CS_6083_B_Intro to Database_2022_Spring

Project_Part_2

Jianhao Li(jl12173), Yimo Liu(yl9484), Ziyan Qiu(zq2048)

Submission date: 05/12/2022

Content table:

Introduction:

We designed the car rental service for WOW. Our works include Database design, and associated web interface function(Including front and back-end). Our systems

Database Design:

The database design would be like following pictures.

The Customers are separated as INDIVIDUAL(Individual customer) and CORPORATE_CUSTOMER. They are subtypes under CUSTOMER Table. The CORPORATE_CUSTOMER is associated with CORPORATE table. CUSTOMER table is

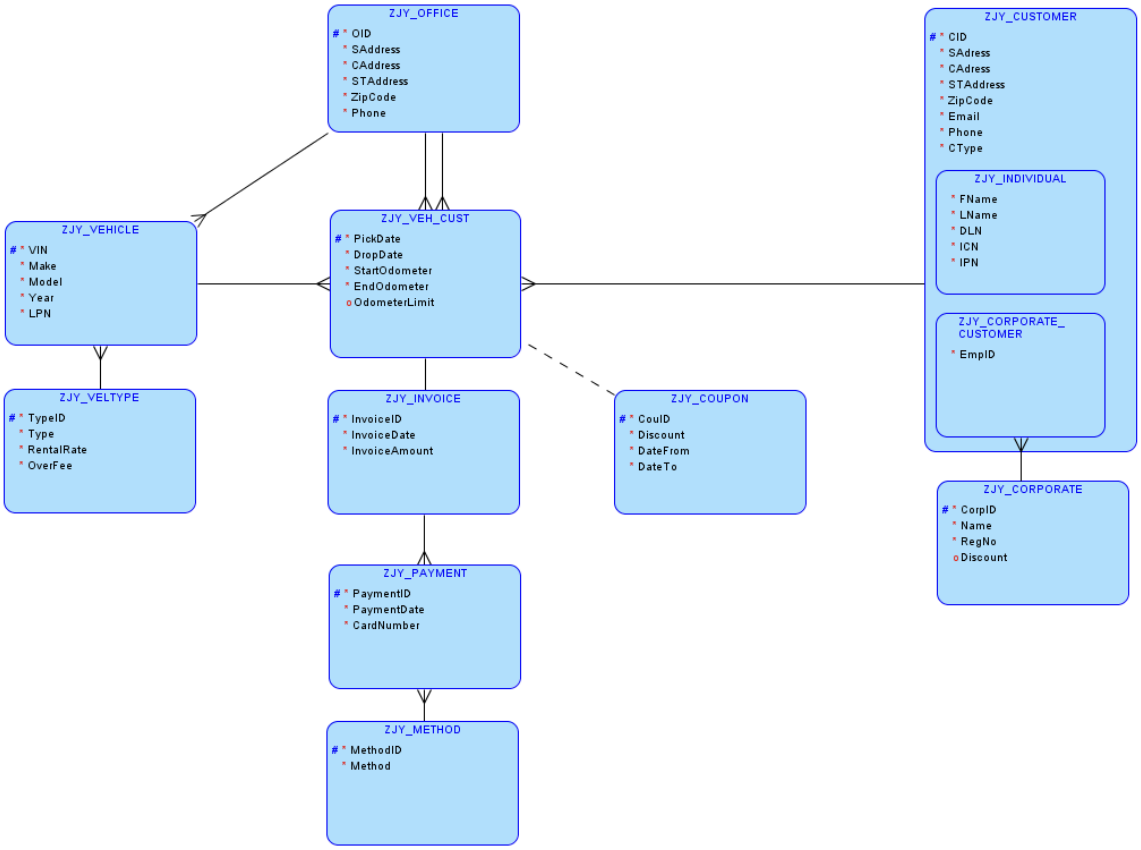
multi-to-multi relationship with VEHICLE. The associate VEH_CUST table indicates the transaction. Each entry in it means one customer rent one car for one time.

For each entry in VEH_CUST, once the transaction is done(when DropDate and EndOdometer are confirmed). Invoice would be auto-generated. Also, the details of the payment get recorded in PAYMENT and METHOD table.

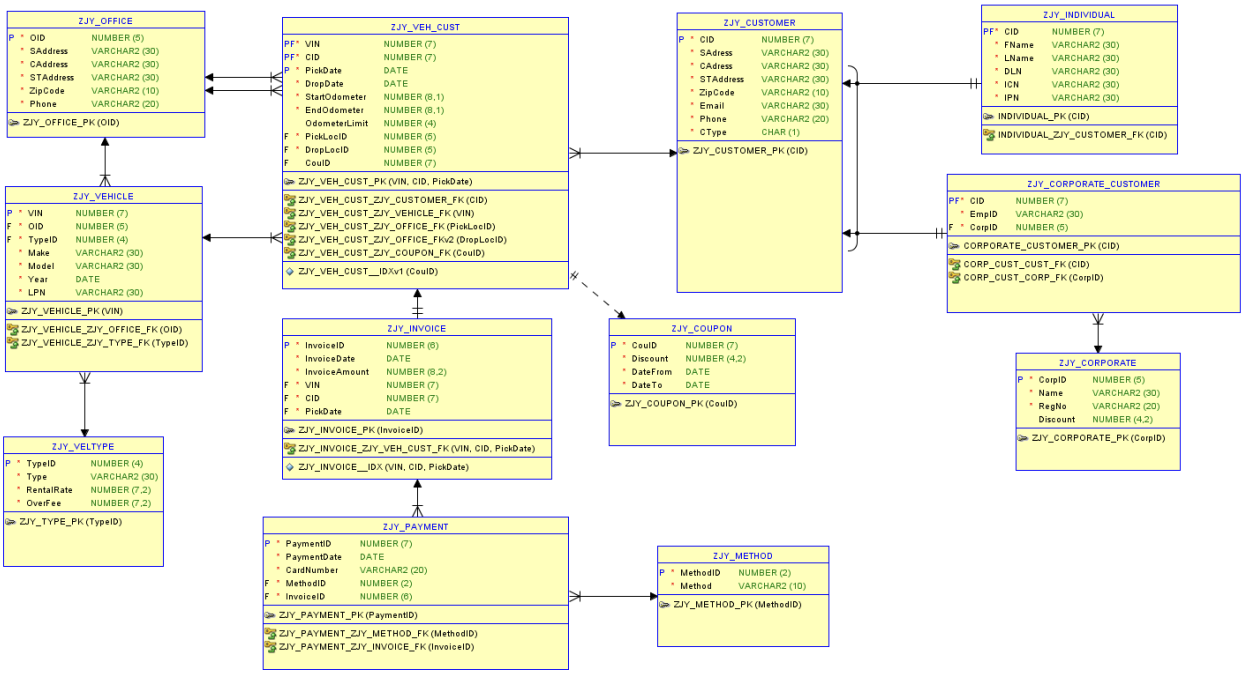
Assumptions:

