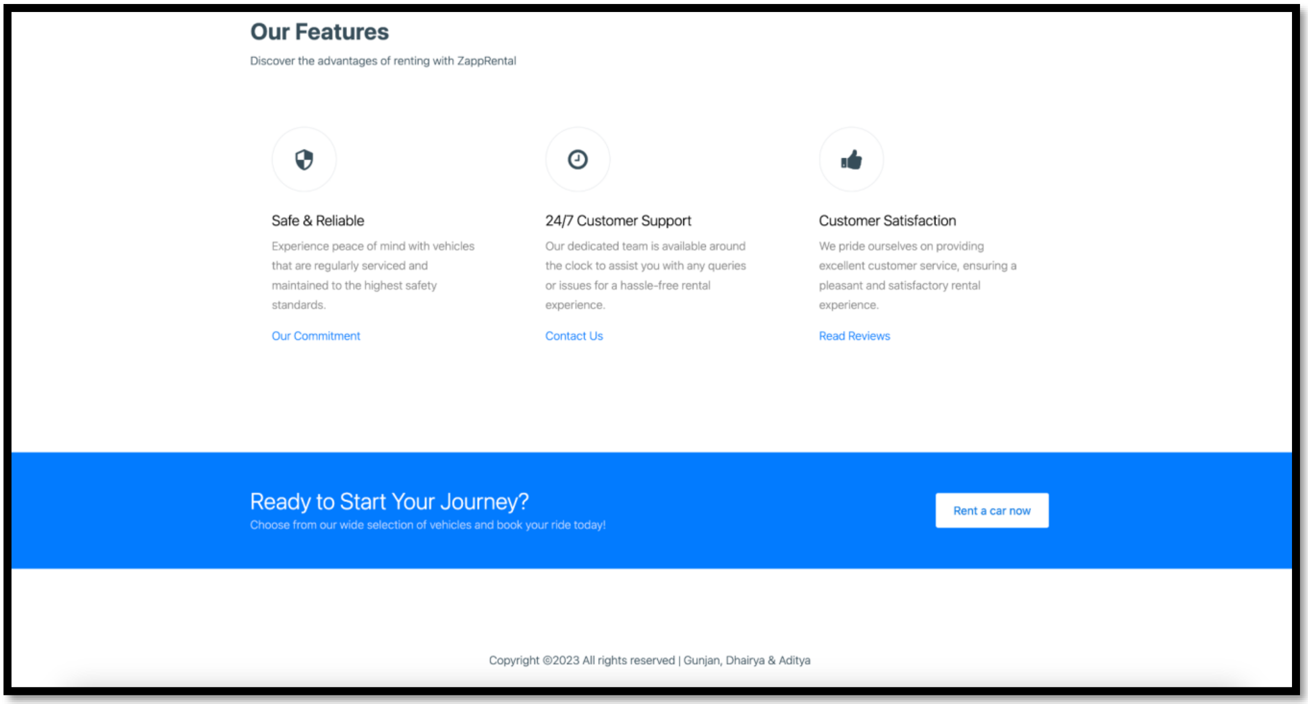
Year: 2023

Honda Civic

Year: 2023

Subaru Outback

Year: 2023



Our Features

Discover the advantages of renting with ZappRental

Safe & Reliable

Experience peace of mind with vehicles that are regularly serviced and maintained to the highest safety standards.

24/7 Customer Support

Our dedicated team is available around the clock to assist you with any queries or issues for a hassle-free rental experience.

Customer Satisfaction

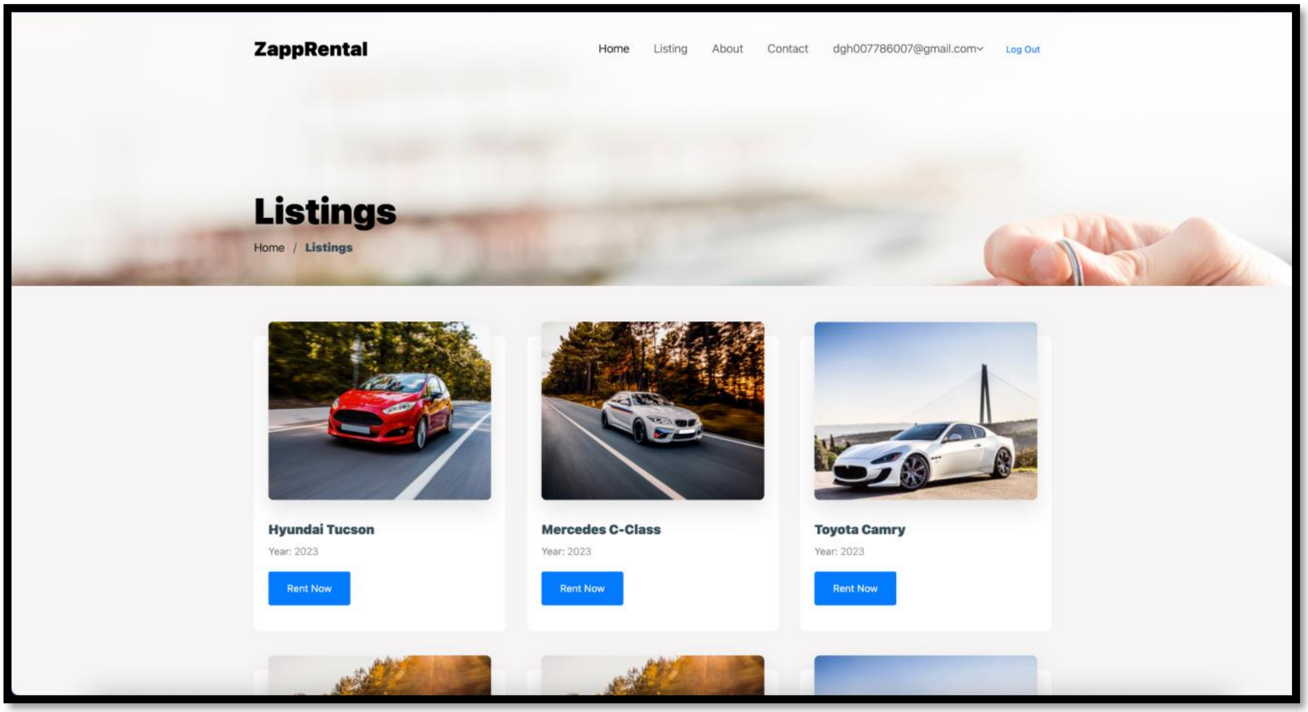
We pride ourselves on providing excellent customer service, ensuring a pleasant and satisfactory rental experience.

Ready to Start Your Journey?

Choose from our wide selection of vehicles and book your ride today!

Copyright ©2023 All rights reserved | Gunjan, Dhairya & Aditya

Listing page:

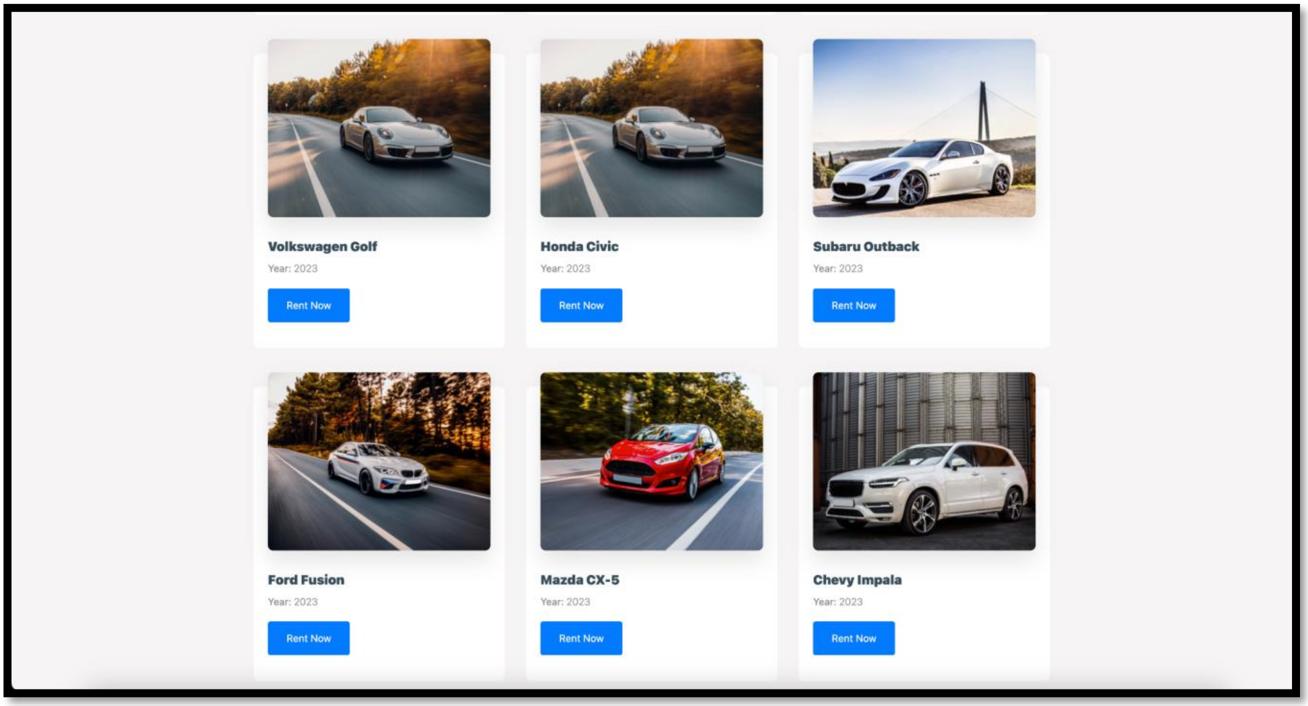


Listings

Hyundai Tucson
Year: 2023

Mercedes C-Class
Year: 2023

Toyota Camry
Year: 2023



Volkswagen Golf
Year: 2023

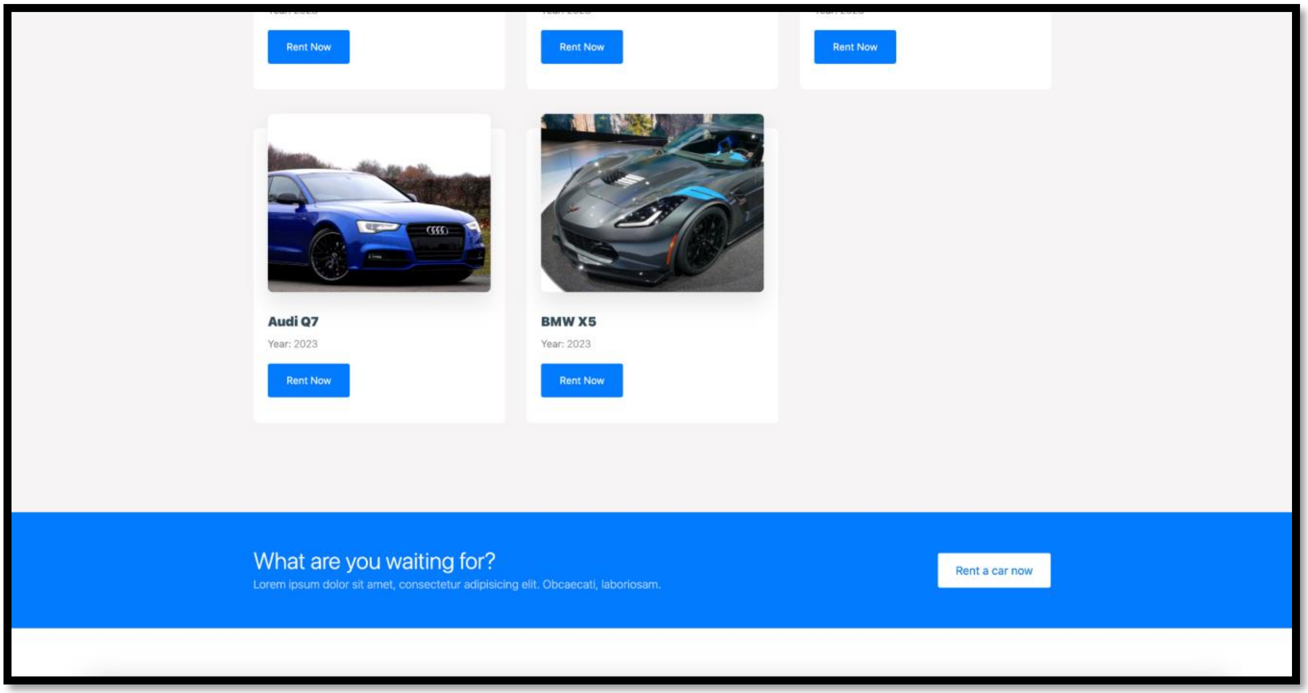
Honda Civic
Year: 2023

Subaru Outback
Year: 2023

Ford Fusion
Year: 2023

Mazda CX-5
Year: 2023

Chevy Impala
Year: 2023



Audi Q7

Year: 2023

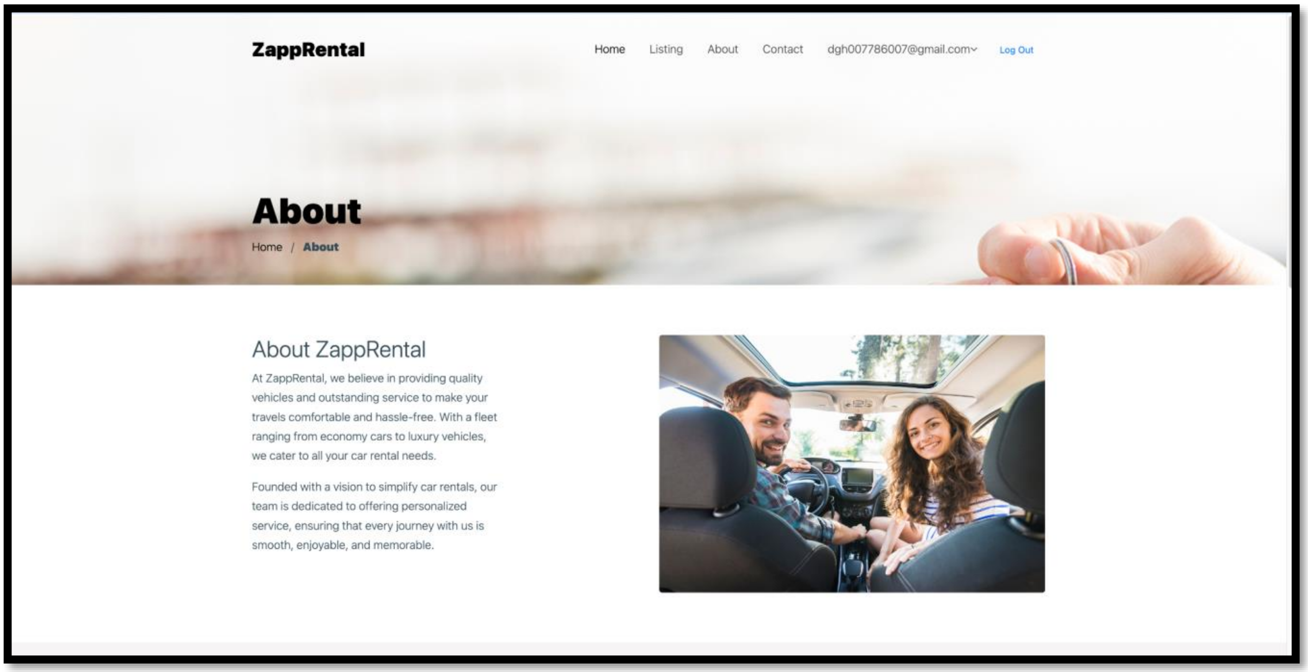
BMW X5

Year: 2023

What are you waiting for?

