Final Project

Student ID: ct2673, kz2193, yl8651

Name: Chuyun Tan, Kexin Zhou, Yihan Liu

Course: Principles of Database Systems

Section Number: CS-GY 6083-B, Spring 2022

Date: 05/11/2022

Table of Content

1. Summery 3
2. Tools 5
3. DDL Code 5
4. Tables and Records 16
5. Web Application Screenshots 21
   - Home Page(Login) 21
   - Individual Signup 21
   - Corporate Signup: 23
   - Service Options(For Customers) 24
   - Rent A Car 24
   - Rental History of Customers: 26
   - Officer Options: 27
   - Return A Car: 27
   - Return Successfully 28
   - Generate Invoice 28
   - Multi-card Payment 29
   - Payment failed 29
   - Add New Vehicles 30
   - Log Out 30
6. Security Features 30
7. Learning 31
8. Business Analysis 32
   - Q1 Three Table Join 32
   - Q2 Multi-row Subquery 32
   - Q3 Correlated Subquery 32
   - Q4 SET operator query 33
   - Q5 WITH clause 34
   - Q6 TOP-N query 34
9. Extra Features 35
   - Index 35

1. Summery

WOW (World On Wheels) is a car rental company with offices in several cities offering car rental services. Each office has multiple rental vehicles of various classes. Our system could help WOW record large-scale data in database neatly.

The home page of this website is the login and registration page. Customers could choose whether to sign up for a personal account or a company account. All their registration information would be saved in the table CKY_CUSTOMER.

With this rental vehicle system, customers could search rental information, reserve vehicles, and review their past rental service online quickly.

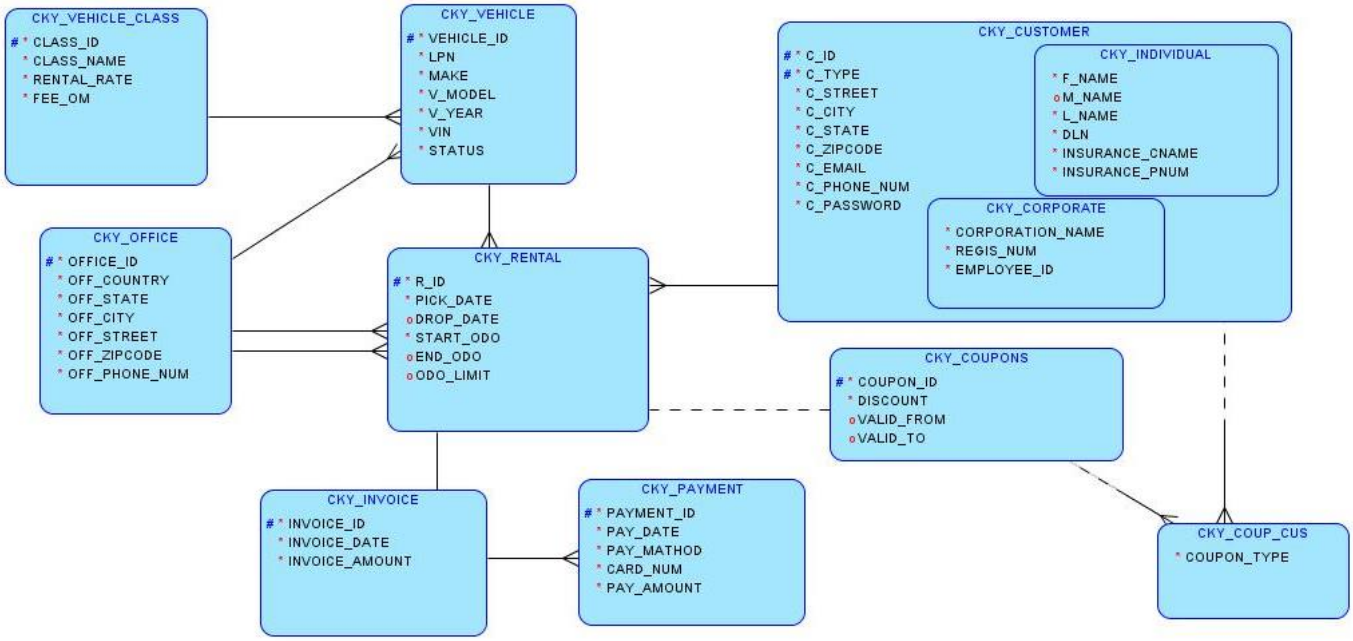
When clicking "Rent A Car" button, customers can jump to the car rental page. In this page, they need to choose an office to pick up a car, a vehicle class, a specific car which is available in this office, and the dates they want to pick up and return on. All these information will save in the table CKY_RENTAL with their CUSTOMER_ID. Moreover, customers can also view their historical rental records and bills' detail on this website.

Also, this system release burden of employee's work in WOW. Employee can use this website to complete rental service and view report of clients.

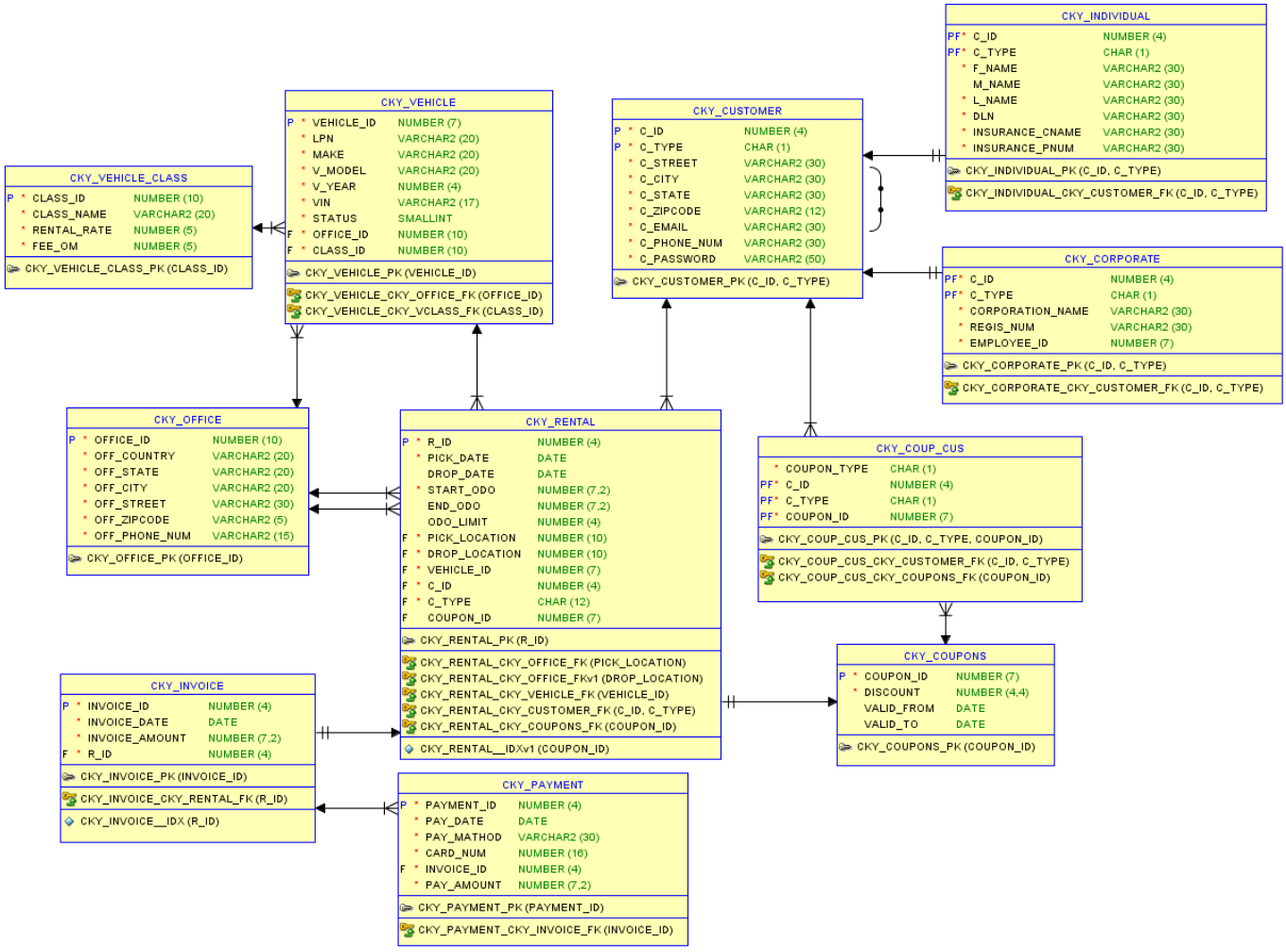
When customers want to return the car in an office, the employee can enter their RENTAL_ID to find the rental service. Then, employee need to choose the office where the customer wants to drop off at, and input odometer limitation of the returning car as well as end odometer. Lastly, the employee can choose the coupon if customers want to use from their account. After that, the invoice of this rental will be generated and displayed for customer to confirm. Then, the employee can enter the payment information that provided by customer to complete the payment. Also, multiple payment can be added to one invoice, and the sum of the payment amounts needs to correspond to the total amount.