- Use OID to identify an office
- Use TypeID to identify the type of the rental vehicle Use CID to identify a customer
- Use CouID to identify a kind of coupon A customer must rent a vehicle to be a customer
- Use InvoiceID to identify an invoice
- Use PaymentID to identify a payment The pick up and drop off location of each rental must be the same as office location
- Use CorpID to identify the corporate
- The corporate customer may or may not have a discount
- The EmpID is used in the customer's corporate database to uniquely distinguish a record Coupon can exist without rental, a rent can happen without a coupon
- When a employee rent a vehicle for himself, it is treated as individual customer
- Invoice date is the drop off date of the vehicle.

Logical model:



Relational model:



Developing technology stack:

We use vue.js for the frontend and Django for the backend. The backend is a restful api server, so we can develop the frontend and backend independently. The database is MySQL. Some IDEs and tools we use are: WebStrom, PyCharm, MySQL Workbench, Postman, and Oracle Data Modeler.

DDL code:

```
-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE zjy_corporate (
   corpid    INT NOT NULL AUTO_INCREMENT COMMENT 'ID of the corporate',
   name      VARCHAR(30) NOT NULL COMMENT 'Name of the corporate',
   regno     VARCHAR(20) NOT NULL COMMENT 'Registration number ',
   discount DECIMAL(4, 2) COMMENT 'discount of the corporate',
   PRIMARY KEY (corpid)
);
```

```
-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE zjy_corporate_customer (
   cid    INT NOT NULL COMMENT 'customer ID',
   empid  VARCHAR(30) NOT NULL COMMENT 'Employee ID of the customer who rents the car
on a corporate account',
   corpid INT NOT NULL,
   PRIMARY KEY (cid)
);


-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE zjy_coupon (
   couid    INT NOT NULL AUTO_INCREMENT COMMENT 'coupon ID',
   discount DECIMAL(4, 2) NOT NULL COMMENT 'percentage of discount',
   datefrom DATETIME NOT NULL COMMENT 'valid date from',
   dateto   DATETIME NOT NULL COMMENT 'valid date to',
   PRIMARY KEY (couid)
);


-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE zjy_customer (
   cid       INT NOT NULL AUTO_INCREMENT COMMENT 'customer ID',
   username  VARCHAR(30) NOT NULL COMMENT 'user name',
   pwd       VARCHAR(30) NOT NULL COMMENT 'password',
   sadress   VARCHAR(30) NOT NULL COMMENT 'street address',
   cadress   VARCHAR(30) NOT NULL COMMENT 'city address',
   staddress VARCHAR(30) NOT NULL COMMENT 'state address',
   zipcode   VARCHAR(10) NOT NULL COMMENT 'zip code',
   email     VARCHAR(30) NOT NULL COMMENT 'email',
   phone     VARCHAR(20) NOT NULL COMMENT 'phone number',
   ctype     CHAR(1) NOT NULL COMMENT 'customer type',
   PRIMARY KEY (cid)
);

ALTER TABLE zjy_customer
   ADD CONSTRAINT ch_inh_zjy_customer CHECK ( ctype IN ( 'C', 'I' ) );


-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE zjy_individual (
```

```
   cid   INT NOT NULL COMMENT 'customer ID',
   fname VARCHAR(30) NOT NULL COMMENT 'first name',
   lname VARCHAR(30) NOT NULL COMMENT 'last name',
   dln   VARCHAR(30) NOT NULL COMMENT ' Driver License Number',
   icn   VARCHAR(30) NOT NULL COMMENT 'Insurance Company Name',
   ipn   VARCHAR(30) NOT NULL COMMENT 'Insurance Policy Number',
   PRIMARY KEY (cid)
);


-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE zjy_invoice (
   invoiceid     INT NOT NULL AUTO_INCREMENT,
   invoicedate   DATETIME NOT NULL COMMENT 'Date of the invoice',
   invoiceamount DECIMAL(8, 2) NOT NULL COMMENT 'Amount of the invoice',
   vin           INT NOT NULL,
   cid           INT NOT NULL,
   pickdate      DATETIME NOT NULL,
   PRIMARY KEY (invoiceid)
);


-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE UNIQUE INDEX zjy_invoice__idx ON
   zjy_invoice (
       vin
   ASC,
       cid
   ASC,
       pickdate
   ASC );


-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE zjy_method (
   methodid TINYINT NOT NULL AUTO_INCREMENT COMMENT 'ID of the payment method',
   method   VARCHAR(10) NOT NULL COMMENT 'credit/debit/gift card',
   PRIMARY KEY (methodid)
);


-- SQLINES LICENSE FOR EVALUATION USE ONLY
```

```
CREATE TABLE zjy_office (