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Obcaecati, laboriosam.

About Page:



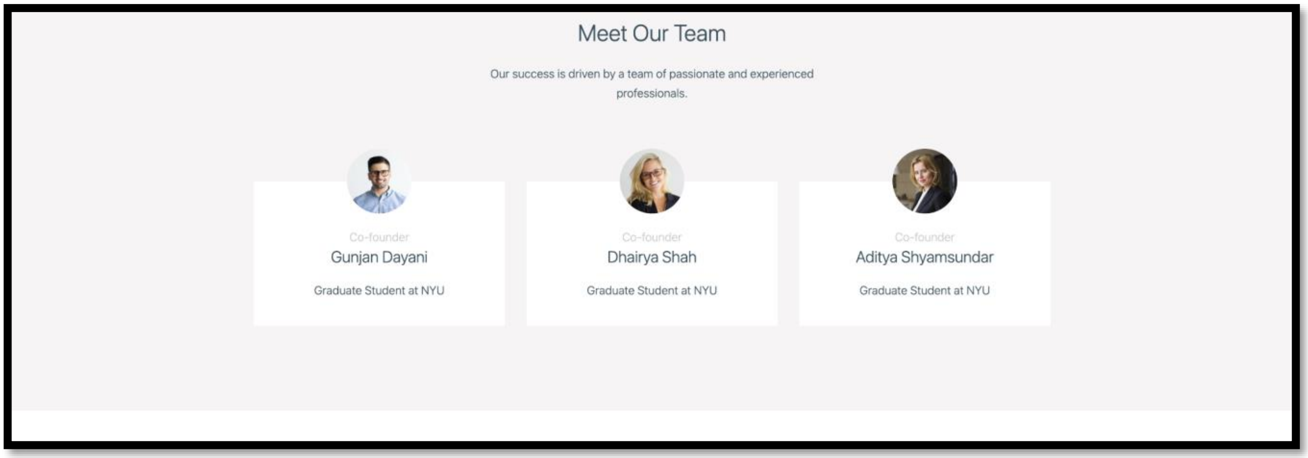
About

Home / **About**

About ZappRental

At ZappRental, we believe in providing quality vehicles and outstanding service to make your travels comfortable and hassle-free. With a fleet ranging from economy cars to luxury vehicles, we cater to all your car rental needs.

Founded with a vision to simplify car rentals, our team is dedicated to offering personalized service, ensuring that every journey with us is smooth, enjoyable, and memorable.



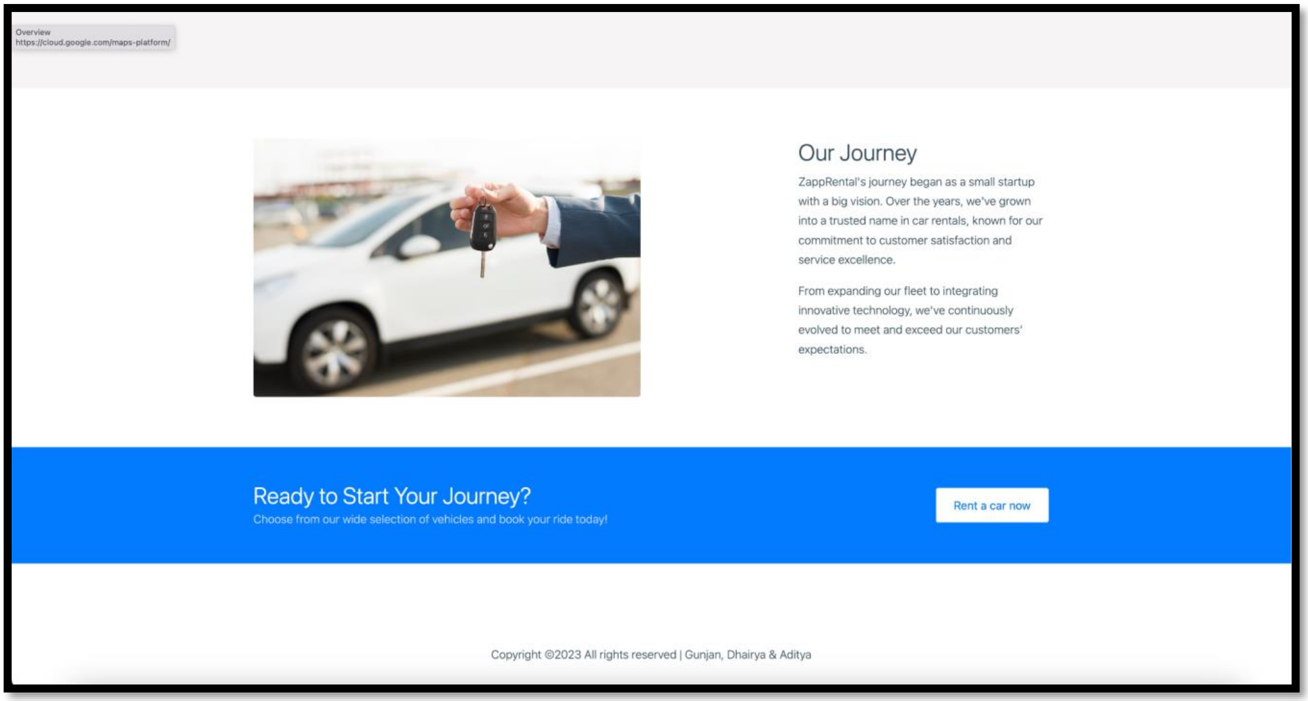
Meet Our Team

Our success is driven by a team of passionate and experienced professionals.

Co-founder
Gunjan Dayani
Graduate Student at NYU

Co-founder
Dhairya Shah
Graduate Student at NYU

Co-founder
Aditya Shyamsundar
Graduate Student at NYU



Overview
https://cloud.google.com/maps-platform/

Our Journey

ZappRental's journey began as a small startup with a big vision. Over the years, we've grown into a trusted name in car rentals, known for our commitment to customer satisfaction and service excellence.

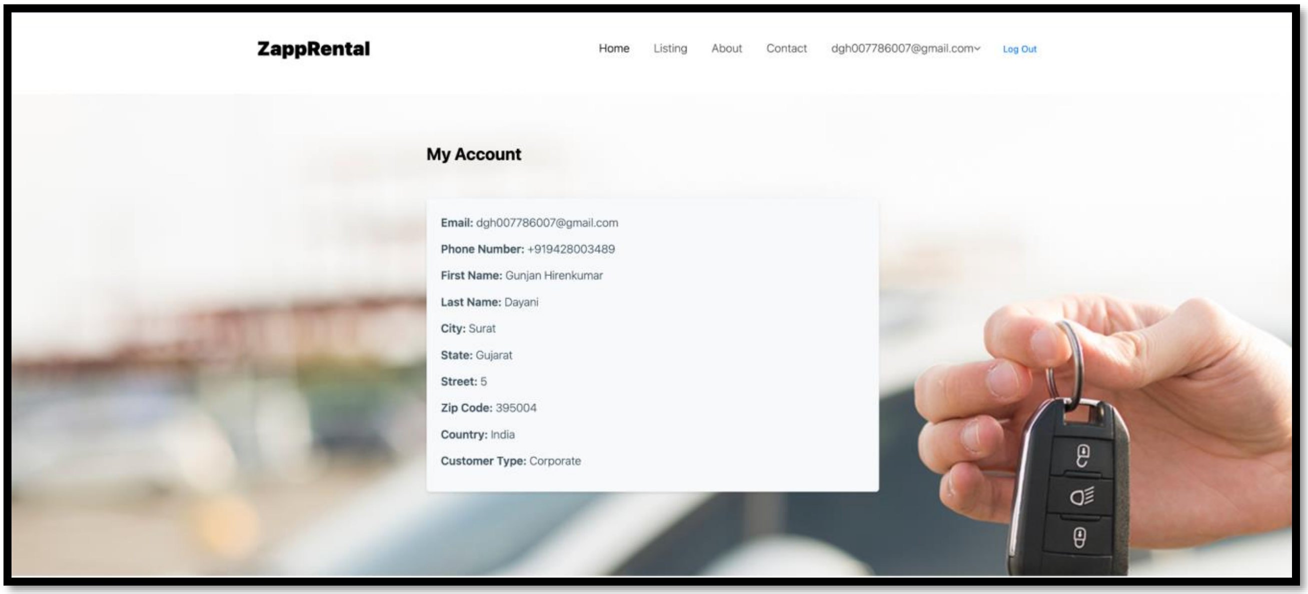
From expanding our fleet to integrating innovative technology, we've continuously evolved to meet and exceed our customers' expectations.

Ready to Start Your Journey?

Choose from our wide selection of vehicles and book your ride today!

Copyright ©2023 All rights reserved | Gunjan, Dhairya & Aditya

Account page:



My Account

Email: dgh007786007@gmail.com

Phone Number: +919428003489

First Name: Gunjan Hirenkumar

Last Name: Dayani

City: Surat

State: Gujarat

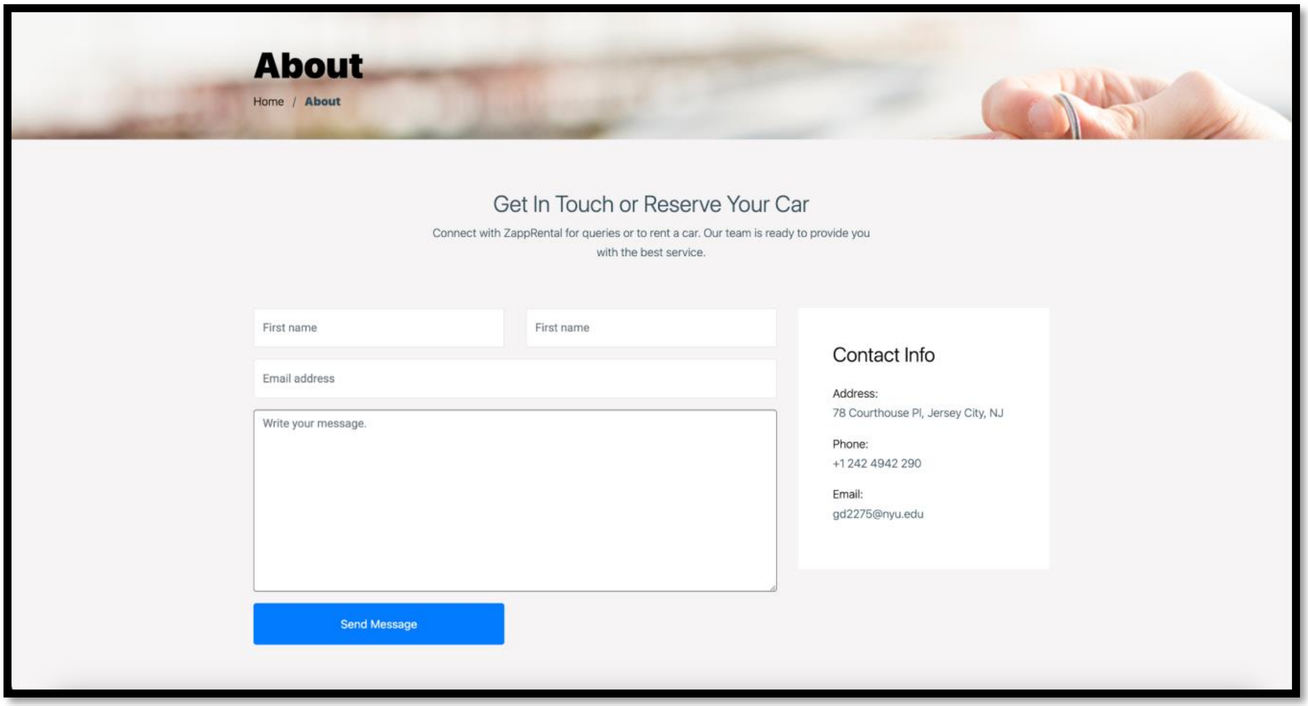
Street: 5

Zip Code: 395004

Country: India

Customer Type: Corporate

Contact Page:



Get In Touch or Reserve Your Car

Connect with ZappRental for queries or to rent a car. Our team is ready to provide you with the best service.