This system could also calculate many kinds of statistics and report of clients, e.g., top sale office, popular vehicles, and high consumption customers. With this information, WOW could make wiser decisions and improvement in the future.

Logical Model:



Relation Model:



2. Tools

Database: MySQL
Front-end: Vue, Ajax, HTML, CSS, JavaScript
Back-end: Django, Python

3. DDL Code

```
CREATE TABLE cky_corporate (
    c_id                NUMBER(4) NOT NULL,
    c_type              CHAR(1) NOT NULL,
    corporation_name VARCHAR2(30) NOT NULL,
    regis_num           VARCHAR2(30) NOT NULL,
    employee_id         NUMBER(7) NOT NULL
);

COMMENT ON COLUMN cky_corporate.c_id IS
    'ID of the customer.';

COMMENT ON COLUMN cky_corporate.c_type IS
    'Type of the customer.';

COMMENT ON COLUMN cky_corporate.corporation_name IS
    'Name of the corporation.';

COMMENT ON COLUMN cky_corporate.regis_num IS
    'Registration number';

COMMENT ON COLUMN cky_corporate.employee_id IS
    'Employee ID ';

ALTER TABLE cky_corporate ADD CONSTRAINT cky_corporate_pk PRIMARY KEY
( c_id,
                                                                c_type );

CREATE TABLE cky_coup_cus (
    coupon_type CHAR(1) NOT NULL,
    c_id            NUMBER(4) NOT NULL,
    c_type          CHAR(1) NOT NULL,
    coupon_id     NUMBER(7) NOT NULL
);

COMMENT ON COLUMN cky_coup_cus.coupon_type IS
```

```
    'Type of the coupon.';

ALTER TABLE cky_coup_cus
    ADD CONSTRAINT cky_coup_cus_pk PRIMARY KEY ( c_id,
                                                 c_type,
                                                 coupon_id );

CREATE TABLE cky_coupons (
    coupon_id    NUMBER(7) NOT NULL,
    discount     NUMBER(4, 4) NOT NULL,
    valid_from DATE,
    valid_to     DATE
);

COMMENT ON COLUMN cky_coupons.coupon_id IS
    'ID of discount coupon.';

COMMENT ON COLUMN cky_coupons.discount IS
    'Percentage of discount.';

COMMENT ON COLUMN cky_coupons.valid_from IS
    'Start date of the discount coupon.';

COMMENT ON COLUMN cky_coupons.valid_to IS
    'End date of the discount coupon.';

ALTER TABLE cky_coupons ADD CONSTRAINT cky_coupons_pk PRIMARY KEY
( coupon_id );

CREATE TABLE cky_customer (
    c_id            NUMBER(4) NOT NULL,
    c_type          CHAR(1) NOT NULL,
    c_street        VARCHAR2(30) NOT NULL,
    c_city          VARCHAR2(30) NOT NULL,
    c_state         VARCHAR2(30) NOT NULL,
    c_zipcode       VARCHAR2(12) NOT NULL,
    c_email         VARCHAR2(30) NOT NULL,
    c_phone_num VARCHAR2(30) NOT NULL,
    c_password    VARCHAR2(50) NOT NULL
);

ALTER TABLE cky_customer
    ADD CONSTRAINT ch_inh_cky_customer CHECK ( c_type IN ( 'C', 'I' ) );
```

```
COMMENT ON COLUMN cky_customer.c_id IS
    'ID of the customer.';

COMMENT ON COLUMN cky_customer.c_type IS
    'Type of the customer.';

COMMENT ON COLUMN cky_customer.c_street IS
    'Street in customer''s address.';

COMMENT ON COLUMN cky_customer.c_city IS
    'City of customer''s address.';

COMMENT ON COLUMN cky_customer.c_state IS