   oid       INT NOT NULL AUTO_INCREMENT COMMENT 'ID of the office location',
   saddress  VARCHAR(30) NOT NULL COMMENT 'street address',
   caddress  VARCHAR(30) NOT NULL COMMENT 'city address',
   staddress VARCHAR(30) NOT NULL COMMENT 'state address',
   zipcode   VARCHAR(10) NOT NULL COMMENT 'zip code',
   phone     VARCHAR(20) NOT NULL COMMENT 'phone number',
   PRIMARY KEY (oid)
);



-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE zjy_payment (
   paymentid   INT NOT NULL AUTO_INCREMENT COMMENT 'ID of the payment',
   paymentdate DATETIME NOT NULL COMMENT 'date of the payment',
   cardnumber  VARCHAR(20) NOT NULL COMMENT 'number of card for the payment',
   methodid    TINYINT NOT NULL,
   invoiceid   INT NOT NULL,
   PRIMARY KEY (paymentid)
);



-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE zjy_veh_cust (
   vin           INT NOT NULL,
   cid           INT NOT NULL,
   pickdate      DATETIME NOT NULL COMMENT 'Pickup Date',
   dropdate      DATETIME NOT NULL COMMENT 'Drop off Date',
   startodometer DECIMAL(8, 1) NOT NULL COMMENT 'Start Odometer',
   endodometer   DECIMAL(8, 1) NOT NULL COMMENT 'End Odometer',
   odometerlimit SMALLINT COMMENT 'Daily Odometer Limit',
   picklocid     INT NOT NULL,
   droplocid     INT NOT NULL,
   couid         INT
);

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE UNIQUE INDEX zjy_veh_cust__idxv1 ON
   zjy_veh_cust (
       couid
```

```
   ASC );

ALTER TABLE zjy_veh_cust
   ADD CONSTRAINT zjy_veh_cust_pk PRIMARY KEY ( vin,
                                                cid,
                                                pickdate );


-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE zjy_vehicle (
   vin    INT NOT NULL COMMENT 'Vehicle Identification Number',
   oid    INT NOT NULL,
   typeid SMALLINT NOT NULL,
   make   VARCHAR(30) NOT NULL,
   model  VARCHAR(30) NOT NULL,
   year   DATETIME NOT NULL,
   lpn    VARCHAR(30) NOT NULL COMMENT 'License Plate number'
);

ALTER TABLE zjy_vehicle ADD CONSTRAINT zjy_vehicle_pk PRIMARY KEY ( vin );


-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE zjy_veltype (
   typeid     SMALLINT NOT NULL AUTO_INCREMENT COMMENT 'Type ID of the vehicle',
   type       VARCHAR(30) NOT NULL COMMENT 'Type of the vehicle',
   rentalrate DECIMAL(7, 2) NOT NULL COMMENT 'Rental Rate per day',
   overfee    DECIMAL(7, 2) NOT NULL COMMENT 'fees for over mileage',
   PRIMARY KEY (typeid)
);



ALTER TABLE zjy_corporate_customer
   ADD CONSTRAINT corp_cust_corp_fk FOREIGN KEY ( corpid )
       REFERENCES zjy_corporate ( corpid );

ALTER TABLE zjy_corporate_customer
   ADD CONSTRAINT corp_cust_cust_fk FOREIGN KEY ( cid )
       REFERENCES zjy_customer ( cid );

ALTER TABLE zjy_individual
```

```
   ADD CONSTRAINT individual_zjy_customer_fk FOREIGN KEY ( cid )
       REFERENCES zjy_customer ( cid );

ALTER TABLE zjy_invoice
   ADD CONSTRAINT zjy_invoice_zjy_veh_cust_fk FOREIGN KEY ( vin,
                                                            cid,
                                                            pickdate )
       REFERENCES zjy_veh_cust ( vin,
                                 cid,
                                 pickdate );

ALTER TABLE zjy_payment
   ADD CONSTRAINT zjy_payment_zjy_invoice_fk FOREIGN KEY ( invoiceid )
       REFERENCES zjy_invoice ( invoiceid );

ALTER TABLE zjy_payment
   ADD CONSTRAINT zjy_payment_zjy_method_fk FOREIGN KEY ( methodid )
       REFERENCES zjy_method ( methodid );

ALTER TABLE zjy_veh_cust
   ADD CONSTRAINT zjy_veh_cust_zjy_coupon_fk FOREIGN KEY ( couid )
       REFERENCES zjy_coupon ( couid );

ALTER TABLE zjy_veh_cust
   ADD CONSTRAINT zjy_veh_cust_zjy_customer_fk FOREIGN KEY ( cid )
       REFERENCES zjy_customer ( cid );

ALTER TABLE zjy_veh_cust
   ADD CONSTRAINT zjy_veh_cust_zjy_office_fk FOREIGN KEY ( picklocid )
       REFERENCES zjy_office ( oid );

ALTER TABLE zjy_veh_cust
   ADD CONSTRAINT zjy_veh_cust_zjy_office_fkv2 FOREIGN KEY ( droplocid )
       REFERENCES zjy_office ( oid );

ALTER TABLE zjy_veh_cust
   ADD CONSTRAINT zjy_veh_cust_zjy_vehicle_fk FOREIGN KEY ( vin )
       REFERENCES zjy_vehicle ( vin );

ALTER TABLE zjy_vehicle
   ADD CONSTRAINT zjy_vehicle_zjy_office_fk FOREIGN KEY ( oid )
       REFERENCES zjy_office ( oid );
```