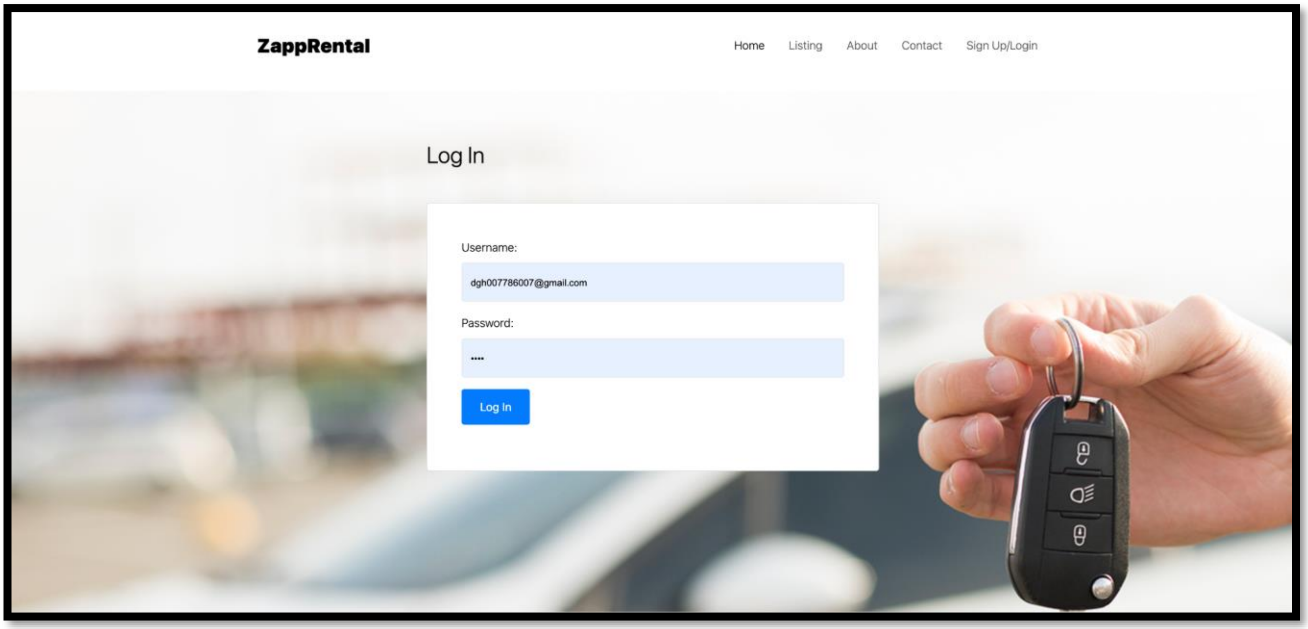
Contact Info

Address:
78 Courthouse Pl, Jersey City, NJ

Phone:
+1 242 4942 290

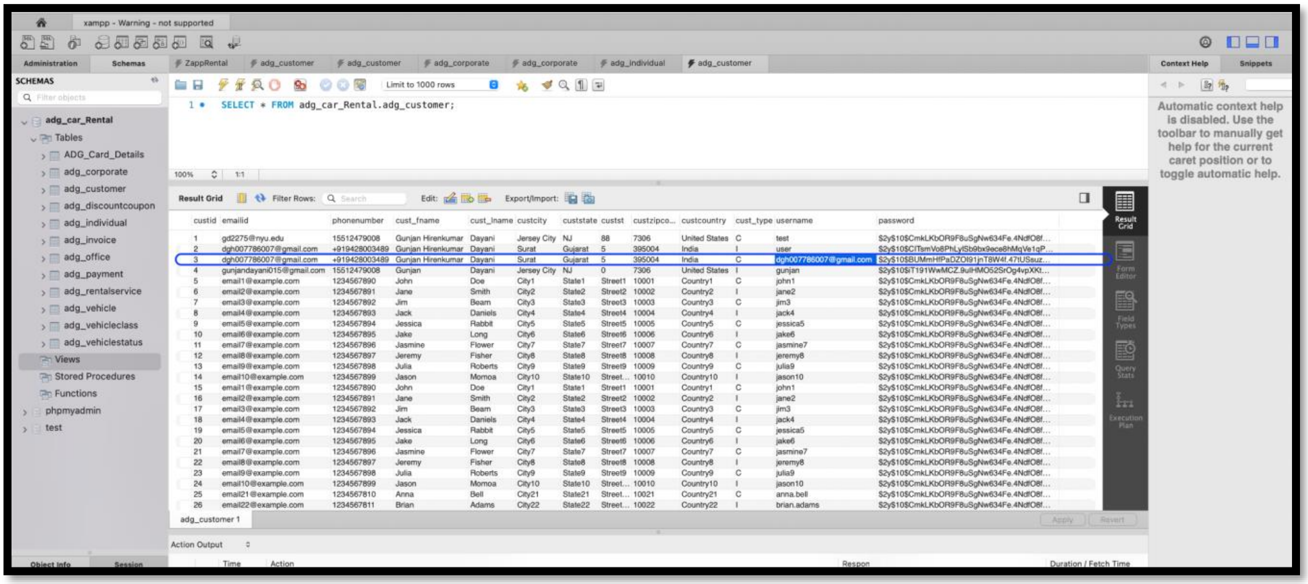
Email:
gd2275@nyu.edu

Login Page:



Username: dgh007786007@gmail.com

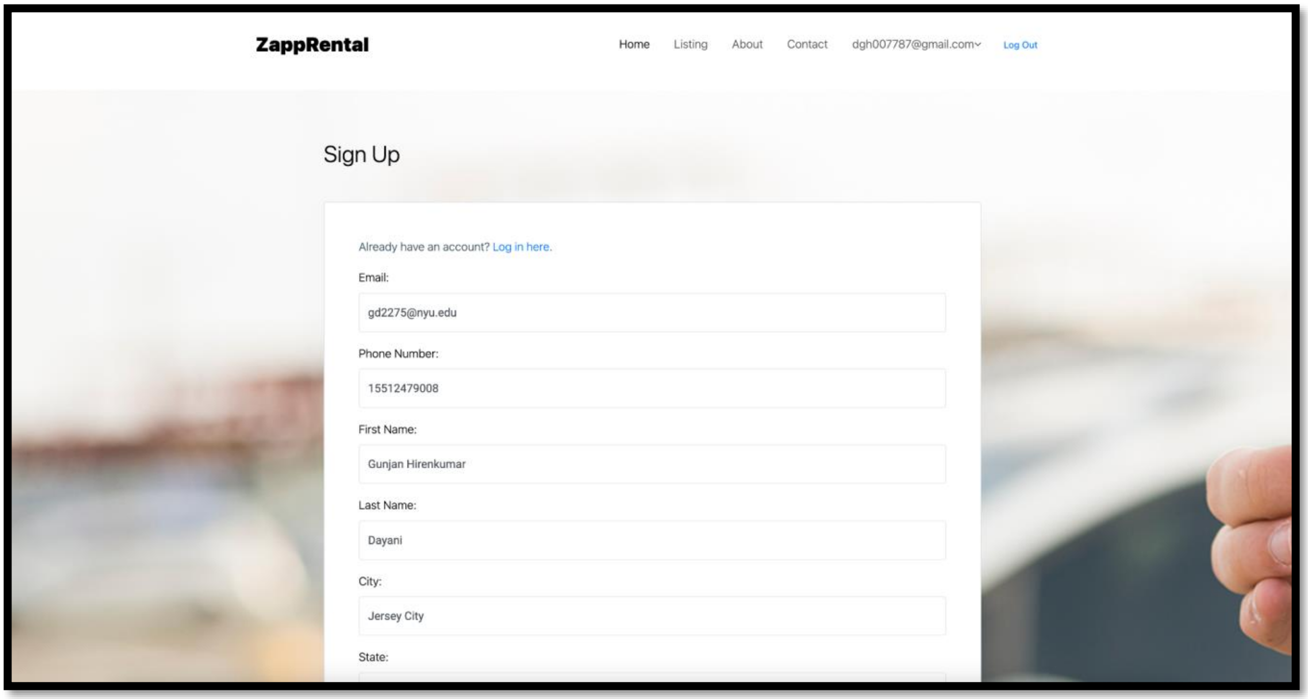
Password: ••••



```
SELECT * FROM adg_car_Rental.adg_customer;
```

custid	emailid	phonenumber	cust_fname	cust_lname	custcity	custstate	custst	custzipco...	custcountry	cust_type	username	password
1	gd2275@nyu.edu	15512479008	Gunjan Hirenkumar	Dayani	Jersey City	NJ	88	7306	United States	C	test	$2y$10$CmkLKbOR9F8uSgNw634Fe.4NdfO8f...
2	dgh007786007@gmail.com	+919428003489	Gunjan Hirenkumar	Dayani	Surat	Gujarat	5	395004	India	I	user	$2y$10$CITsmVo8PhLylSb9bx9ece8hMqVe1qP...
3	dgh007786007@gmail.com	+919428003489	Gunjan Hirenkumar	Dayani	Surat	Gujarat	5	395004	India	C	dgh007786007@gmail.com	$2y$10$BUMmHfPaDZOI91jnT8W4f.47tUSsuz...
4	gunjandayani015@gmail.com	15512479008	Gunjan	Dayani	Jersey City	NJ	0	7306	United States	I	gunjan	$2y$10$iT191WwMCZ.9ulHMO52SrOg4vpXKt...
5	email1@example.com	1234567890	John	Doe	City1	State1	Street1	10001	Country1	C	john1	$2y$10$CmkLKbOR9F8uSgNw634Fe.4NdfO8f...
6	email2@example.com	1234567891	Jane	Smith	City2	State2	Street2	10002	Country2	I	jane2	$2y$10$CmkLKbOR9F8uSgNw634Fe.4NdfO8f...
7	email3@example.com	1234567892	Jim	Beam	City3	State3	Street3	10003	Country3	C	jim3	$2y$10$CmkLKbOR9F8uSgNw634Fe.4NdfO8f...
8	email4@example.com	1234567893	Jack	Daniels	City4	State4	Street4	10004	Country4	I	jack4	$2y$10$CmkLKbOR9F8uSgNw634Fe.4NdfO8f...
9	email5@example.com	1234567894	Jessica	Rabbit	City5	State5	Street5	10005	Country5	C	jessica5	$2y$10$CmkLKbOR9F8uSgNw634Fe.4NdfO8f...
10	email6@example.com	1234567895	Jake	Long	City6	State6	Street6	10006	Country6	I	jake6	$2y$10$CmkLKbOR9F8uSgNw634Fe.4NdfO8f...
11	email7@example.com	1234567896	Jasmine	Flower	City7	State7	Street7	10007	Country7	C	jasmine7	$2y$10$CmkLKbOR9F8uSgNw634Fe.4NdfO8f...
12	email8@example.com	1234567897	Jeremy	Fisher	City8	State8	Street8	10008	Country8	I	jeremy8	$2y$10$CmkLKbOR9F8uSgNw634Fe.4NdfO8f...
13	email9@example.com	1234567898	Julia	Roberts	City9	State9	Street9	10009	Country9	C	julia9	$2y$10$CmkLKbOR9F8uSgNw634Fe.4NdfO8f...
14	email10@example.com	1234567899	Jason	Momoa	City10	State10	Street...	10010	Country10	I	jason10	$2y$10$CmkLKbOR9F8uSgNw634Fe.4NdfO8f...
15	email1@example.com	1234567890	John	Doe	City1	State1	Street1	10001	Country1	C	john1	$2y$10$CmkLKbOR9F8uSgNw634Fe.4NdfO8f...
16	email2@example.com	1234567891	Jane	Smith	City2	State2	Street2	10002	Country2	I	jane2	$2y$10$CmkLKbOR9F8uSgNw634Fe.4NdfO8f...
17	email3@example.com	1234567892	Jim	Beam	City3	State3	Street3	10003	Country3	C	jim3	$2y$10$CmkLKbOR9F8uSgNw634Fe.4NdfO8f...
18	email4@example.com	1234567893	Jack	Daniels	City4	State4	Street4	10004	Country4	I	jack4	$2y$10$CmkLKbOR9F8uSgNw634Fe.4NdfO8f...
19	email5@example.com	1234567894	Jessica	Rabbit	City5	State5	Street5	10005	Country5	C	jessica5	$2y$10$CmkLKbOR9F8uSgNw634Fe.4NdfO8f...
20	email6@example.com	1234567895	Jake	Long	City6	State6	Street6	10006	Country6	I	jake6	$2y$10$CmkLKbOR9F8uSgNw634Fe.4NdfO8f...
21	email7@example.com	1234567896	Jasmine	Flower	City7	State7	Street7	10007	Country7	C	jasmine7	$2y$10$CmkLKbOR9F8uSgNw634Fe.4NdfO8f...
22	email8@example.com	1234567897	Jeremy	Fisher	City8	State8	Street8	10008	Country8	I	jeremy8	$2y$10$CmkLKbOR9F8uSgNw634Fe.4NdfO8f...
23	email9@example.com	1234567898	Julia	Roberts	City9	State9	Street9	10009	Country9	C	julia9	$2y$10$CmkLKbOR9F8uSgNw634Fe.4NdfO8f...
24	email10@example.com	1234567899	Jason	Momoa	City10	State10	Street...	10010	Country10	I	jason10	$2y$10$CmkLKbOR9F8uSgNw634Fe.4NdfO8f...
25	email21@example.com	1234567810	Anna	Bell	City21	State21	Street...	10021	Country21	C	anna.bell	$2y$10$CmkLKbOR9F8uSgNw634Fe.4NdfO8f...
26	email22@example.com	1234567811	Brian	Adams	City22	State22	Street...	10022	Country22	I	brian.adams	$2y$10$CmkLKbOR9F8uSgNw634Fe.4NdfO8f...