    'State of customer''s address.';

COMMENT ON COLUMN cky_customer.c_zipcode IS
    'Zipcode of customer''s address.';

COMMENT ON COLUMN cky_customer.c_email IS
    'Email address of the customer.';

COMMENT ON COLUMN cky_customer.c_phone_num IS
    'Phone number of the customer.';

COMMENT ON COLUMN cky_customer.c_password IS
    'Customer Password.';

ALTER TABLE cky_customer ADD CONSTRAINT cky_customer_pk PRIMARY KEY
( c_id,
                                                                c_type );

CREATE TABLE cky_individual (
    c_id                NUMBER(4) NOT NULL,
    c_type              CHAR(1) NOT NULL,
    f_name               VARCHAR2(30) NOT NULL,
    m_name               VARCHAR2(30),
    l_name               VARCHAR2(30) NOT NULL,
    dln                 VARCHAR2(30) NOT NULL,
    insurance_cname VARCHAR2(30) NOT NULL,
    insurance_pnum   VARCHAR2(30) NOT NULL
);

COMMENT ON COLUMN cky_individual.c_id IS
    'ID of the customer.';
```

```
COMMENT ON COLUMN cky_individual.c_type IS
    'Type of the customer.';

COMMENT ON COLUMN cky_individual.f_name IS
    'First name of the customer.';

COMMENT ON COLUMN cky_individual.m_name IS
    'Middle name of the customer.';

COMMENT ON COLUMN cky_individual.l_name IS
    'Last name of the customer.';

COMMENT ON COLUMN cky_individual.dln IS
    'Driver License Number';

COMMENT ON COLUMN cky_individual.insurance_cname IS
    'Insurance Company Name ';

COMMENT ON COLUMN cky_individual.insurance_pnum IS
    'Insurance Policy Number';

ALTER TABLE cky_individual ADD CONSTRAINT cky_individual_pk PRIMARY KEY
( c_id,
                                                                    c_type );

CREATE TABLE cky_invoice (
    invoice_id        NUMBER(4) NOT NULL,
    invoice_date      DATE NOT NULL,
    invoice_amount NUMBER(7, 2) NOT NULL,
    r_id                 NUMBER(4) NOT NULL
);

COMMENT ON COLUMN cky_invoice.invoice_id IS
    'unique number of invoice';

COMMENT ON COLUMN cky_invoice.invoice_date IS
    'Invoice Date';

COMMENT ON COLUMN cky_invoice.invoice_amount IS
    'Invoice Amount';

CREATE UNIQUE INDEX cky_invoice__idx ON
    cky_invoice (
```

```
        r_id
    ASC );

ALTER TABLE cky_invoice ADD CONSTRAINT cky_invoice_pk PRIMARY KEY
( invoice_id );

CREATE TABLE cky_office (
    office_id       NUMBER(10) NOT NULL,
    off_country     VARCHAR2(20) NOT NULL,
    off_state       VARCHAR2(20) NOT NULL,
    off_city        VARCHAR2(20) NOT NULL,
    off_street      VARCHAR2(30) NOT NULL,
    off_zipcode     VARCHAR2(5) NOT NULL,
    off_phone_num VARCHAR2(15) NOT NULL
);

COMMENT ON COLUMN cky_office.office_id IS
    'ID of office in different location';

COMMENT ON COLUMN cky_office.off_country IS
    'country of the office';

COMMENT ON COLUMN cky_office.off_state IS
    'STATE OF OFFICE';

COMMENT ON COLUMN cky_office.off_city IS
    'City of office';

COMMENT ON COLUMN cky_office.off_street IS
    'street address of office';

COMMENT ON COLUMN cky_office.off_zipcode IS
    'ZIP CODE OF THE OFFICE';

COMMENT ON COLUMN cky_office.off_phone_num IS