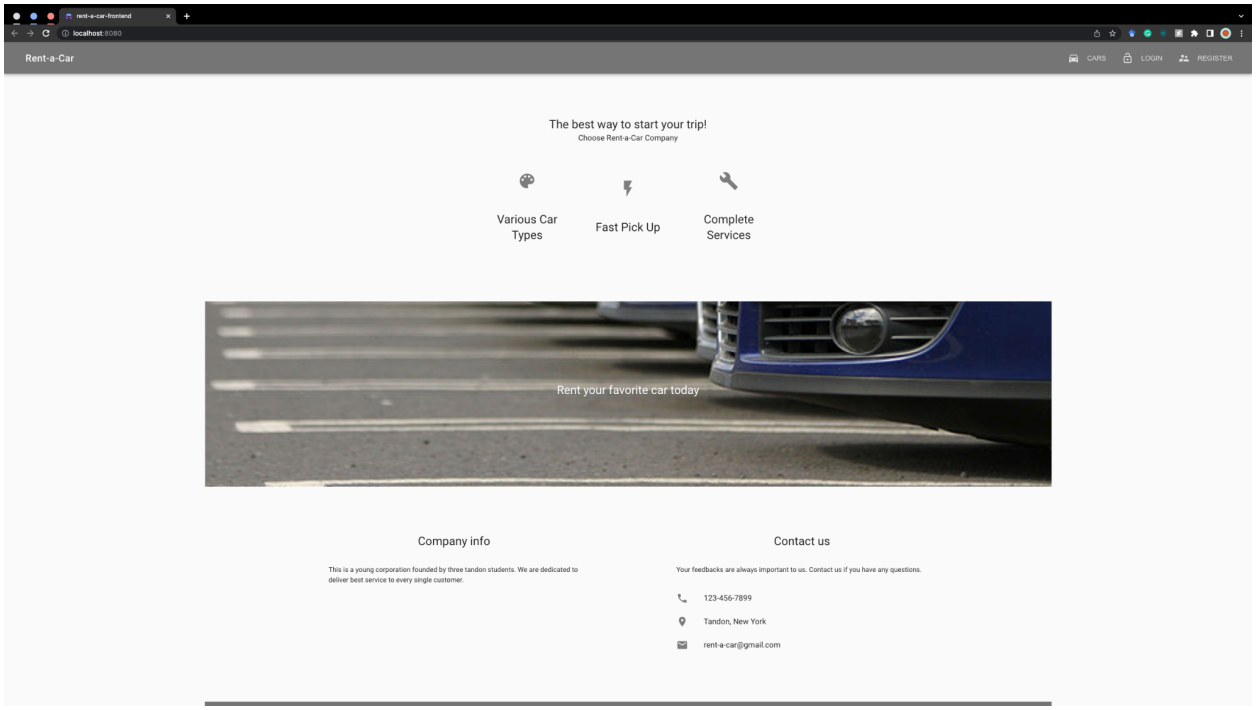
```
ALTER TABLE zjy_vehicle
   ADD CONSTRAINT zjy_vehicle_zjy_type_fk FOREIGN KEY ( typeid )
       REFERENCES zjy_veltype ( typeid );
```

Tables and records:

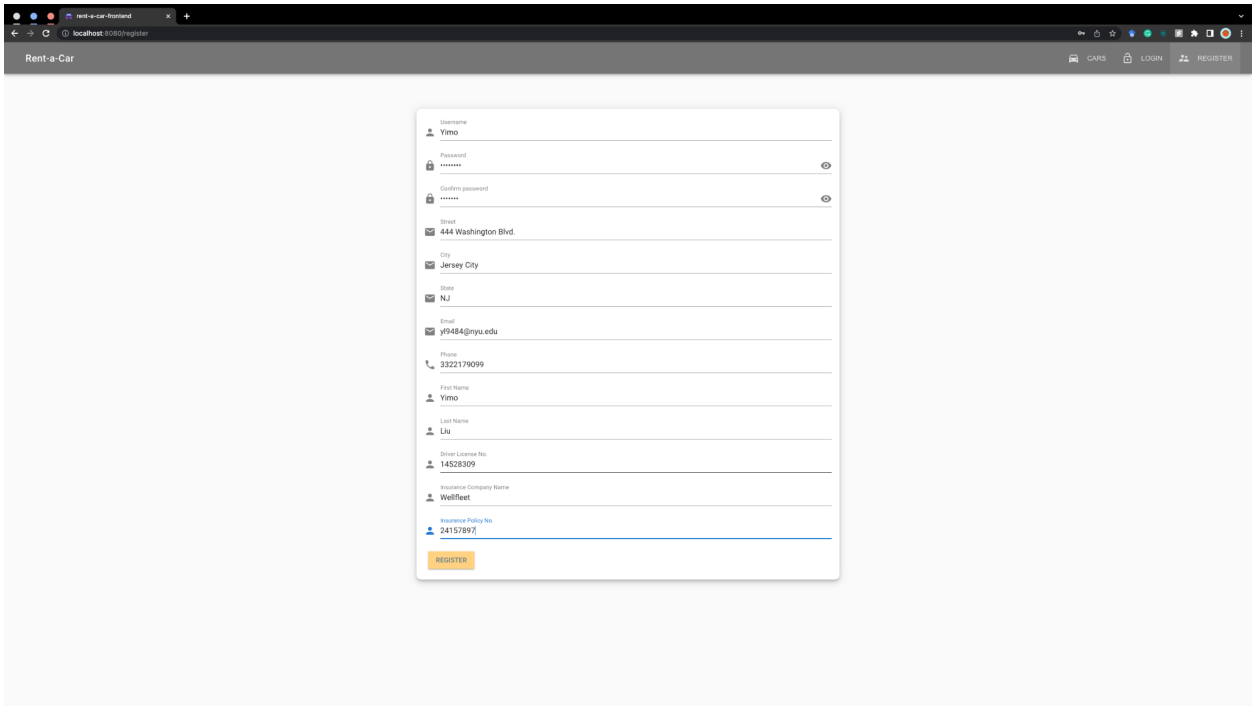
- ZJY_OFFICE: No. of records 10
- ZJY_VEHICLE: No. of records 15
- ZJY_VELTYPE: No. of records 14
- ZJY_CUSTOMER: No. of records 15
- ZJY_INDIVIDUAL: No. of records 10
- ZJY_CORPORATE_CUSTOMER: No. of records 5
- ZJY_CORPORATE: No. of records 5
- ZJY_COUPON: No. of records 13
- ZJY_VEH_CUST: No. of records 15
- ZJY_INVOICE: No. of records 15
- ZJY_PAYMENT: No. of records 15
- ZJY_METHOD: No. of records 3