Sign up:



Sign Up

Already have an account? Log in here.

Email: gd2275@nyu.edu

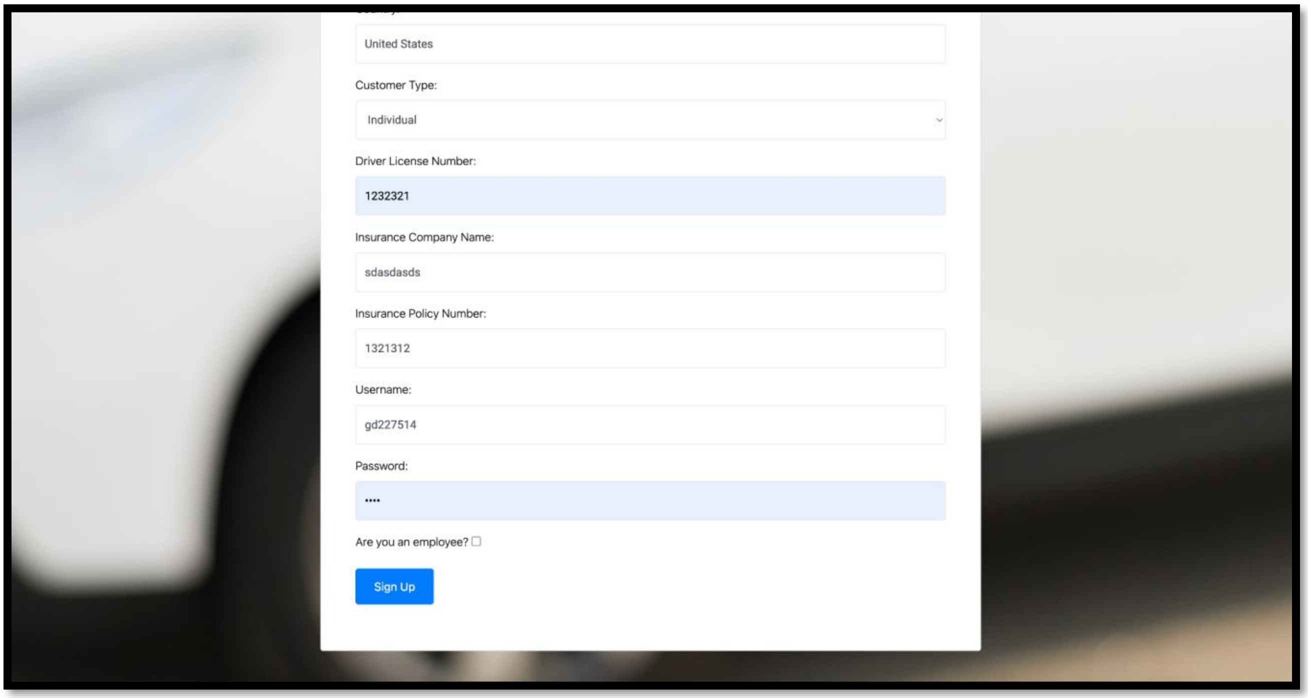
Phone Number: 15512479008

First Name: Gunjan Hirenkumar

Last Name: Dayani

City: Jersey City

State:



United States

Customer Type: Individual

Driver License Number: 1232321

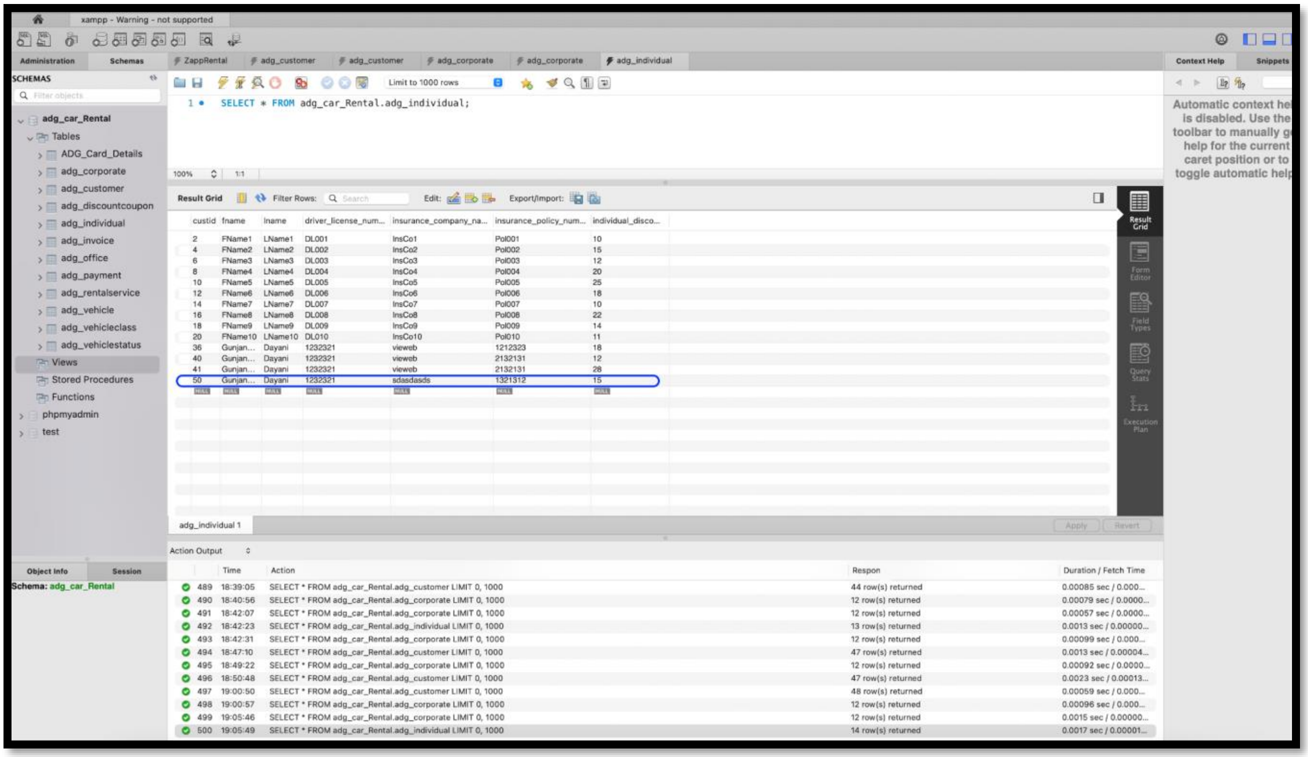
Insurance Company Name: sdasdasds

Insurance Policy Number: 1321312

Username: gd227514

Password: ••••

Are you an employee? □

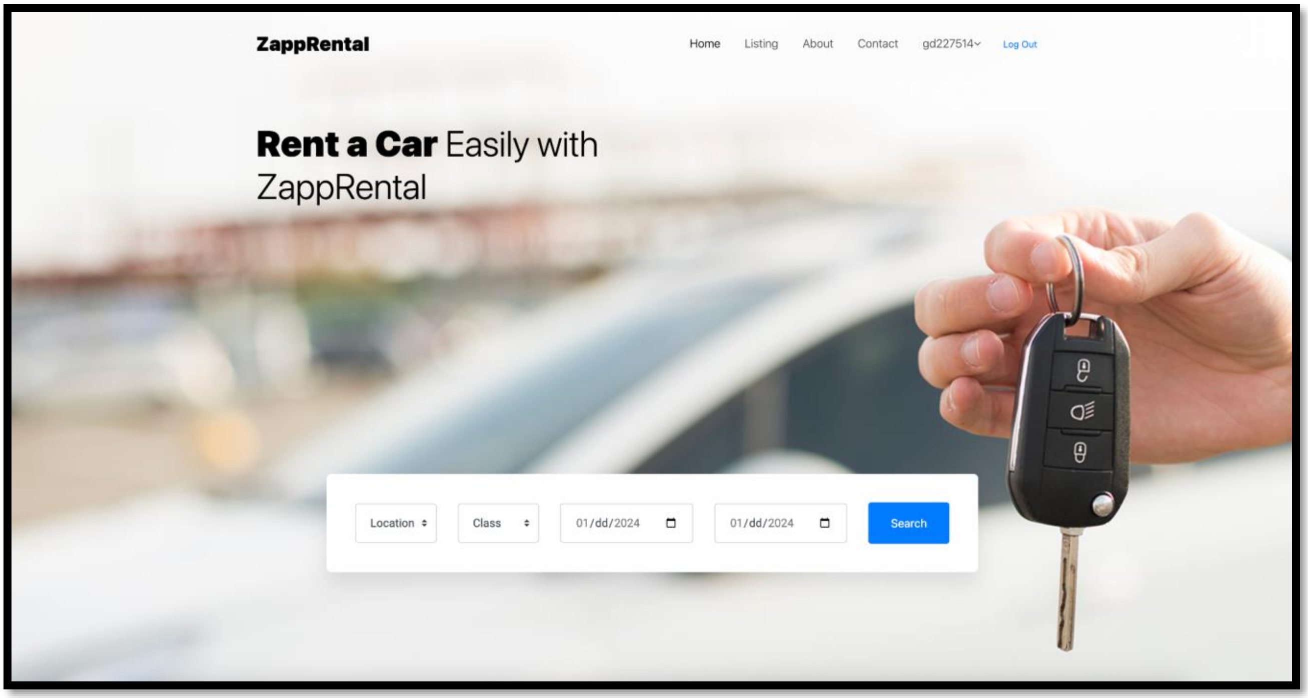


```
1 •  SELECT * FROM adg_car_Rental.adg_individual;
```

custid	fname	lname	driver_license_num...	insurance_company_na...	insurance_policy_num...	individual_disco...
2	FName1	LName1	DL001	InsCo1	Pol001	10
4	FName2	LName2	DL002	InsCo2	Pol002	15
6	FName3	LName3	DL003	InsCo3	Pol003	12
8	FName4	LName4	DL004	InsCo4	Pol004	20
10	FName5	LName5	DL005	InsCo5	Pol005	25
12	FName6	LName6	DL006	InsCo6	Pol006	18
14	FName7	LName7	DL007	InsCo7	Pol007	10
16	FName8	LName8	DL008	InsCo8	Pol008	22
18	FName9	LName9	DL009	InsCo9	Pol009	14
20	FName10	LName10	DL010	InsCo10	Pol010	11
36	Gunjan...	Dayani	1232321	vieweb	1212323	18
40	Gunjan...	Dayani	1232321	vieweb	2132131	12
41	Gunjan...	Dayani	1232321	vieweb	2132131	28
50	Gunjan...	Dayani	1232321	sdasdasds	1321312	15
NULL	NULL	NULL	NULL	NULL	NULL	NULL

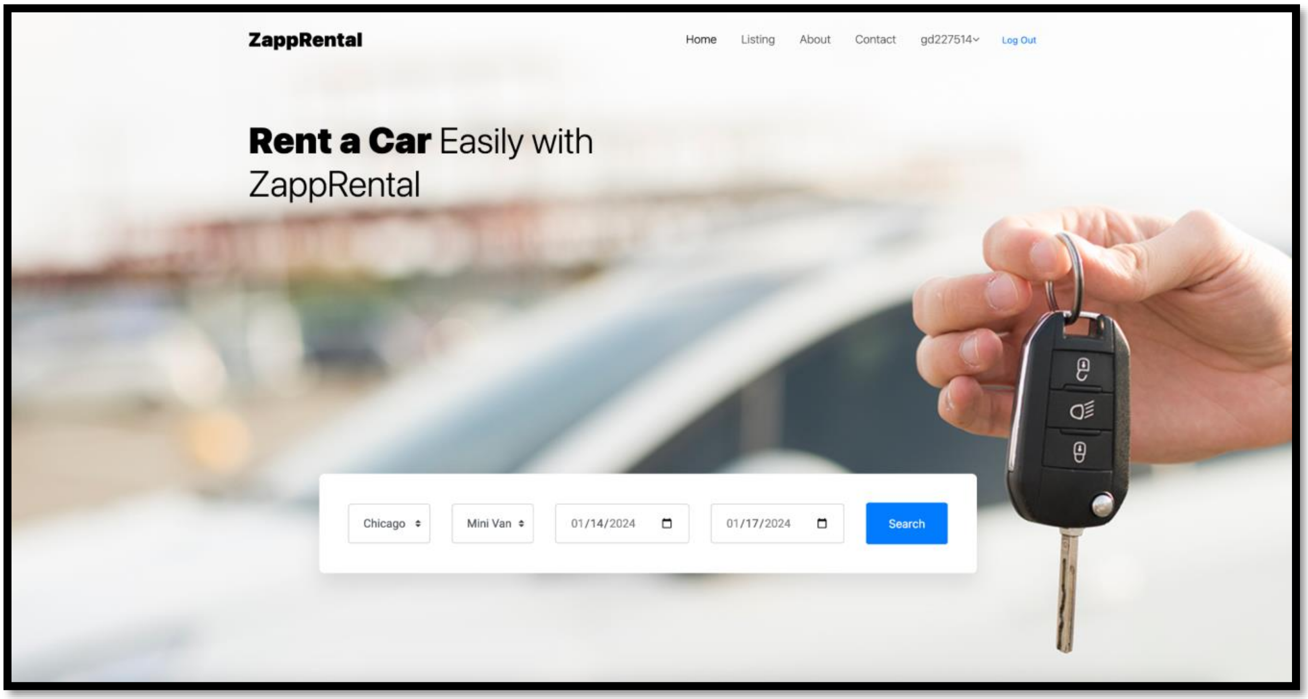
Action Output

```
     Time      Action                                                   Respon               Duration / Fetch Time
489  18:39:05  SELECT * FROM adg_car_Rental.adg_customer LIMIT 0, 1000   44 row(s) returned   0.00085 sec / 0.000...
490  18:40:56  SELECT * FROM adg_car_Rental.adg_corporate LIMIT 0, 1000  12 row(s) returned   0.00079 sec / 0.0000...
491  18:42:07  SELECT * FROM adg_car_Rental.adg_corporate LIMIT 0, 1000  12 row(s) returned   0.00057 sec / 0.0000...
492  18:42:23  SELECT * FROM adg_car_Rental.adg_individual LIMIT 0, 1000 13 row(s) returned   0.0013 sec / 0.00000...
493  18:42:31  SELECT * FROM adg_car_Rental.adg_corporate LIMIT 0, 1000  12 row(s) returned   0.00099 sec / 0.000...
494  18:47:10  SELECT * FROM adg_car_Rental.adg_customer LIMIT 0, 1000   47 row(s) returned   0.0013 sec / 0.00004...
495  18:49:22  SELECT * FROM adg_car_Rental.adg_corporate LIMIT 0, 1000  12 row(s) returned   0.00092 sec / 0.0000...
496  18:50:48  SELECT * FROM adg_car_Rental.adg_customer LIMIT 0, 1000   47 row(s) returned   0.0023 sec / 0.00013...
497  19:00:50  SELECT * FROM adg_car_Rental.adg_customer LIMIT 0, 1000   48 row(s) returned   0.00059 sec / 0.000...
498  19:00:57  SELECT * FROM adg_car_Rental.adg_corporate LIMIT 0, 1000  12 row(s) returned   0.00096 sec / 0.000...
499  19:05:46  SELECT * FROM adg_car_Rental.adg_corporate LIMIT 0, 1000  12 row(s) returned   0.0015 sec / 0.00000...
500  19:05:49  SELECT * FROM adg_car_Rental.adg_individual LIMIT 0, 1000 14 row(s) returned   0.0017 sec / 0.00001...
```