    'phone number of office';

ALTER TABLE cky_office ADD CONSTRAINT cky_office_pk PRIMARY KEY ( office_id );

CREATE TABLE cky_payment (
    payment_id NUMBER(4) NOT NULL,
    pay_date     DATE NOT NULL,
    pay_mathod VARCHAR2(30) NOT NULL,
    card_num     NUMBER(16) NOT NULL,
```

```
    invoice_id NUMBER(4) NOT NULL,
    pay_amount NUMBER(7, 2) NOT NULL
);

COMMENT ON COLUMN cky_payment.payment_id IS
    'unique number of payment';

COMMENT ON COLUMN cky_payment.pay_date IS
    'payment date';

COMMENT ON COLUMN cky_payment.pay_mathod IS
    'payment method';

COMMENT ON COLUMN cky_payment.card_num IS
    'card number of payment';

COMMENT ON COLUMN cky_payment.pay_amount IS
    'PAYMENT AMOUNT';

ALTER TABLE cky_payment ADD CONSTRAINT cky_payment_pk PRIMARY KEY
( payment_id );

CREATE TABLE cky_rental (
    r_id            NUMBER(4) NOT NULL,
    pick_date       DATE NOT NULL,
    drop_date       DATE,
    start_odo       NUMBER(7, 2) NOT NULL,
    end_odo         NUMBER(7, 2),
    odo_limit       NUMBER(4),
    pick_location NUMBER(10) NOT NULL,
    drop_location NUMBER(10) NOT NULL,
    vehicle_id      NUMBER(7) NOT NULL,
    c_id            NUMBER(4) NOT NULL,
    c_type          CHAR(12) NOT NULL,
    coupon_id       NUMBER(7) NOT NULL
);

COMMENT ON COLUMN cky_rental.r_id IS
    'Unique number of each rental service';

COMMENT ON COLUMN cky_rental.pick_date IS
    'Pick up Date of rental service';

COMMENT ON COLUMN cky_rental.drop_date IS
```

```
    'Drop off date of rental service';

COMMENT ON COLUMN cky_rental.start_odo IS
    'Start odometer of the car';

COMMENT ON COLUMN cky_rental.end_odo IS
    'End odometer of the car';

COMMENT ON COLUMN cky_rental.odo_limit IS
    'Daily odometer limit of the car';

CREATE UNIQUE INDEX cky_rental__idxv1 ON
    cky_rental (
        coupon_id
    ASC );

ALTER TABLE cky_rental ADD CONSTRAINT cky_rental_pk PRIMARY KEY ( r_id );

CREATE TABLE cky_vehicle (
    vehicle_id NUMBER(7) NOT NULL,
    lpn          VARCHAR2(20) NOT NULL,
    make         VARCHAR2(20) NOT NULL,
    v_model      VARCHAR2(20) NOT NULL,
    v_year       NUMBER(4) NOT NULL,
    vin          VARCHAR2(17) NOT NULL,
    status       SMALLINT NOT NULL,
    office_id    NUMBER(10) NOT NULL,
    class_id     NUMBER(10) NOT NULL
);

COMMENT ON COLUMN cky_vehicle.vehicle_id IS
    'Vehicle ID';

COMMENT ON COLUMN cky_vehicle.lpn IS
    'License plate number to identify vehicle';

COMMENT ON COLUMN cky_vehicle.make IS
    'MAKE of a vehicle';

COMMENT ON COLUMN cky_vehicle.v_model IS
    'MODEL of a vehicle';

COMMENT ON COLUMN cky_vehicle.v_year IS
    'Year that produce the vehicle';
```

```
COMMENT ON COLUMN cky_vehicle.vin IS
    'Vehicle Identification Number';

COMMENT ON COLUMN cky_vehicle.status IS
    'If the vehicle is rented, status is 1, or status is 0.';

ALTER TABLE cky_vehicle ADD CONSTRAINT cky_vehicle_pk PRIMARY KEY
( vehicle_id );

CREATE TABLE cky_vehicle_class (
    class_id       NUMBER(10) NOT NULL,
    class_name     VARCHAR2(20) NOT NULL,