Web Application Screenshots:

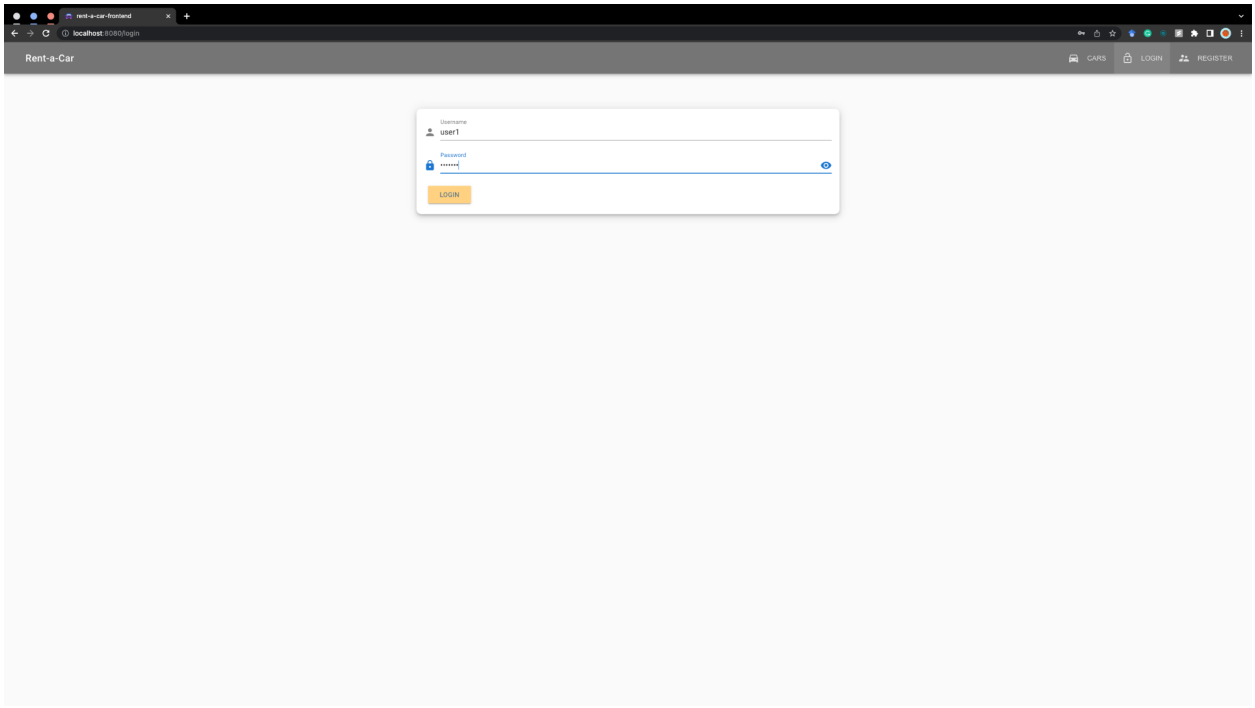
Main page:



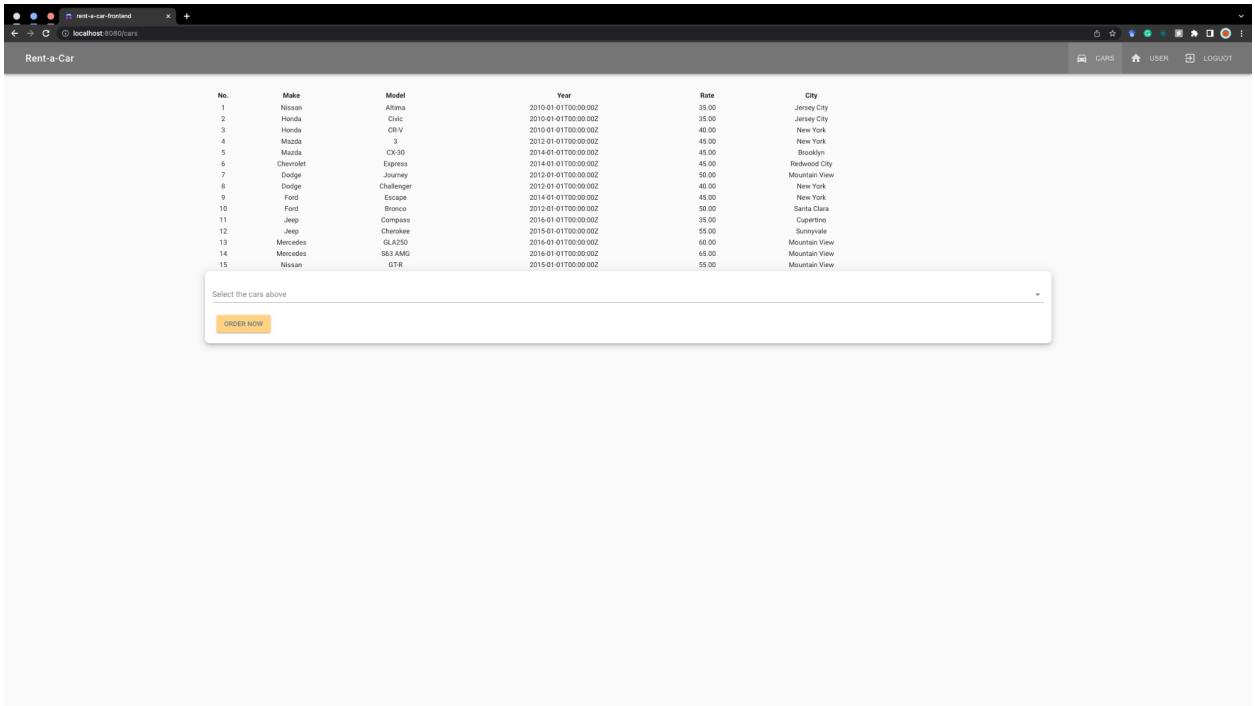
Register page:



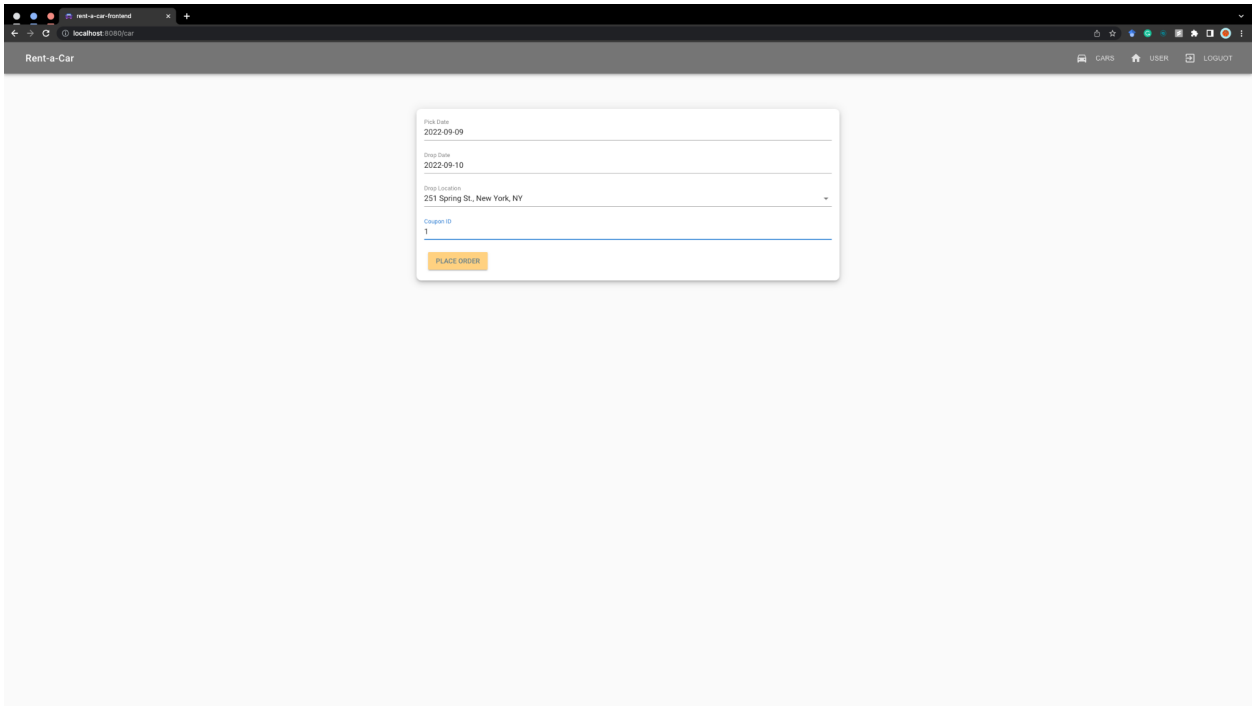
Login page:



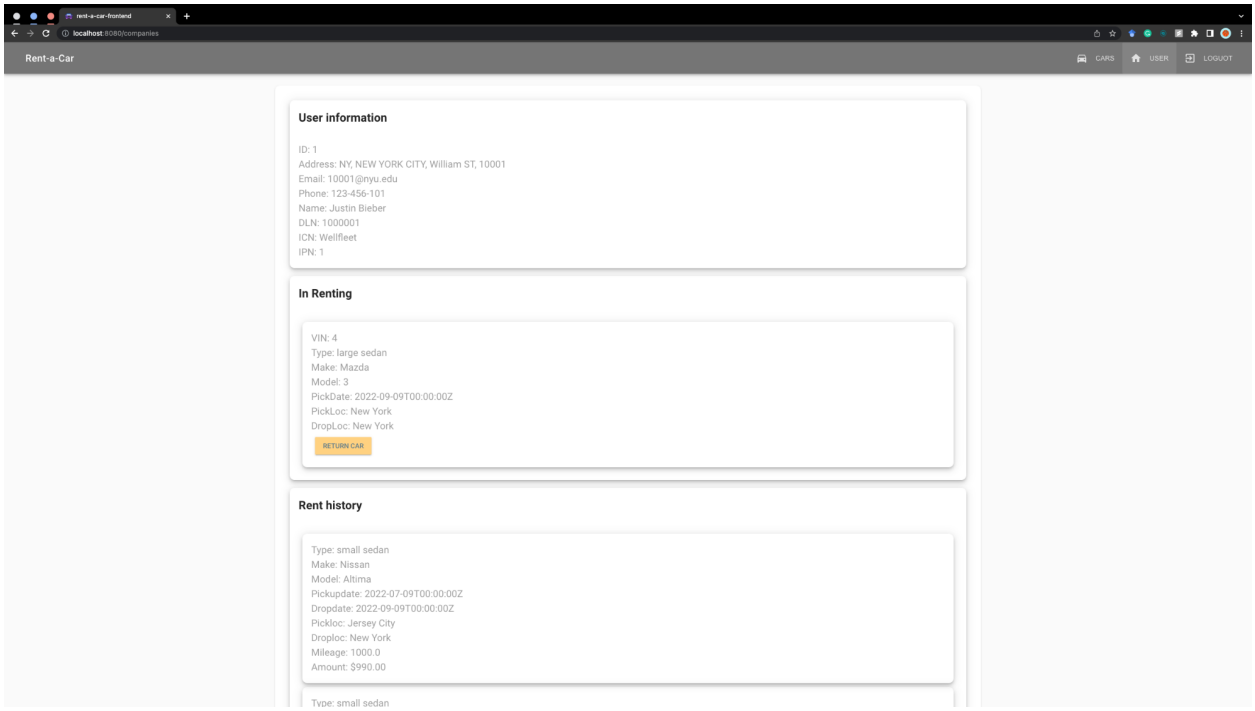
Cars page:



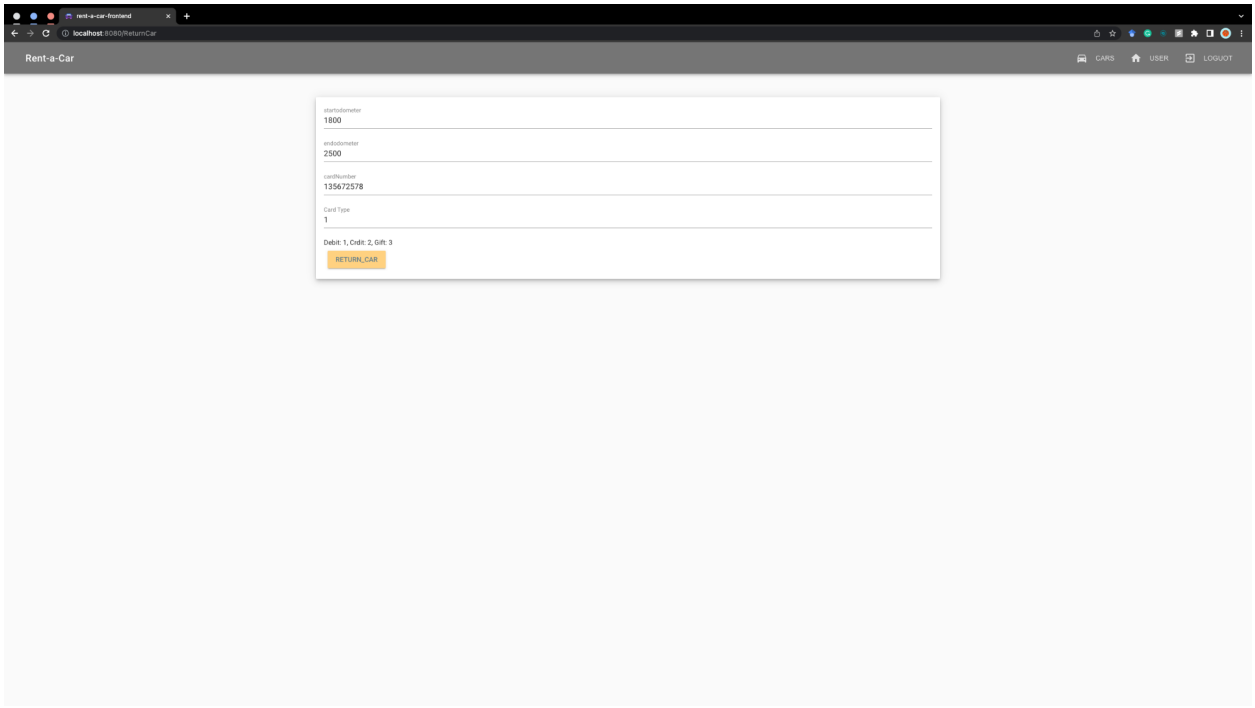
Place order:



User page:



Return car:



Security features:

SQL injection is prevented by the ORM(Object-relational mapping) provided Django. With the ORM, we don’t have SQL strings in our code. Various functions and APIs are used to access and manipulate data. The ORM will check any input parameters before executing it, so any potential SQL injections will be blocked.

Reflections:

In our work with other teammates, we learned the spirit of cooperation and the whole system of hosting a database in web environment. We learned the database design will affect the performance and scalability of the overall system profoundly. Before working on the project, we had a very weak foundation on the front end web design. But through our hard work and learning, we have the chance to challenge our ability and successfully present our work.