Rent a Car Easily with
ZappRental

Searching for a car:



Rent a Car Easily with ZappRental

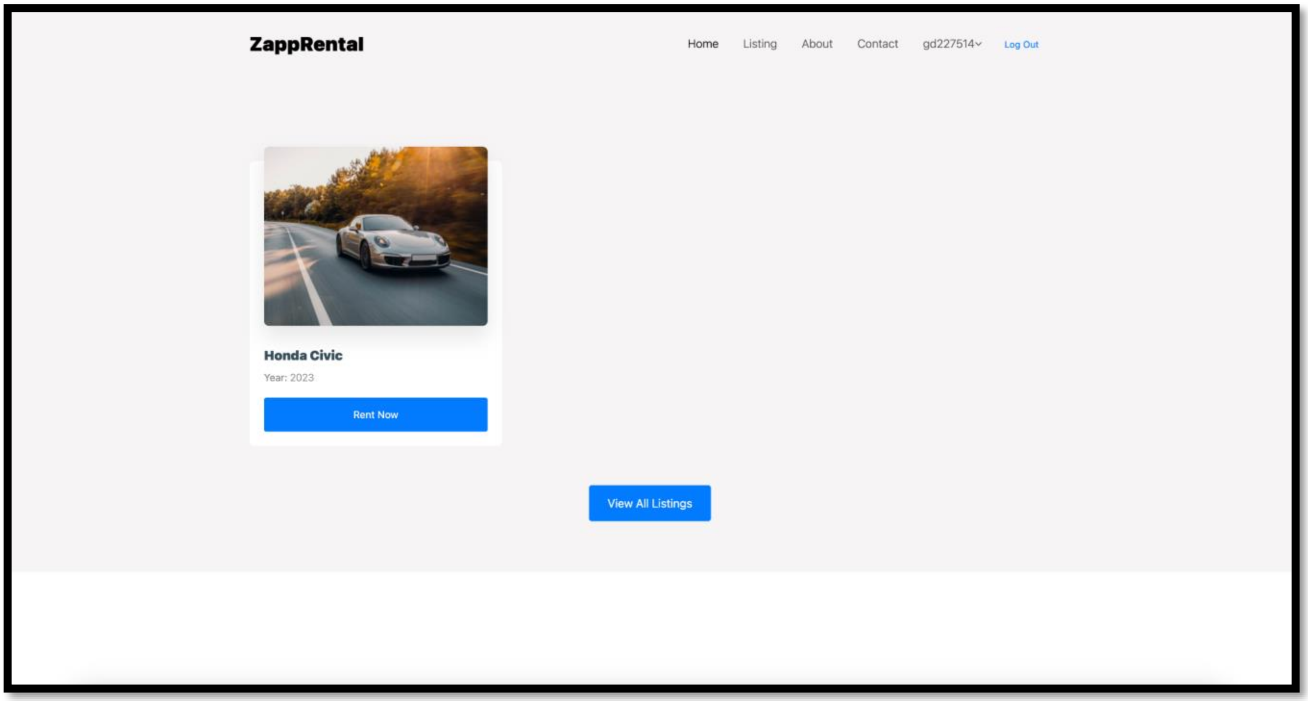
Chicago

Mini Van

01/14/2024

01/17/2024

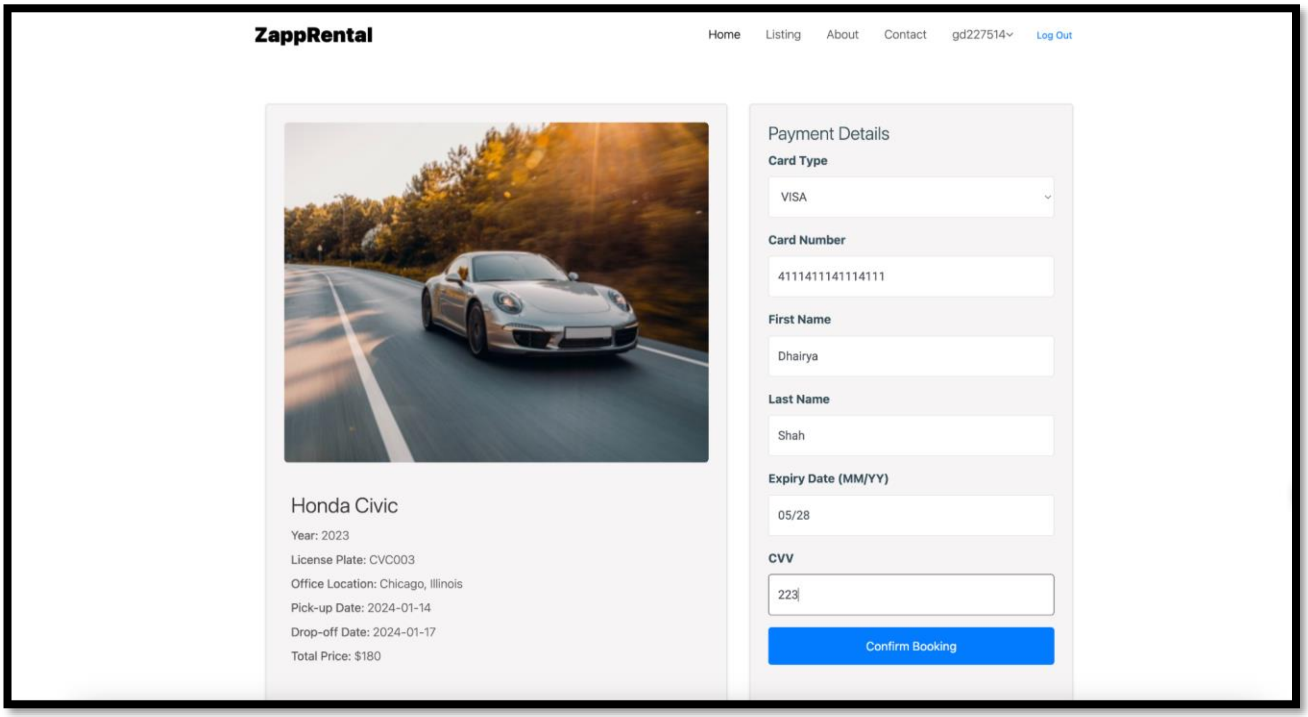
Search results page:



Honda Civic

Year: 2023

Booking page:



Honda Civic

Year: 2023

License Plate: CVC003

Office Location: Chicago, Illinois

Pick-up Date: 2024-01-14

Drop-off Date: 2024-01-17

Total Price: $180

Payment Details

Card Type: VISA

Card Number: 4111411141114111

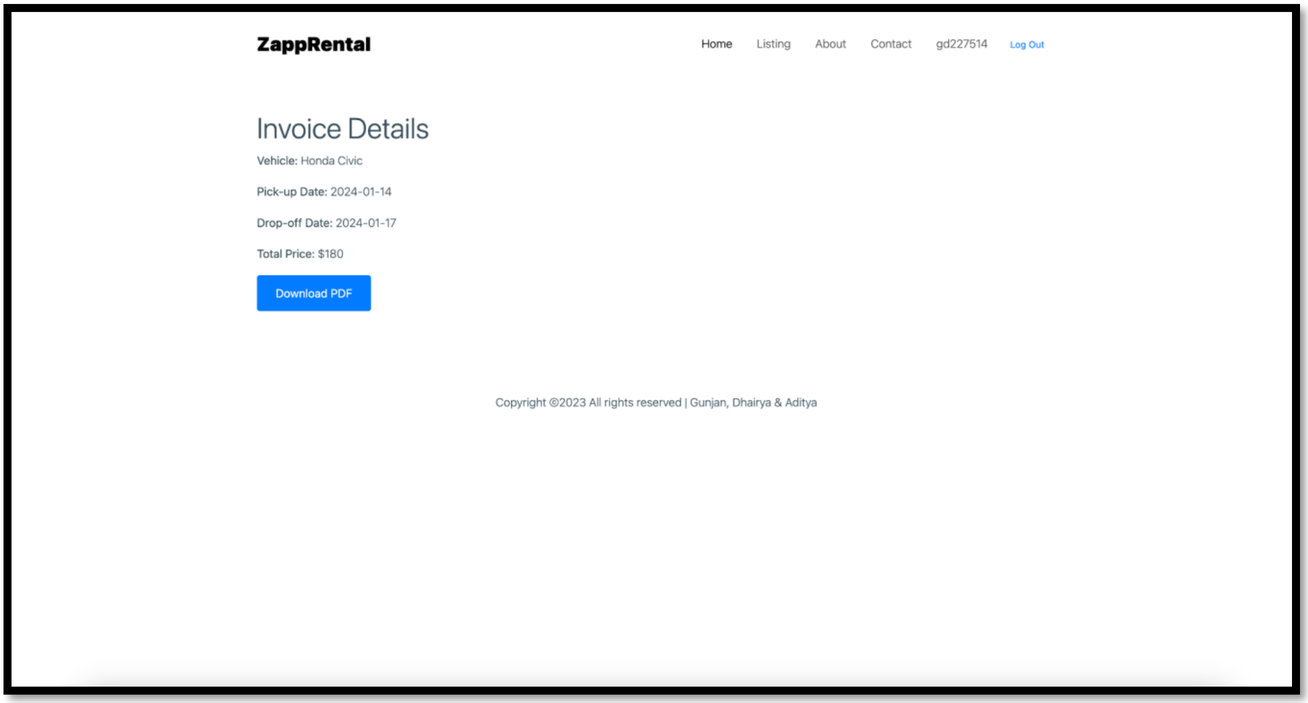
First Name: Dhairya

Last Name: Shah

Expiry Date (MM/YY): 05/28

CVV: 223

Invoice Page:



Invoice Details

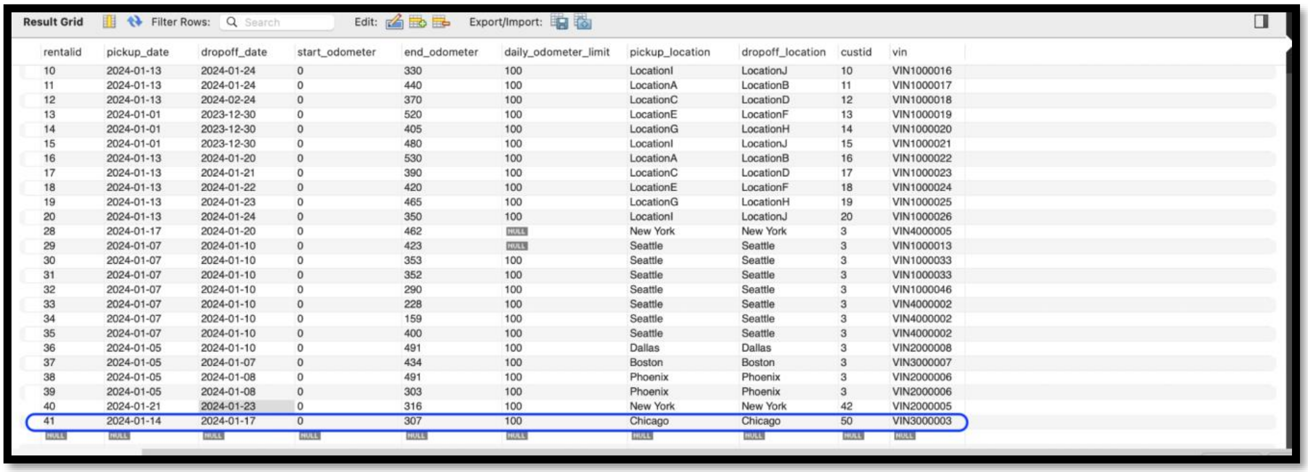
Vehicle: Honda Civic

Pick-up Date: 2024-01-14

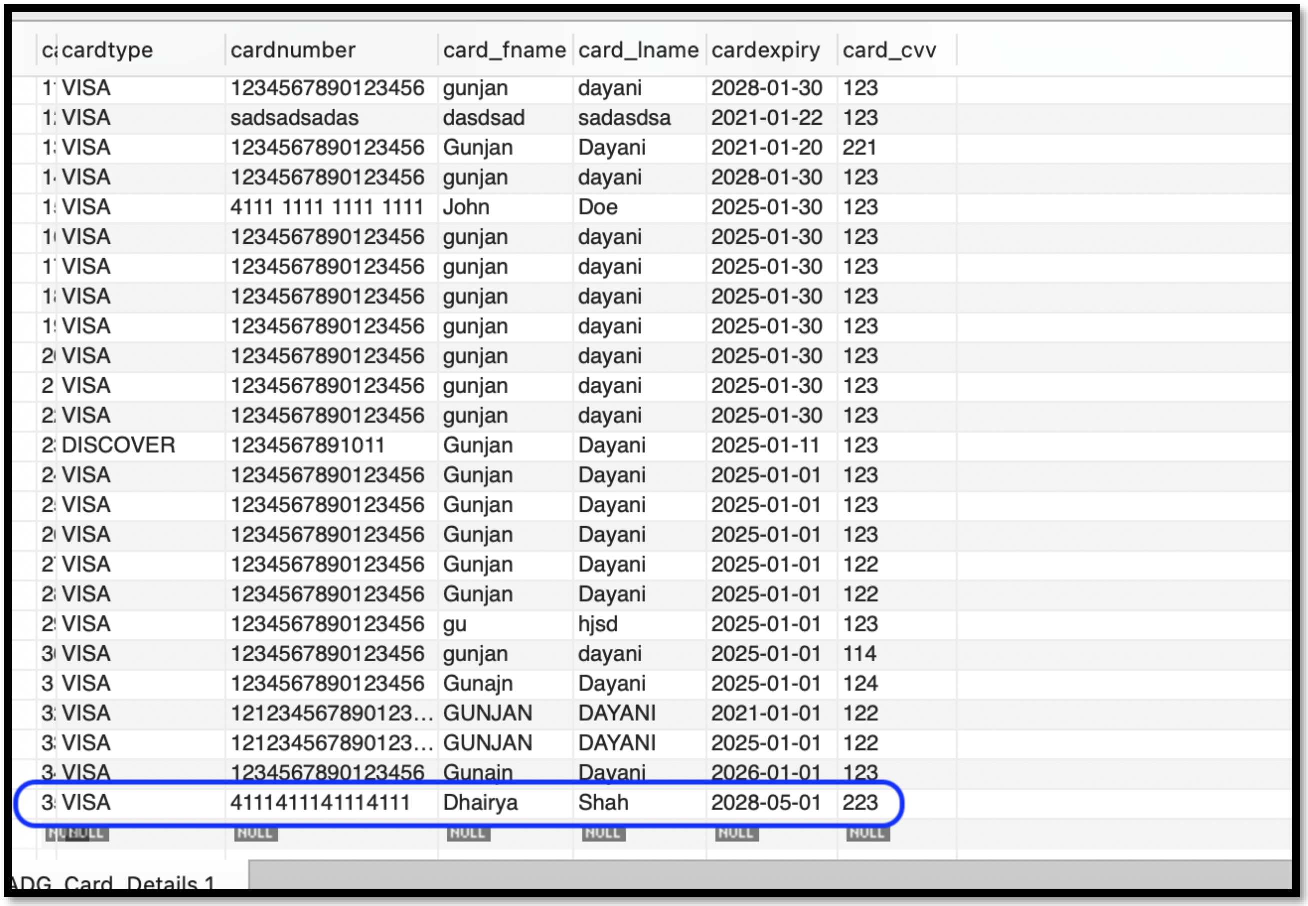
Drop-off Date: 2024-01-17

Total Price: $180

Copyright ©2023 All rights reserved | Gunjan, Dhairya & Aditya



rentalid	pickup_date	dropoff_date	start_odometer	end_odometer	daily_odometer_limit	pickup_location	dropoff_location	custid	vin
10	2024-01-13	2024-01-24	0	330	100	LocationI	LocationJ	10	VIN1000016
11	2024-01-13	2024-01-24	0	440	100	LocationA	LocationB	11	VIN1000017
12	2024-01-13	2024-02-24	0	370	100	LocationC	LocationD	12	VIN1000018
13	2024-01-01	2023-12-30	0	520	100	LocationE	LocationF	13	VIN1000019
14	2024-01-01	2023-12-30	0	405	100	LocationG	LocationH	14	VIN1000020
15	2024-01-01	2023-12-30	0	480	100	LocationI	LocationJ	15	VIN1000021
16	2024-01-13	2024-01-20	0	530	100	LocationA	LocationB	16	VIN1000022
17	2024-01-13	2024-01-21	0	390	100	LocationC	LocationD	17	VIN1000023
18	2024-01-13	2024-01-22	0	420	100	LocationE	LocationF	18	VIN1000024
19	2024-01-13	2024-01-23	0	465	100	LocationG	LocationH	19	VIN1000025
20	2024-01-13	2024-01-24	0	350	100	LocationI	LocationJ	20	VIN1000026
28	2024-01-17	2024-01-20	0	462	NULL	New York	New York	3	VIN4000005
29	2024-01-07	2024-01-10	0	423	NULL	Seattle	Seattle	3	VIN1000013
30	2024-01-07	2024-01-10	0	353	100	Seattle	Seattle	3	VIN1000033
31	2024-01-07	2024-01-10	0	352	100	Seattle	Seattle	3	VIN1000033
32	2024-01-07	2024-01-10	0	290	100	Seattle	Seattle	3	VIN1000046
33	2024-01-07	2024-01-10	0	228	100	Seattle	Seattle	3	VIN4000002
34	2024-01-07	2024-01-10	0	159	100	Seattle	Seattle	3	VIN4000002
35	2024-01-07	2024-01-10	0	400	100	Seattle	Seattle	3	VIN4000002
36	2024-01-05	2024-01-10	0	491	100	Dallas	Dallas	3	VIN2000008
37	2024-01-05	2024-01-07	0	434	100	Boston	Boston	3	VIN3000007
38	2024-01-05	2024-01-08	0	491	100	Phoenix	Phoenix	3	VIN2000006
39	2024-01-05	2024-01-08	0	303	100	Phoenix	Phoenix	3	VIN2000006
40	2024-01-21	2024-01-23	0	316	100	New York	New York	42	VIN2000005
41	2024-01-14	2024-01-17	0	307	100	Chicago	Chicago	50	VIN3000003
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL



c	cardtype	cardnumber	card_fname	card_lname	cardexpiry	card_cvv
1	VISA	1234567890123456	gunjan	dayani	2028-01-30	123
1:	VISA	sadsadsadas	dasdsad	sadasdsa	2021-01-22	123
1:	VISA	1234567890123456	Gunjan	Dayani	2021-01-20	221
1·	VISA	1234567890123456	gunjan	dayani	2028-01-30	123
1!	VISA	4111 1111 1111 1111	John	Doe	2025-01-30	123
1(	VISA	1234567890123456	gunjan	dayani	2025-01-30	123
1	VISA	1234567890123456	gunjan	dayani	2025-01-30	123
1:	VISA	1234567890123456	gunjan	dayani	2025-01-30	123
1!	VISA	1234567890123456	gunjan	dayani	2025-01-30	123
2(	VISA	1234567890123456	gunjan	dayani	2025-01-30	123
2	VISA	1234567890123456	gunjan	dayani	2025-01-30	123
2:	VISA	1234567890123456	gunjan	dayani	2025-01-30	123
2:	DISCOVER	1234567891011	Gunjan	Dayani	2025-01-11	123
2·	VISA	1234567890123456	Gunjan	Dayani	2025-01-01	123
2!	VISA	1234567890123456	Gunjan	Dayani	2025-01-01	123
2(	VISA	1234567890123456	Gunjan	Dayani	2025-01-01	123
2	VISA	1234567890123456	Gunjan	Dayani	2025-01-01	122
2:	VISA	1234567890123456	Gunjan	Dayani	2025-01-01	122
2!	VISA	1234567890123456	gu	hjsd	2025-01-01	123
3(	VISA	1234567890123456	gunjan	dayani	2025-01-01	114
3	VISA	1234567890123456	Gunajn	Dayani	2025-01-01	124
3:	VISA	121234567890123…	GUNJAN	DAYANI	2021-01-01	122
3:	VISA	121234567890123…	GUNJAN	DAYANI	2025-01-01	122
3·	VISA	1234567890123456	Gunjan	Dayani	2026-01-01	123
3:	VISA	4111411141114111	Dhairya	Shah	2028-05-01	223
NU	NULL	NULL	NULL	NULL	NULL	NULL

Details of security features / Extra Credit:

Password Encryption: Password encryption in databases is a critical security measure employed to safeguard sensitive user information. Encryption ensures that passwords are stored in an unreadable format, providing an additional layer of protection against unauthorized access. When a user creates or updates their password, it goes through a hashing or encryption algorithm that transforms it into a unique, irreversible string of characters. This process ensures that even if the database is compromised, the original passwords cannot be easily deciphered. When a user attempts to log in, their entered password is encrypted using the same algorithm and compared against the stored encrypted password. If they match, access is granted.

SQL Injection Protection: Implementing stored procedures is an effective approach to prevent SQL injections and enhance the security of a database. Stored procedures are pre-compiled and stored in the database itself. They allow developers to define a set of SQL statements as a single unit, which can be executed repeatedly with different parameters.

Validation of each field in form: Validating each field in a form is essential for security and data integrity. It prevents malicious data entry, protecting against attacks like SQL injections and cross-site scripting. Field validation ensures that only correct data types and formats are entered, safeguarding the application from unauthorized manipulations and maintaining its stability. This feature is fundamental for any secure and reliable application.

Session Management: Session management is a critical aspect of web security. It is the process of securely handling the user's session from login to logout. Proper session management prevents unauthorized access to a user's session and the sensitive information within it. When a user logs in, the system should generate a new session identifier (session ID) that is complex, unpredictable, and securely stored on the server. The session ID is then passed back to the user's browser to maintain the state. The user's session should be correctly invalidated upon logout, and the session ID should be renewed at regular intervals and definitely upon reauthentication. It is also important to implement secure transmission (e.g., HTTPS) for all session-related communication to prevent session hijacking attacks.

Lessons Learned

Reflections:

Comprehensive Requirements Gathering: The project highlighted the importance of thorough requirements analysis. Taking the time to fully understand the business needs and objectives upfront helps in designing a database system that meets those requirements accurately

Effective Data Modeling: The process of entity identification, attribute definition, and relationship mapping is crucial for successful data modeling. It is essential to capture all relevant entities and their relationships accurately to establish data integrity and maintain consistency

Data Consistency and Integrity: Applying CHECK constraints to enforce data consistency is essential. It prevents invalid or inconsistent data from entering the system, ensuring reliable and accurate information. Careful consideration of domain constraints for attributes helps in maintaining data integrity and reducing data quality issues.

Scalability and Flexibility: The database system should be designed with scalability and flexibility in mind. As the car rental company grows and evolves, the database should be able to handle increasing volumes of data and accommodate new features or requirements. This requires careful consideration of table structures, indexes, and performance optimization techniques.

Continuous Monitoring and Maintenance: A centralized relational database system requires ongoing monitoring and maintenance to ensure its optimal performance. Regular backups, updates, and monitoring of system health are essential for data protection and system reliability. It is also important to regularly review and update the database schema to adapt to changing business requirements

Collaboration and Communication: Effective collaboration and communication between the project team, stakeholders, and end-users are key to project success. Regular communication channels, such as meetings or progress reports, help in aligning expectations, resolving issues, and ensuring that the final database system meets the needs of all stakeholders.

Constraints:

Time Constraints: One of the primary constraints faced during the project was the time limitation. Developing a centralized relational database system for a car rental company involves thorough analysis, design, implementation, and testing. Time constraints can impact the depth of requirements analysis, the level of system testing, and the overall project timeline. It is crucial to manage time effectively, prioritize tasks, and ensure that key project milestones are met.

Technological Constraints: The choice of a suitable database management system (DBMS) can be influenced by technological constraints. Existing technology infrastructure, compatibility with other systems, and organizational preferences can limit the selection of a DBMS. It is important to evaluate the capabilities of the chosen DBMS and ensure that it can support the required functionalities and performance expectations.

Data Availability and Quality: The availability and quality of data required for the database system can be a significant constraint. Incomplete or inconsistent data can impact the accuracy and reliability of the system. It may require additional effort to clean and validate the existing data or establish data acquisition processes to ensure data integrity. Collaboration with relevant departments or data sources is crucial to address this constraint effectively.