    rental_rate NUMBER(5) NOT NULL,
    fee_om          NUMBER(5) NOT NULL
);

COMMENT ON COLUMN cky_vehicle_class.class_id IS
    'Class of vehicle';

COMMENT ON COLUMN cky_vehicle_class.class_name IS
    'CLASS NAME OF VEHICLE';

COMMENT ON COLUMN cky_vehicle_class.rental_rate IS
    'Rental rate of different class($/day)';

COMMENT ON COLUMN cky_vehicle_class.fee_om IS
    'FEE of over over mileage';

ALTER TABLE cky_vehicle_class ADD CONSTRAINT cky_vehicle_class_pk PRIMARY
KEY ( class_id );

ALTER TABLE cky_corporate
    ADD CONSTRAINT cky_corporate_cky_customer_fk FOREIGN KEY ( c_id,
                                                               c_type )
        REFERENCES cky_customer ( c_id,
                                  c_type );

ALTER TABLE cky_coup_cus
    ADD CONSTRAINT cky_coup_cus_cky_coupons_fk FOREIGN KEY ( coupon_id )
        REFERENCES cky_coupons ( coupon_id );

ALTER TABLE cky_coup_cus
    ADD CONSTRAINT cky_coup_cus_cky_customer_fk FOREIGN KEY ( c_id,
```

```
                                                                c_type )
        REFERENCES cky_customer ( c_id,
                                  c_type );

ALTER TABLE cky_individual
    ADD CONSTRAINT cky_individual_cky_customer_fk FOREIGN KEY ( c_id,
                                                                c_type )
        REFERENCES cky_customer ( c_id,
                                  c_type );

ALTER TABLE cky_invoice
    ADD CONSTRAINT cky_invoice_cky_rental_fk FOREIGN KEY ( r_id )
        REFERENCES cky_rental ( r_id );

ALTER TABLE cky_payment
    ADD CONSTRAINT cky_payment_cky_invoice_fk FOREIGN KEY ( invoice_id )
        REFERENCES cky_invoice ( invoice_id );

ALTER TABLE cky_rental
    ADD CONSTRAINT cky_rental_cky_coupons_fk FOREIGN KEY ( coupon_id )
        REFERENCES cky_coupons ( coupon_id );

ALTER TABLE cky_rental
    ADD CONSTRAINT cky_rental_cky_customer_fk FOREIGN KEY ( c_id,
                                                            c_type )
        REFERENCES cky_customer ( c_id,
                                  c_type );

ALTER TABLE cky_rental
    ADD CONSTRAINT cky_rental_cky_office_fk FOREIGN KEY ( pick_location )
        REFERENCES cky_office ( office_id );

ALTER TABLE cky_rental
    ADD CONSTRAINT cky_rental_cky_office_fkv1 FOREIGN KEY ( drop_location )
        REFERENCES cky_office ( office_id );

ALTER TABLE cky_rental
    ADD CONSTRAINT cky_rental_cky_vehicle_fk FOREIGN KEY ( vehicle_id )
        REFERENCES cky_vehicle ( vehicle_id );

ALTER TABLE cky_vehicle
    ADD CONSTRAINT cky_vehicle_cky_office_fk FOREIGN KEY ( office_id )
        REFERENCES cky_office ( office_id );
```

```
ALTER TABLE cky_vehicle
    ADD CONSTRAINT cky_vehicle_cky_vclass_fk FOREIGN KEY ( class_id )
        REFERENCES cky_vehicle_class ( class_id );

CREATE OR REPLACE TRIGGER arc_fkarc_8_cky_individual BEFORE
    INSERT OR UPDATE OF c_id, c_type ON cky_individual
    FOR EACH ROW
DECLARE
    d CHAR(1);
BEGIN
    SELECT
        a.c_type
    INTO d
    FROM
        cky_customer a
    WHERE