6 SQL Query Results:

```
-- 1. Table joins with at least 3 tables in join

SELECT vin, make, model, saddress || ' ' || CAddress || ' ' || ZipCode AS "Office address"

FROM ZJY_OFFICE NATURAL JOIN ZJY_VEHICLE NATURAL JOIN ZJY_VELTYPE

ORDER BY vin;

-- list the detail of the vehicles and the office they belongs to
```

VIN	MAKE	MODEL	Office address
1	Nissan	Altima	444 Washington Blvd. Jersey City 07310
2	Honda	Civic	444 Washington Blvd. Jersey City 07310
3	Honda	CR-V	251 Spring St. New York 10013
4	Mazda	3	251 Spring St. New York 10013
5	Mazda	CX-30	6 MetroTech Center Brooklyn 11201
6	Chevrolet	Express	209 Redwood Shores Pkwy Redwood City 94065
7	Dodge	Journey	1911 Landings Dr Mountain View 94043
8	Dodge	Challenger	251 Spring St. New York 10013
9	Ford	Escape	251 Spring St. New York 10013
10	Ford	Bronco	4120 Network Cir Santa Clara 95054
11	Jeep	Compass	One Apple Park Way Cupertino 95014
12	Jeep	Cherokee	900 W Maude Ave Sunnyvale 94085
13	Mercedes	GLA250	250 Bryant St Mountain View 94041
14	Mercedes	S63 AMG	250 Bryant St Mountain View 94041
15	Nissan	GT-R	250 Bryant St Mountain View 94041

Download CSV

15 rows selected.

```
-- 2. Multi-row subquery

SELECT typeName, vin, make, model, year

FROM ZJY_VELTYPE JOIN

(

   SELECT typeID, vin, make, model, year

   FROM ZJY_VEHICLE a

   WHERE year >= ALL

   (
```

```
        SELECT year
        FROM ZJY_VEHICLE b
        WHERE a.typeID = b.typeID
    )
    ORDER BY typeID
) USING (typeID)
ORDER BY TYPEID;
```

-- list the vehicle in each type that has the newest year

TYPENAME	VIN	MAKE	MODEL	YEAR
small sedan	1	Nissan	Altima	01-JAN-10
small sedan	2	Honda	Civic	01-JAN-10
middle-size sedan	3	Honda	CR-V	01-JAN-10
large sedan	6	Chevrolet	Express	01-MAY-14
large sedan	9	Ford	Escape	01-MAY-14
large sedan	5	Mazda	CX-30	01-MAY-14
luxury sedan	7	Dodge	Journey	01-MAY-12
suv	8	Dodge	Challenger	01-MAY-12
wagon	10	Ford	Bronco	01-MAY-12
hatchback	11	Jeep	Compass	01-MAY-16
convertible	12	Jeep	Cherokee	01-MAY-15
sports car	13	Mercedes	GLA250	01-MAY-16
supercar	14	Mercedes	S63 AMG	01-MAY-16
mini van	15	Nissan	GT-R	01-MAY-15

Download CSV

14 rows selected.

```
-- 3. Correlated subquery
SELECT typeName,
(
    SELECT count(*)
    FROM ZJY_VEHICLE b
    WHERE a.typeID = b.typeId
) as cnt
FROM ZJY_VELTYPE a
ORDER BY cnt;
-- list the number of vehicles of all type
```

TYPENAME	CNT
luxury suv	0
pickup truck	0
coupe	0
luxury sedan	1
supercar	1
suv	1
mini van	1
wagon	1
hatchback	1
middle-size sedan	1
convertible	1
sports car	1
small sedan	2
large sedan	4

Download CSV

14 rows selected.

-- 4. SET operator query

SELECT vin, cid, pickdate, discount

FROM ZJY_VEH_CUST NATURAL JOIN ZJY_COUPON

UNION

SELECT vin, cid, pickdate, discount

FROM ZJY_VEH_CUST NATURAL JOIN ZJY_CUSTOMER NATURAL JOIN ZJY_CORPORATE_CUSTOMER NATURAL JOIN ZJY_CORPORATE

WHERE discount IS NOT NULL

ORDER BY pickdate;

-- list all the orders that uses a discount, either from a coupon or being a corporate customer with discount

VIN	CID	PICKDATE	DISCOUNT
7	8	07-APR-20	5
11	5	01-JUL-20	5
12	13	01-JAN-21	5
12	13	01-JAN-21	6
10	12	24-MAY-21	5
10	12	24-MAY-21	13
15	15	01-JUN-21	5.6
15	15	01-JUN-21	10
7	5	09-NOV-21	5
4	7	08-JAN-22	10
14	10	01-FEB-22	3
1	1	09-JUL-22	5
1	2	07-SEP-22	6

Download CSV

13 rows selected.

-- 5. Query with in line view or WITH clause

```
WITH maxType AS
(
    SELECT typeID, typeName, RentalRate
    FROM ZJY_VELTYPE
    WHERE RentalRate =
```

```
    (
        SELECT max(RentalRate)
        FROM ZJY_VELTYPE
    )
)
SELECT typeName, RentalRate, vin, make, model, year
FROM ZJY_VEHICLE JOIN maxType USING (typeID);
-- List the vehicle with the type of the highest rental rate
```

TYPENAME	RENTALRATE	VIN	MAKE	MODEL	YEAR
supercar	65	14	Mercedes	S63 AMG	01-MAY-16

Download CSV

```
-- 6. TOP-N query
SELECT *
FROM
(
    SELECT vin, cid, pickdate, discount, rank() over (order by discount DESC) AS myrank
    FROM
    (
        SELECT vin, cid, pickdate, discount
        FROM ZJY_VEH_CUST NATURAL JOIN ZJY_COUPON
        UNION
        SELECT vin, cid, pickdate, discount
        FROM ZJY_VEH_CUST NATURAL JOIN ZJY_CUSTOMER NATURAL JOIN
ZJY_CORPORATE_CUSTOMER NATURAL JOIN ZJY_CORPORATE
        WHERE discount IS NOT NULL
```

```
        ORDER BY pickdate
    )
    ORDER BY discount DESC
)
WHERE myrank <= 2;

-- list the order which are the top 2 in discount rate
```

VIN	CID	PICKDATE	DISCOUNT	MYRANK
10	12	24-MAY-21	13	1
15	15	01-JUN-21	10	2
4	7	08-JAN-22	10	2

Download CSV

3 rows selected.