Business Analysis with SQL using project data:

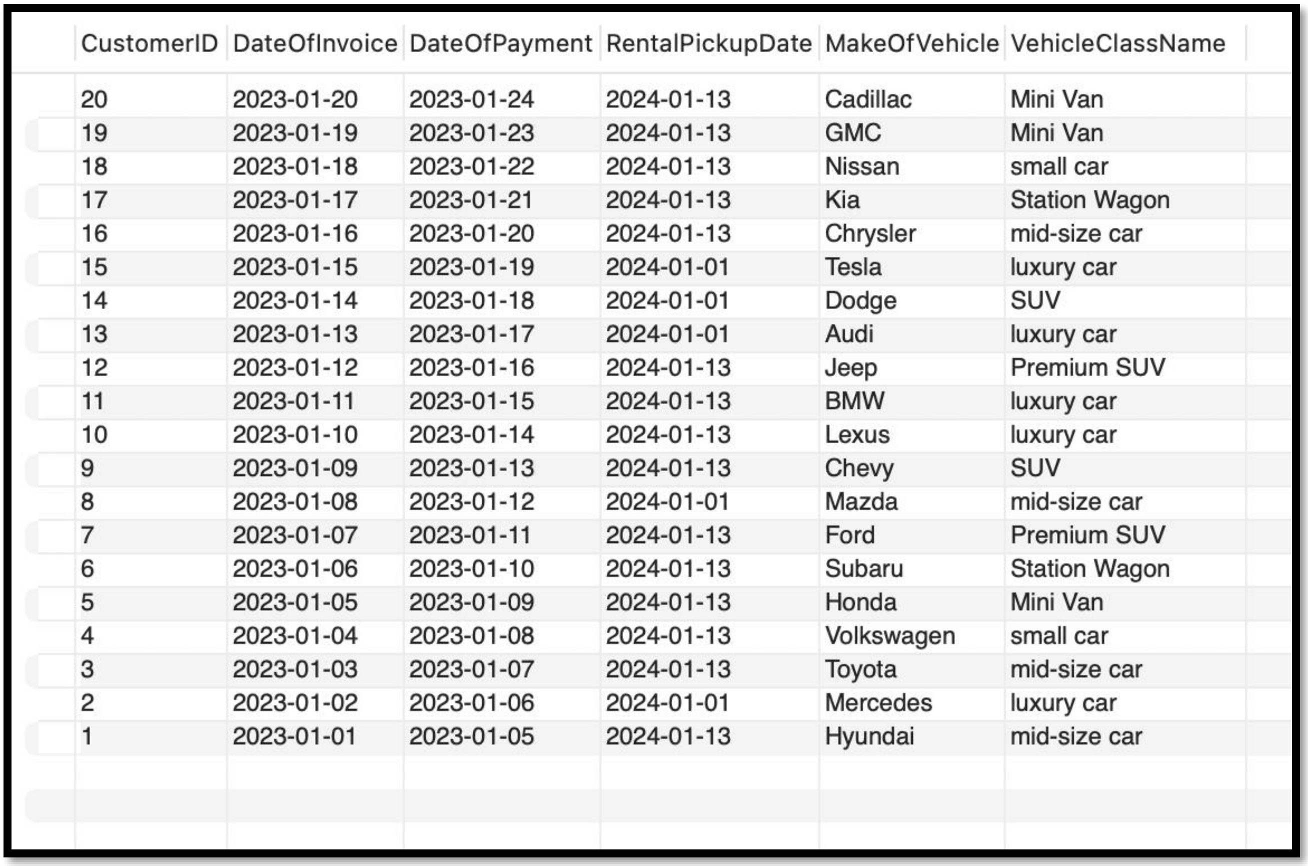
For each of above queries use a proper column alias, built-in functions, appropriate sorting and submit following three items for each of above queries.

Q1) Table joins with at least 3 tables in join.

→ Query:

```
SELECT
   r.CustID AS CustomerID,
   i.InvoiceDate AS DateOfInvoice,
   p.PaymentDate AS DateOfPayment,
   r.Pickup_Date AS RentalPickupDate,
   v.VehicleMake AS MakeOfVehicle,
   vc.ClassName AS VehicleClassName
FROM
   ADG_RentalService r
JOIN
   ADG_Invoice i ON r.RentalID = i.RentalID
JOIN
   ADG_Payment p ON i.InvoiceID = p.InvoiceID
JOIN
   ADG_Vehicle v ON r.VIN = v.VIN
JOIN
   ADG_VehicleClass vc ON v.ClassID = vc.ClassID
ORDER BY
   DateOfInvoice DESC;
```

Result:

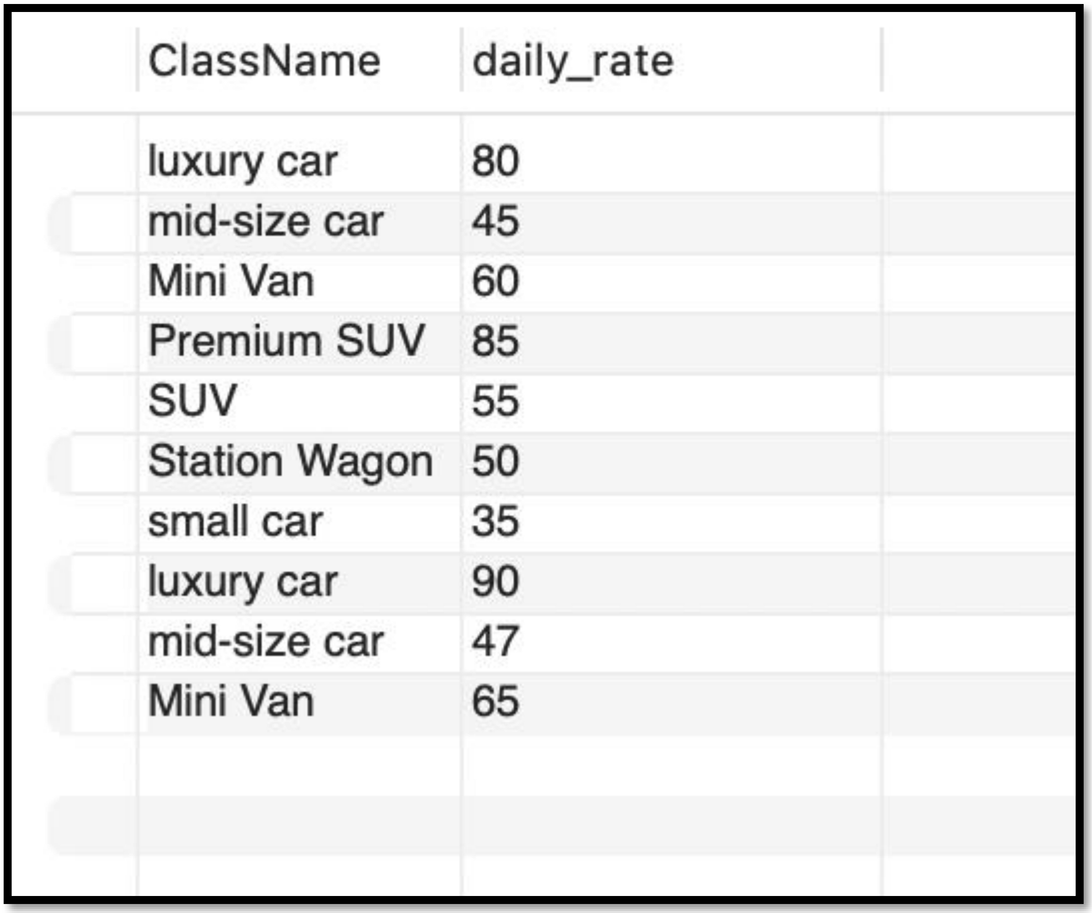


CustomerID	DateOfInvoice	DateOfPayment	RentalPickupDate	MakeOfVehicle	VehicleClassName
20	2023-01-20	2023-01-24	2024-01-13	Cadillac	Mini Van
19	2023-01-19	2023-01-23	2024-01-13	GMC	Mini Van
18	2023-01-18	2023-01-22	2024-01-13	Nissan	small car
17	2023-01-17	2023-01-21	2024-01-13	Kia	Station Wagon
16	2023-01-16	2023-01-20	2024-01-13	Chrysler	mid-size car
15	2023-01-15	2023-01-19	2024-01-01	Tesla	luxury car
14	2023-01-14	2023-01-18	2024-01-01	Dodge	SUV
13	2023-01-13	2023-01-17	2024-01-01	Audi	luxury car
12	2023-01-12	2023-01-16	2024-01-13	Jeep	Premium SUV
11	2023-01-11	2023-01-15	2024-01-13	BMW	luxury car
10	2023-01-10	2023-01-14	2024-01-13	Lexus	luxury car
9	2023-01-09	2023-01-13	2024-01-13	Chevy	SUV
8	2023-01-08	2023-01-12	2024-01-01	Mazda	mid-size car
7	2023-01-07	2023-01-11	2024-01-13	Ford	Premium SUV
6	2023-01-06	2023-01-10	2024-01-13	Subaru	Station Wagon
5	2023-01-05	2023-01-09	2024-01-13	Honda	Mini Van
4	2023-01-04	2023-01-08	2024-01-13	Volkswagen	small car
3	2023-01-03	2023-01-07	2024-01-13	Toyota	mid-size car
2	2023-01-02	2023-01-06	2024-01-01	Mercedes	luxury car
1	2023-01-01	2023-01-05	2024-01-13	Hyundai	mid-size car

Q2) Multi-row subquery
→ Query:

```
SELECT
    ClassName,
    daily_rate
FROM
    ADG_VehicleClass
WHERE
    ClassID IN (
        SELECT
            ClassID
        FROM
            ADG_Vehicle
        WHERE
```

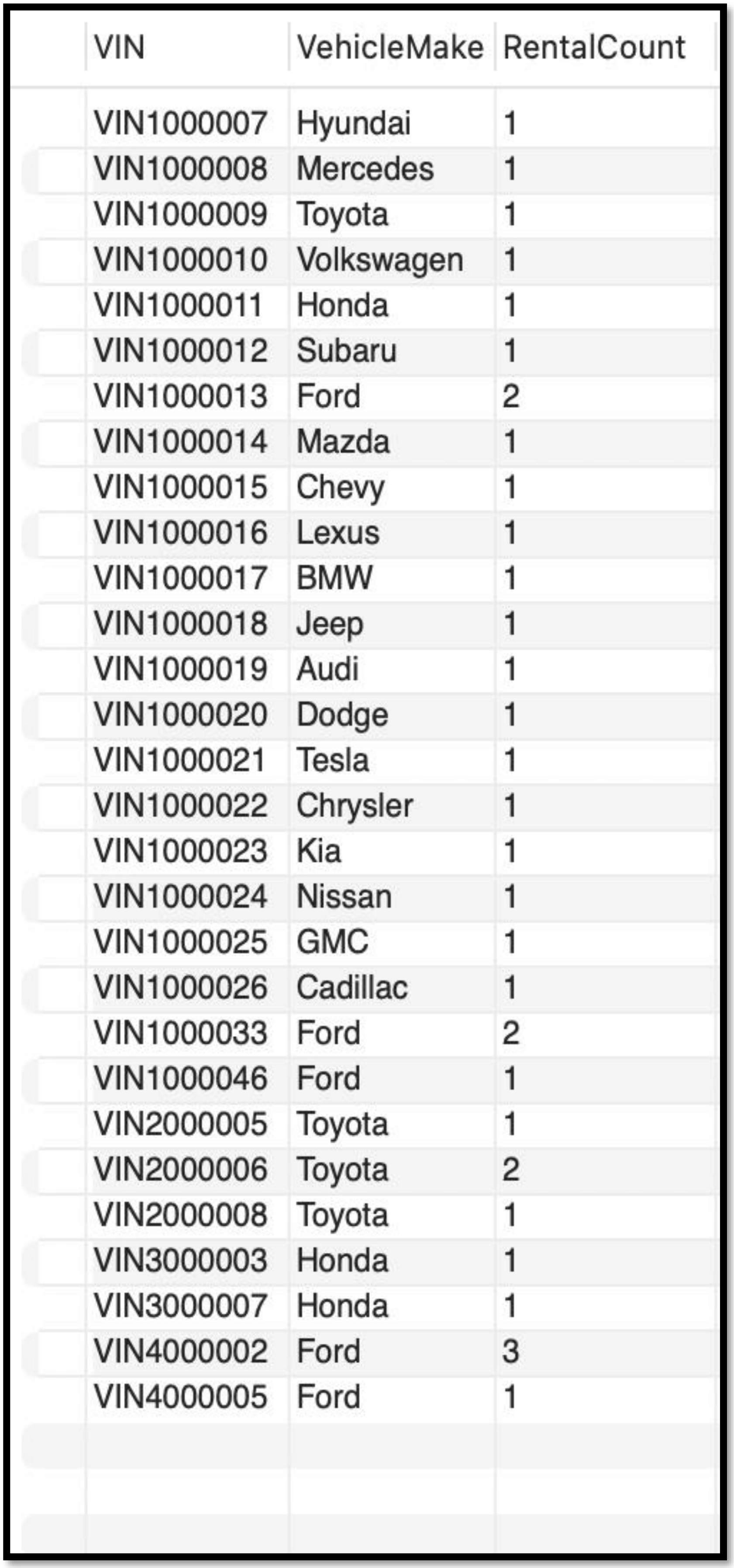
```
            VehicleYear > 2015
        );
```



ClassName	daily_rate
luxury car	80
mid-size car	45
Mini Van	60
Premium SUV	85
SUV	55
Station Wagon	50
small car	35
luxury car	90
mid-size car	47
Mini Van	65

Q3) Correlated subquery.
→ Query:

```
SELECT
    v.VIN,
    v.VehicleMake,
    (SELECT
        COUNT(*)
     FROM
        ADG_RentalService r
     WHERE
     r.VIN = v.VIN
   ) AS RentalCount
FROM
  ADG_Vehicle v
WHERE
  EXISTS (
    SELECT
      1
    FROM
      ADG_RentalService r
    WHERE
      r.VIN = v.VIN
      AND r.Pickup_Date >= '2023-01-01'
  );
```