        a.c_id = :new.c_id
        AND a.c_type = :new.c_type;

    IF ( d IS NULL OR d <> 'I' ) THEN
        raise_application_error(
                                    -20223,
                                    'FK CKY_INDIVIDUAL_CKY_CUSTOMER_FK in Table CKY_INDIVIDUAL violates Arc constraint on Table CKY_CUSTOMER - discriminator column C_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_8_cky_corporate BEFORE
    INSERT OR UPDATE OF c_id, c_type ON cky_corporate
    FOR EACH ROW
DECLARE
    d CHAR(1);
BEGIN
    SELECT
        a.c_type
```

```
    INTO d
    FROM
        cky_customer a
    WHERE
        a.c_id = :new.c_id
        AND a.c_type = :new.c_type;

    IF ( d IS NULL OR d <> 'C' ) THEN
        raise_application_error(
                                -20223,
                                'FK CKY_CORPORATE_CKY_CUSTOMER_FK in Table CKY_CORPORATE violates Arc constraint on Table CKY_CUSTOMER - discriminator column C_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 概要报告:
--
-- CREATE TABLE                            11
-- CREATE INDEX                             2
-- ALTER TABLE                             25
-- 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

4. Tables and Records

TABLE LIST:

Tables_in_cky_scheme (cky%)
cky_corporate
cky_coup_cust
cky_coupons
cky_customer
cky_individual
cky_invoice
cky_office
cky_payment
cky_rental
cky_vehicle
cky_vehicle_class

CKY_OFFICE:

OFFICE_ID	OFF_COUNTRY	OFF_STATE	OFF_CITY	OFF_STREET	OFF_ZIPCODE	OFF_PHONE_NUM
1	US	NJ	Jersey City	605 Pavonia Ave	07306	6464567890
2	US	NJ	Jersey City	112 Mangolia Ave	07306	6463334444
3	US	NJ	Jersey City	20 Newark Ave	07304	3248889999
4	US	NY	New York	46 Bowery	10013	6462121212
5	US	NY	New York	26 Forsyth St	10002	6463232323
6	US	NY	Brooklyn	57 Orange St	11201	6464334443
7	US	NY	Queens	40th Ave	11101	3249229991
8	US	MA	Dorchester	500 Geneva Ave	02122	6767667776
9	US	MA	Boston	206 Washington St	02109	7788778887
10	US	MA	Boston	1 Charles St	02116	9900909009
11	US	NJ	Princeton	201 Goheen Walk	08544	6667778888
12	US	MA	Cambridge	1 Oxford St	02138	7773330000

CKY_VEHICLE_CLASS:

CLASS_ID	CLASS_NAME	RENTAL_RAT	FEE_OM
1	small car	20	2
2	mid-size car	30	3
3	luxury car	80	5
4	SUV	50	4
5	Mini Van	30	3
6	Station Wagon	40	3
7	Hatchback	40	4
8	sports car	70	5
9	compact car	20	2
10	sedan	25	2
11	motorcycle	10	1

CKY_VEHICLE:

VEHICLE_ID	LPN	MAKE	V_MODEL	V_YEAR	VIN	STATUS	CLASS_ID	OFFICE_ID
1	B42LBJ	TELSA	S	2013	5YJSA1DG9DFP14705	0	7	7
2	FM74H	SUBARU	Legacy	2011	4S3BMHB68B3286050	0	10	1
3	PHY81C	SUBARU	Legacy	2012	4F3BMHB68B3285060	0	10	2
4	HXW7509	FORD	Expedition	2015	1FMJU2AT3FEF00187	0	4	2
5	ABC1234	CHEVROLET	Blazer	1991	1GNCS18Z3M0115561	1	3	3
6	EEV2967	ACURA	RDX	2015	5J8TB4H38FL002262	1	2	3
7	215BG2	FORD	F-150	2011	1FTFW1R6XBFB08616	0	5	8
8	99NK92	BMW	i8	2019	WBY2Z4C54KVB81799	0	1	7
9	YXP645	HONDA	CR-V	2004	JHLRD77874C026456	1	9	6
10	SZT144	INTERNATIONAL	MA015	2007	1HTMPAFM37H414790	0	6	10
11	SZT145	TOYOTA	Tundra	2017	5TFAW5F12HX597834	0	5	7
12	KKM531	BMW	i9	2020	5TFAW5F12FL002262	0	3	11
13	JNCS515	TOYOTA	Tundra	2021	1GAABB8Z3M0115561	0	8	12
14	HAPPY99	Volkswagen	Beetle	2022	1HTKJHGF37H414790	0	1	2
43	HS6531	JEEP	WRANGLER	2019	3DMDF2WE3HSK34142	0	4	9
44	LK9472	LAND ROVER	SPORT	2009	9DPKG2SM3ISK32587	0	3	8

CKY_CUSTOMER:

C_ID	C_TYPE	C_STREET	C_CITY	C_STATE	C_ZIPCODE	C_EMAIL	C_PHONE_NUM	C_PASSWORD
1	I	605 Pavonia Ave	Jersey City	NJ	07306	kz1244@nyu.edu	646-750-2639	□Mw橡 K□?□□皋
2	I	605 Pavonia Ave	Jersey City	NJ	07306	ct1855@nyu.edu	551-998-1234	□Mw橡 K□?□□皋
3	I	605 Pavonia Ave	Jersey City	NJ	07306	yl8264@nyu.edu	646-750-3726	□Mw橡 K□?□□皋
4	I	68 Pink Ave	Queen	NY	68721	cs665@gmail.com	582-203-6915	□Mw橡 K□?□□皋
5	C	605 Pavonia Ave	Jersey City	NJ	07306	cyt56@nyu.edu	5519981235	□Mw橡 K□?□□皋
6	C	2650 MacArthur Rd	Whitehall	PA	18052	st67@zinc.com	551-899-1356	□Mw橡 K□?□□皋
7	I	1560 Irene St	Bethlehem	PA	18017	larken@gmail.com	8442548364	□Mw橡 K□?□□皋
8	C	18811 Carmenita Rd	Cerritos	CA	90703	hfdsh@gmail.com	3344326955	□Mw橡 K□?□□皋
9	I	250 Station Cir	Dedham	MA	02026	cdh67@gmail.com	764-248-3578	□Mw橡 K□?□□皋
10	I	250 Station Cir	Dedham	MA	02026	elliec@gmail.com	781-234-0501	□Mw橡 K□?□□皋
11	C	18811 Carmenita Rd	Cerritos	CA	90703	bowlero@gmail.com	5629249363	□Mw橡 K□?□□皋
12	C	264 Springs Colony Cir	Altamonte Springs	FL	32714	dfrg67@filt.com	575-248-5231	□Mw橡 K□?□□皋
13	C	7950 Shoals Dr	Orlando	FL	32817	vd78@ginger.com	624-475-2445	□Mw橡 K□?□□皋
14	I	1648 Heliport Loop	Evansville	IN	47715	robertt@gmail.com	427-853-6542	□Mw橡 K□?□□皋
15	C	3882 Cedarstone Drive	Toledo	OH	43612	ht61@colab.com	785-244-4575	□Mw橡 K□?□□皋
16	I	3643 Victoria Street	Baton Rouge	LA	70810	romeror@gmail.com	678-242-2356	□Mw橡 K□?□□皋
17	C	4493 Boone Crockett Lane	Port Angeles	WA	98362	gre782@xman.com	524-464-2463	□Mw橡 K□?□□皋
18	C	1102 Pringle Drive	Chicago	IL	60606	ce47@wayn.com	367-235-5842	□Mw橡 K□?□□皋
19	I	2024 Goodwin Avenue	Spokane	WA	99205	jasonw@gmail.com	634-253-3664	□Mw橡 K□?□□皋
20	C	4817 Hershell Hollow Road	Johnson City	TN	37615	sf671@gmail.com	462-463-2534	□Mw橡 K□?□□皋

CKY_INDIVIDULE:

C_ID	C_TYPE	F_NAME	M_NAME	L_NAME	DLN	INSURANCE_CNAME	INSURANCE_PNUM
1	I	Kessy		Zhou	FD9385832	EP	2857395078462
2	I	Cherry		Tan	KE6384629	Wareness	85639682995
3	I	Yihan	Aria	Liu	HJ3672123	EP	5789275942356
4	I	Coco		Shine	5767667	vyuvyh	7877888
7	I	Charlie		Brown	99900104	CURE	146989375
9	I	Amy		Greenwood	dd7834	CURE	146989375
10	I	Ellie	W	Cotton	S999888801	GEICO	70892351
14	I	Robert	Jr	Thomson	A0002144	Allstate Insurance	257242267
16	I	Romero	H	Rice	G544-061-73-925-0	Allstate Insurance	550877653
19	I	Jason		White	G544061739250	National Car Services, LLC	430934820

CKY_CORPORATE:

C_ID	C_TYPE	CORPORATION_NAME	REGIS_NUM	EMPLOYEE_ID
5	C	CVS	dfr678	1234
6	C	Zinc Inc	du789	23767
8	C	GUI Company	bght347	348
11	C	Yee Company	152961	23
12	C	Filt Inc	35215	178
13	C	Ginger	14353	78
15	C	Colab	35146	61
17	C	Xman	14463	782
18	C	Wayn	57314	47
20	C	Gradescope	57435	351

CKY_COUPON:

COUPON_ID	DISCOUNT	VALID_FROM	VALID_TO
1	0.3	2022-01-20	2022-03-10
2	0.2	2022-02-15	2022-04-03
3	0.25	2022-05-11	2022-09-11
4	0.1	2022-06-23	2022-11-23
5	0.15	2022-07-08	2022-12-08