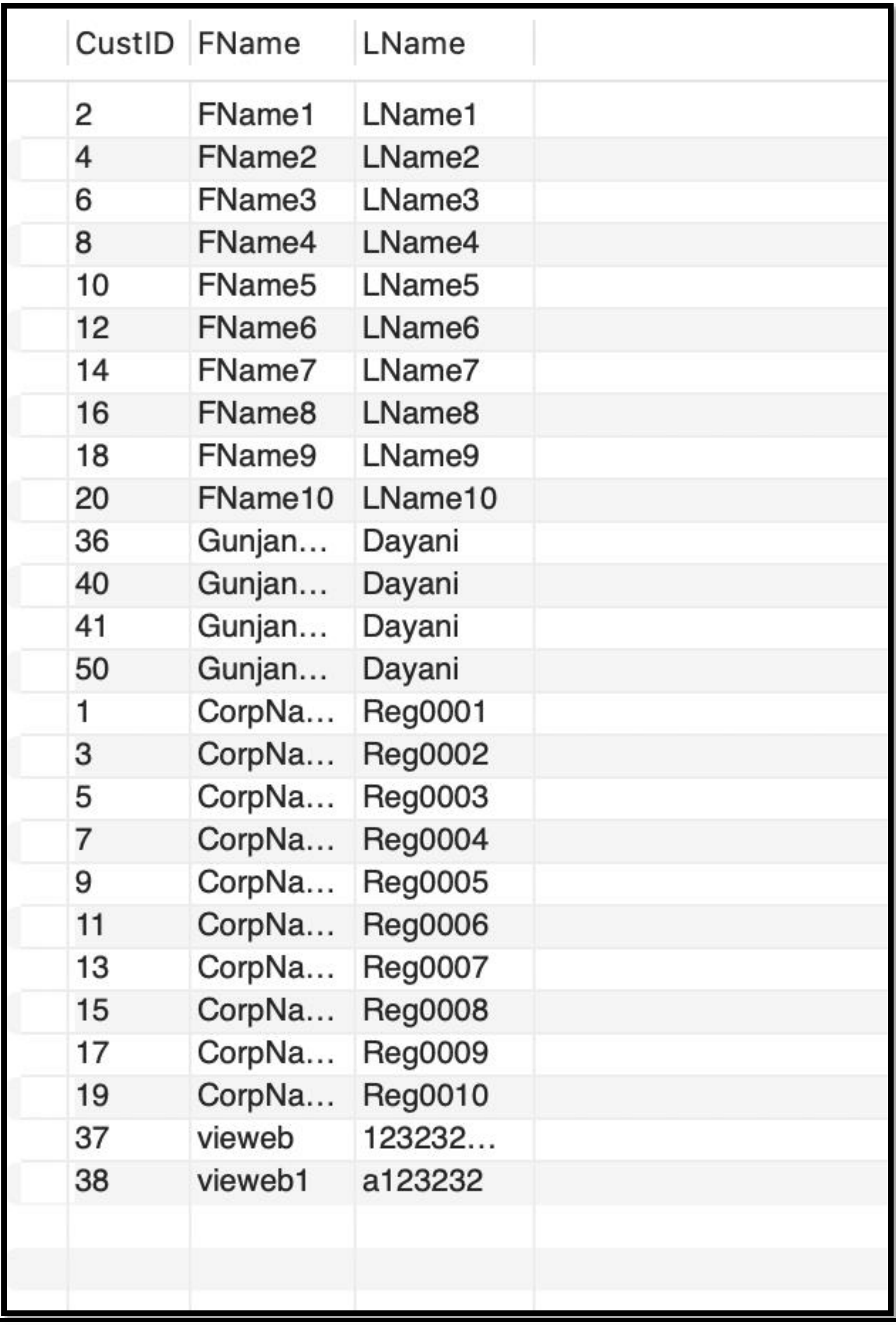


VIN	VehicleMake	RentalCount
VIN1000007	Hyundai	1
VIN1000008	Mercedes	1
VIN1000009	Toyota	1
VIN1000010	Volkswagen	1
VIN1000011	Honda	1
VIN1000012	Subaru	1
VIN1000013	Ford	2
VIN1000014	Mazda	1
VIN1000015	Chevy	1
VIN1000016	Lexus	1
VIN1000017	BMW	1
VIN1000018	Jeep	1
VIN1000019	Audi	1
VIN1000020	Dodge	1
VIN1000021	Tesla	1
VIN1000022	Chrysler	1
VIN1000023	Kia	1
VIN1000024	Nissan	1
VIN1000025	GMC	1
VIN1000026	Cadillac	1
VIN1000033	Ford	2
VIN1000046	Ford	1
VIN2000005	Toyota	1
VIN2000006	Toyota	2
VIN2000008	Toyota	1
VIN3000003	Honda	1
VIN3000007	Honda	1
VIN4000002	Ford	3
VIN4000005	Ford	1

Q4) SET operator query.

→ Query:

```
SELECT
   CustID,
   FName,
   LName
FROM
   ADG_Individual
UNION
SELECT
   CustID,
   Corporation_Name,
   Registration_Number
FROM
   ADG_Corporate;
```

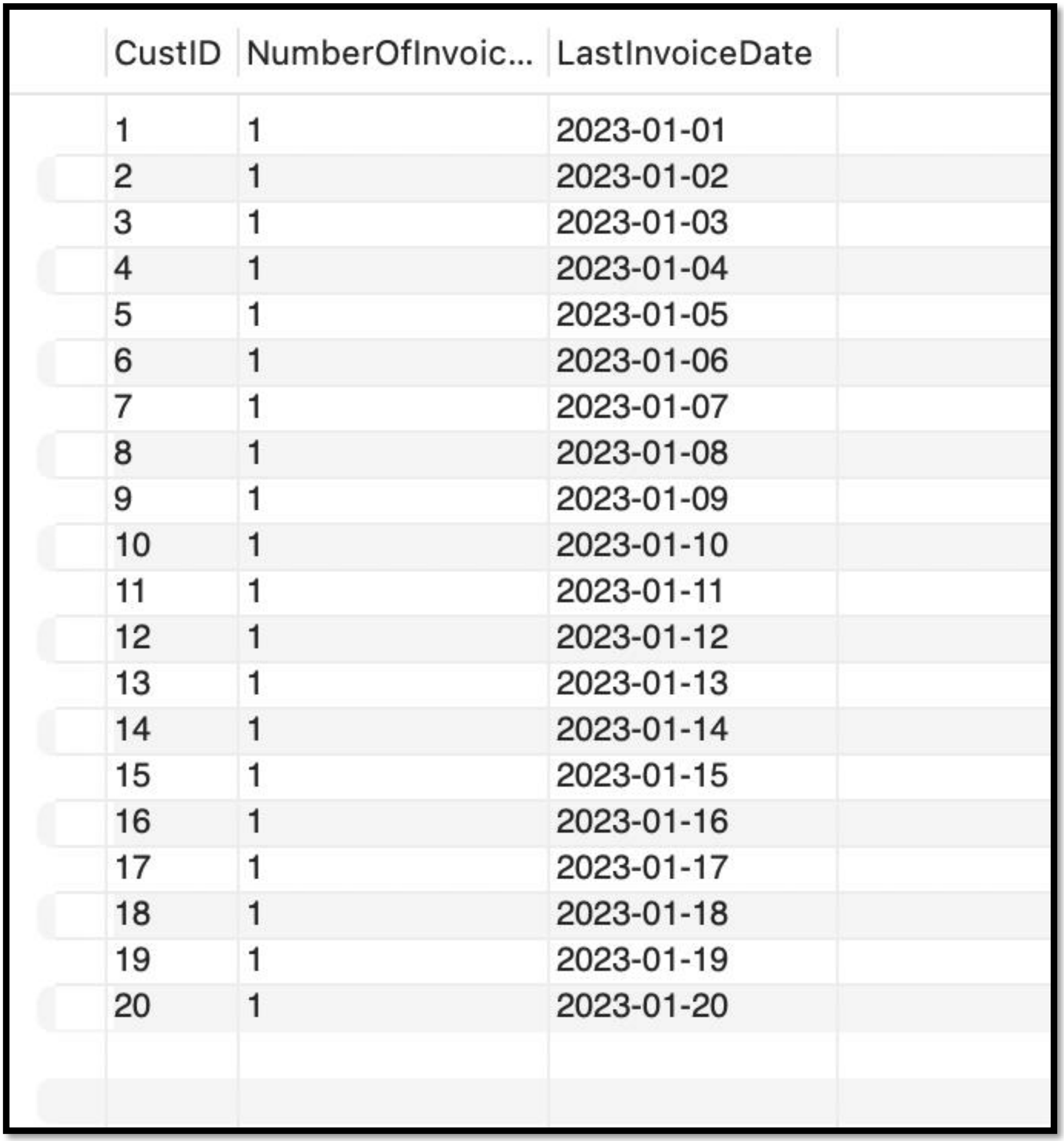


CustID	FName	LName
2	FName1	LName1
4	FName2	LName2
6	FName3	LName3
8	FName4	LName4
10	FName5	LName5
12	FName6	LName6
14	FName7	LName7
16	FName8	LName8
18	FName9	LName9
20	FName10	LName10
36	Gunjan…	Dayani
40	Gunjan…	Dayani
41	Gunjan…	Dayani
50	Gunjan…	Dayani
1	CorpNa…	Reg0001
3	CorpNa…	Reg0002
5	CorpNa…	Reg0003
7	CorpNa…	Reg0004
9	CorpNa…	Reg0005
11	CorpNa…	Reg0006
13	CorpNa…	Reg0007
15	CorpNa…	Reg0008
17	CorpNa…	Reg0009
19	CorpNa…	Reg0010
37	vieweb	123232…
38	vieweb1	a123232

Q5) Query with in-line view or WITH clause

→ Query:

```
WITH CustomerInvoices AS (
    SELECT
        r.CustID,
        i.InvoiceID,
        i.InvoiceDate
    FROM
        ADG_RentalService r
    JOIN
        ADG_Invoice i ON r.RentalID = i.RentalID
)
SELECT
    CustID,
    COUNT(InvoiceID) AS NumberOfInvoices,
    MAX(InvoiceDate) AS LastInvoiceDate
FROM
    CustomerInvoices
GROUP BY
    CustID;
```

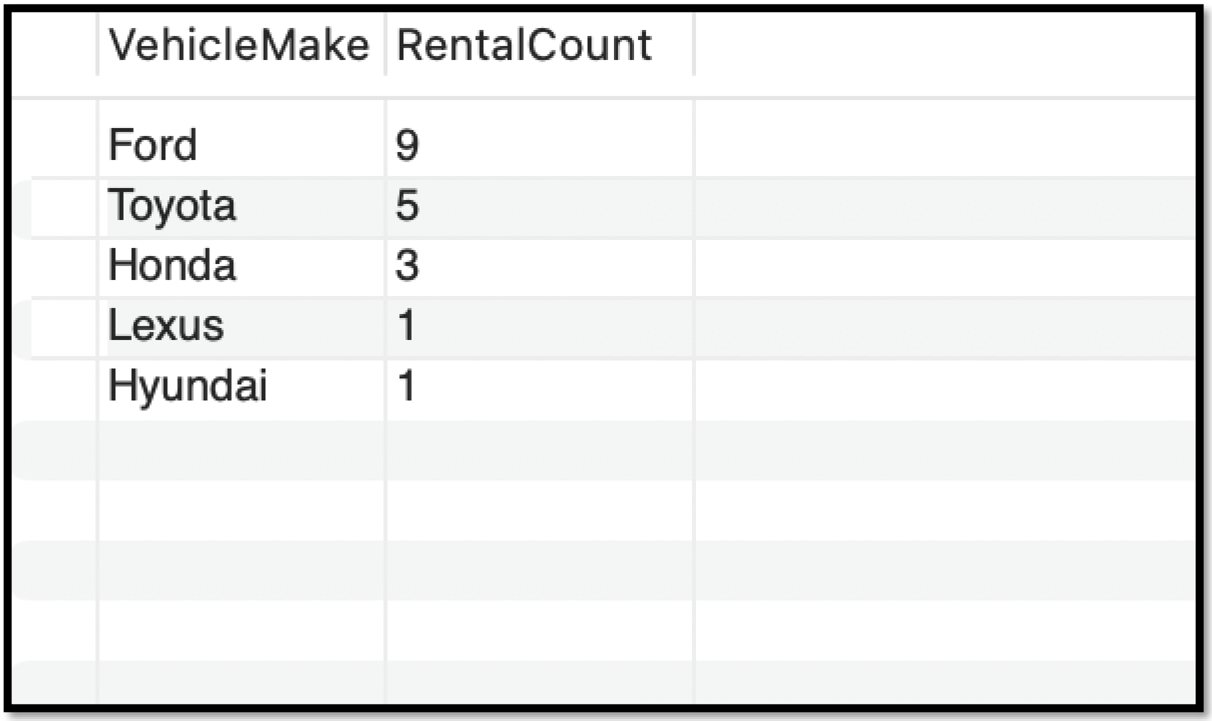


CustID	NumberOfInvoic...	LastInvoiceDate
1	1	2023-01-01
2	1	2023-01-02
3	1	2023-01-03
4	1	2023-01-04
5	1	2023-01-05
6	1	2023-01-06
7	1	2023-01-07
8	1	2023-01-08
9	1	2023-01-09
10	1	2023-01-10
11	1	2023-01-11
12	1	2023-01-12
13	1	2023-01-13
14	1	2023-01-14
15	1	2023-01-15
16	1	2023-01-16
17	1	2023-01-17
18	1	2023-01-18
19	1	2023-01-19
20	1	2023-01-20

Q6) TOP-N/BOTTOM-N query:
→ Query:

```
SELECT
    VehicleMake,
    COUNT(*) AS RentalCount
```

```
FROM
   ADG_Vehicle v
JOIN
   ADG_RentalService r ON v.VIN = r.VIN
GROUP BY
   VehicleMake
ORDER BY
   RentalCount DESC
LIMIT 5;
```



VehicleMake	RentalCount
Ford	9
Toyota	5
Honda	3
Lexus	1
Hyundai	1