6	0.2	(Null)	(Null)
7	0.1	(Null)	(Null)
8	0.15	(Null)	(Null)
9	0.25	2022-11-11	2023-04-11
10	0.1	2022-12-14	2023-05-14

CKY_COUP_CUST:

COUPON_ID	C_ID	COUPON_TYPE	C_TYPE
1	1	I	I
1	14	I	I
2	2	I	I
2	14	I	I
3	3	I	I
3	16	I	I
4	4	I	I
4	16	I	I
5	7	I	I
5	16	I	I
6	5	C	C
6	13	C	C
6	17	C	C
6	20	C	C
7	6	C	C
7	12	C	C
7	15	C	C
7	20	C	C
8	8	C	C
8	11	C	C
8	18	C	C
9	7	I	I
9	9	I	I
9	19	I	I
10	10	I	I
10	19	I	I

CKY_RENTAL:

R_ID	PICK_DATE	DROP_DATE	PICK_LOCATION	DROP_LOCATION	START_ODO	END_ODO	ODO_LIMIT	COUPON_ID	VEHICLE_ID	C_ID	C_TYPE
1	2022-05-03	2022-05-12	3	3	0	2300	150	(Null)	14	1	I
2	2022-06-08	2022-06-16	5	7	0	800	50	3	8	3	I
3	2022-04-04	2022-04-11	11	5	0	2400	200	6	43	5	C
4	2022-03-15	2022-03-15	4	8	0	300	200	(Null)	7	7	I
5	2022-02-10	2022-02-13	12	6	0	2000	200	(Null)	1	9	I
6	2022-03-09	2022-03-11	6	7	0	1000	300	7	1	12	C
7	2022-01-04	2022-01-08	7	10	0	800	200	(Null)	10	14	I
8	2022-04-09	2022-04-16	5	9	0	2000	300	(Null)	43	16	I
9	2022-03-29	2022-04-02	1	11	0	1700	200	8	12	18	C
10	2022-04-29	2022-05-04	2	12	0	2190	300	6	13	20	C

CKY_INVOICE:

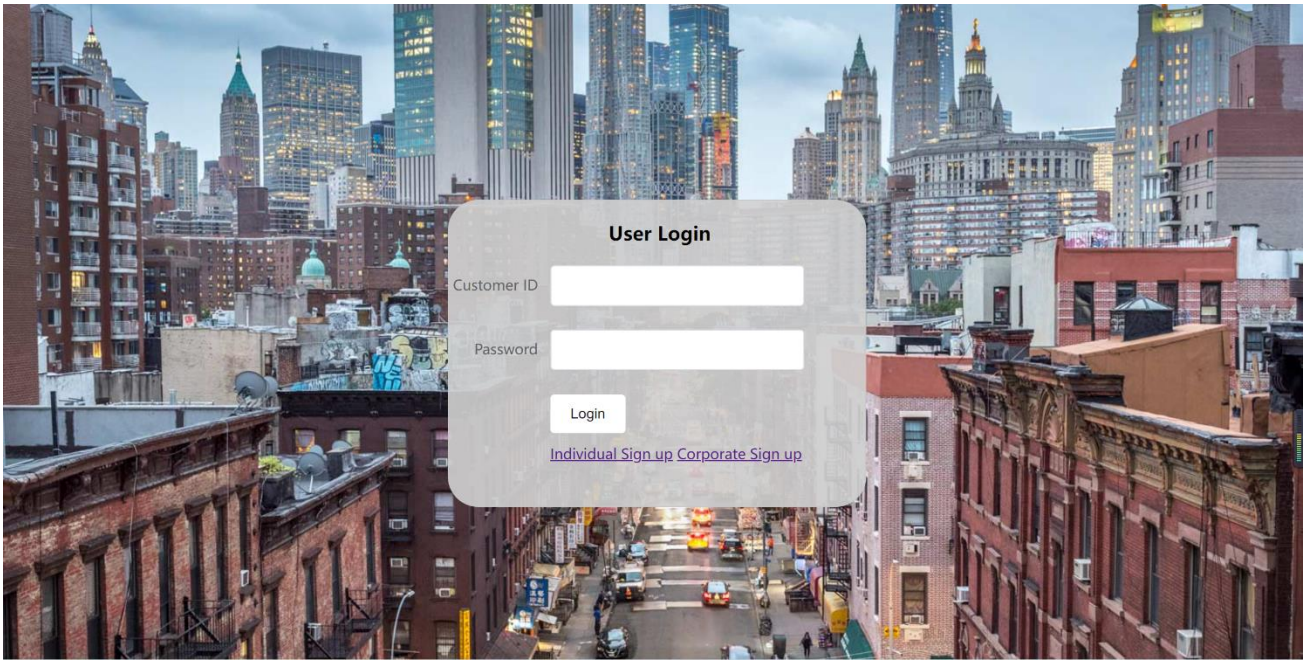
INVOICE_ID	INVOICE_DATE	INVOICE_AMOUNT	R_ID
46	2022-05-12	1800	1
47	2022-06-16	660	2
48	2022-04-11	2880	3
49	2022-03-15	330	4
50	2022-02-13	4960	5
51	2022-03-11	468	6
52	2022-01-08	200	7
53	2022-04-16	400	8
54	2022-04-02	5032	9
55	2022-05-04	4256	10
56	2022-05-09	140	11

CKY_PAYMENT:

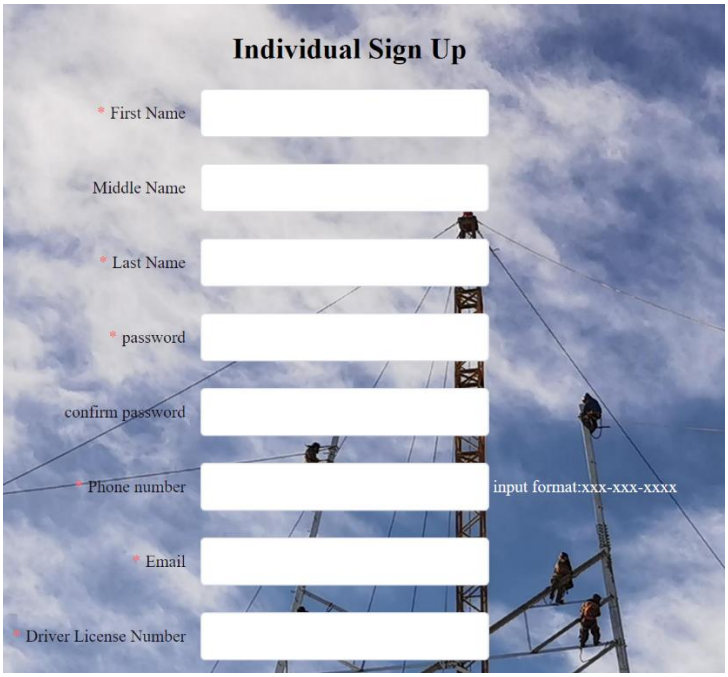
PAYMENT_ID	PAY_METHOD	CARD_NUM	PAY_AMOUNT	INVOICE_ID	PAY_DATE
20	gift card	1234567890123456	500	46	2022-05-12
21	credit	1234567890234567	1300	46	2022-05-12
22	debit	1234567890123456	660	47	2022-06-16
23	gift card	1234567890123456	1000	48	2022-04-11
24	debit	1234567890234567	940	48	2022-04-11
25	credit	1234567890345678	940	48	2022-04-11
26	debit	1234567890123456	165	49	2022-03-15
27	debit	1234567890234567	165	49	2022-03-15
28	credit	1234567890098765	4960	50	2022-02-13
29	gift card	1234567890123456	468	51	2022-03-11
30	debit	1234567890456789	100	52	2022-01-08
31	gift card	1234567890123456	100	52	2022-01-08
32	debit	1234567890567890	400	53	2022-04-16
33	gift card	1234567890123456	2000	54	2022-04-02
34	debit	1234567890098765	3000	54	2022-04-02
35	credit	1234567890567890	32	54	2022-04-02
36	credit	1234567890098765	4256	55	2022-05-04
37	debit	2345678901234567	140	56	2022-05-09

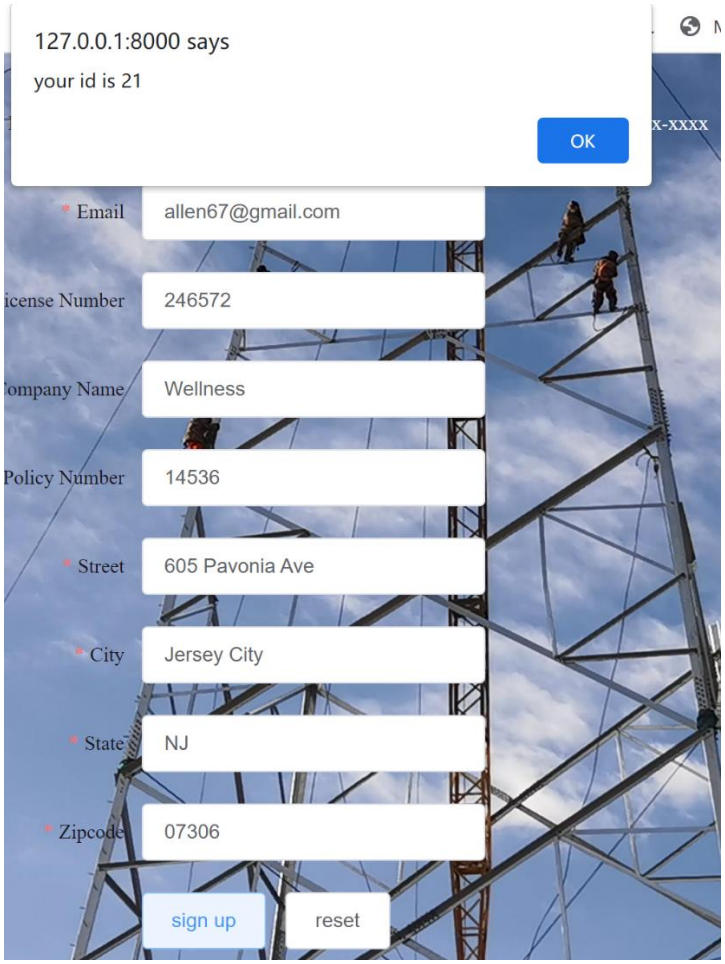
5. Web Application Screenshots

Home Page(Login)

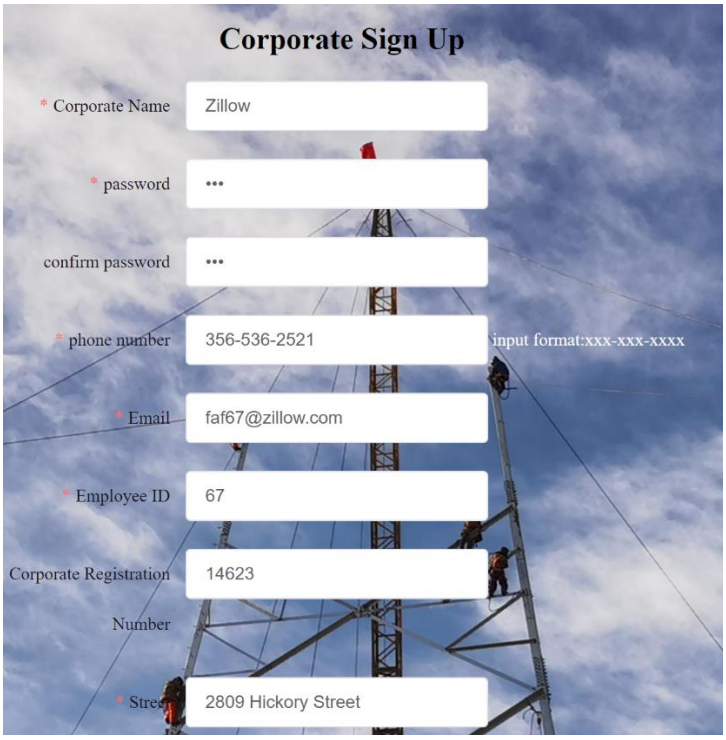


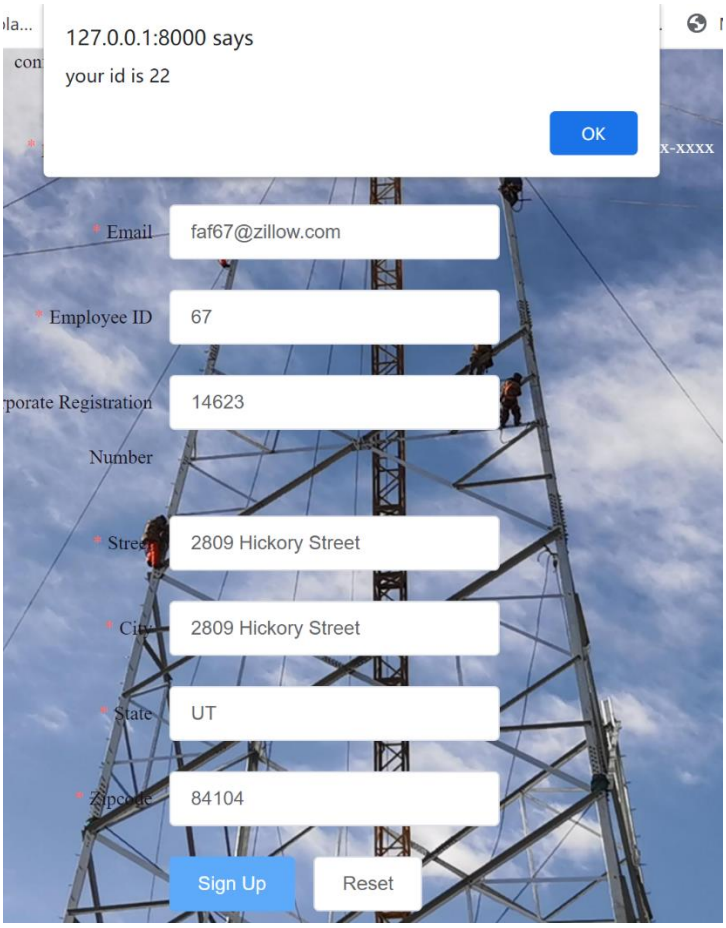
Individual Signup



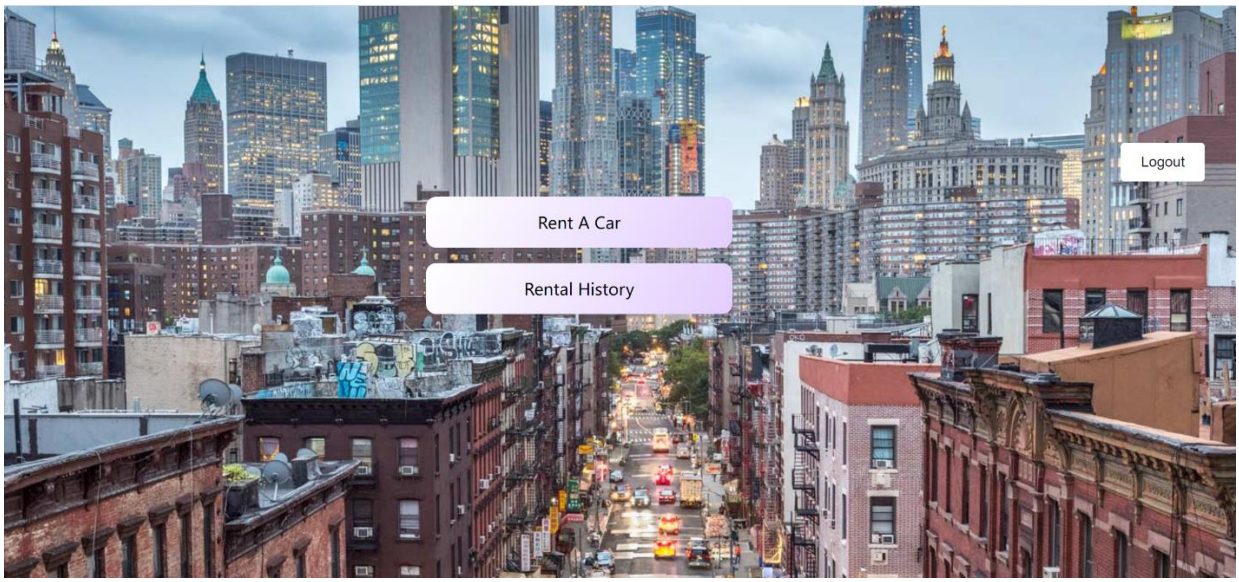

127.0.0.1:8000 says
your id is 21
OK
x-xxxx
* Email
allen67@gmail.com
icense Number
246572
ompany Name
Wellness
Policy Number
14536
* Street
605 Pavonia Ave
* City
Jersey City
* State
NJ
* Zipcode
07306
sign up
reset

Corporate Signup:

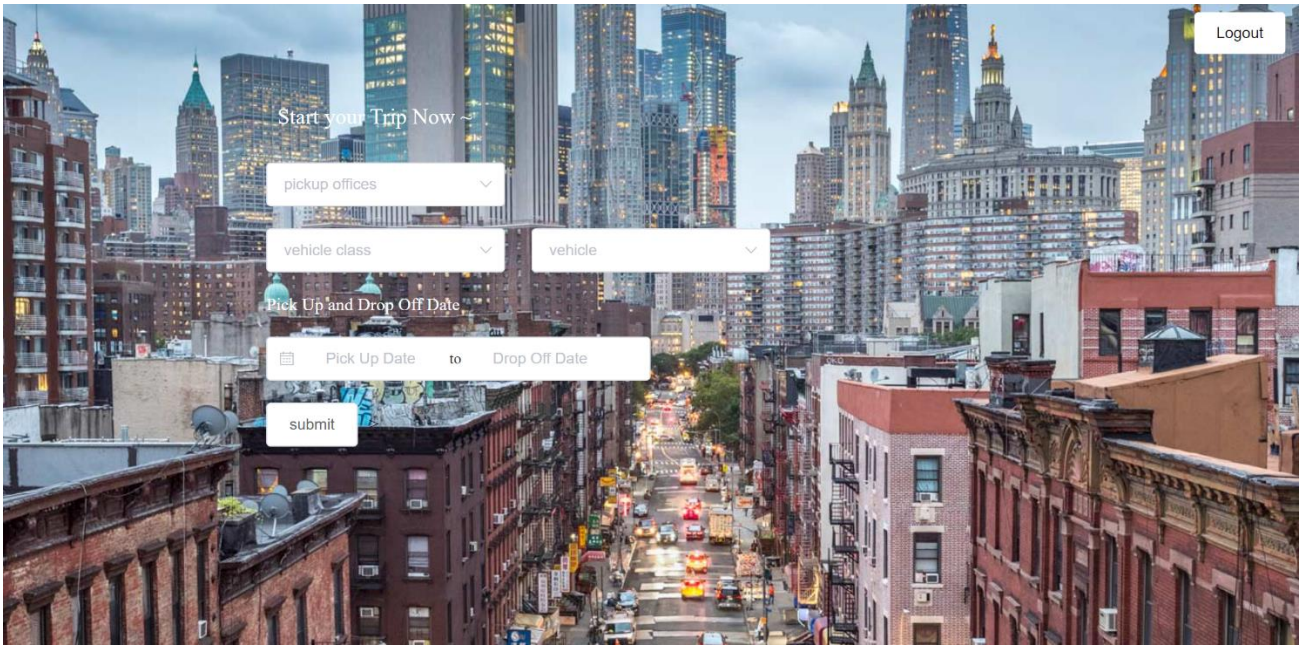




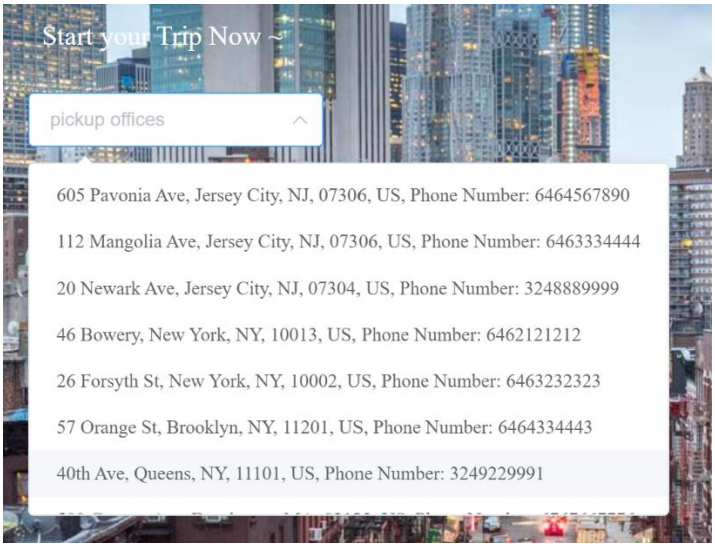
Service Options(For Customers)



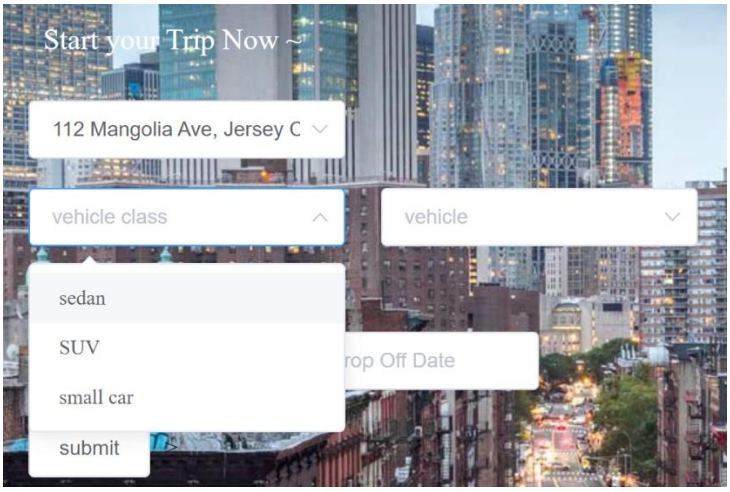
Rent A Car



Select Pickup Office:

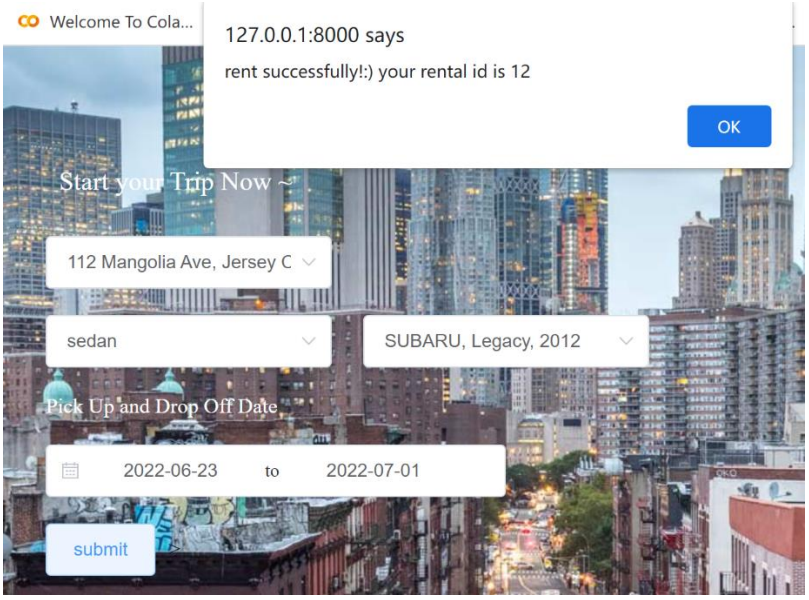


Select Vehicle Class and Vehicle:

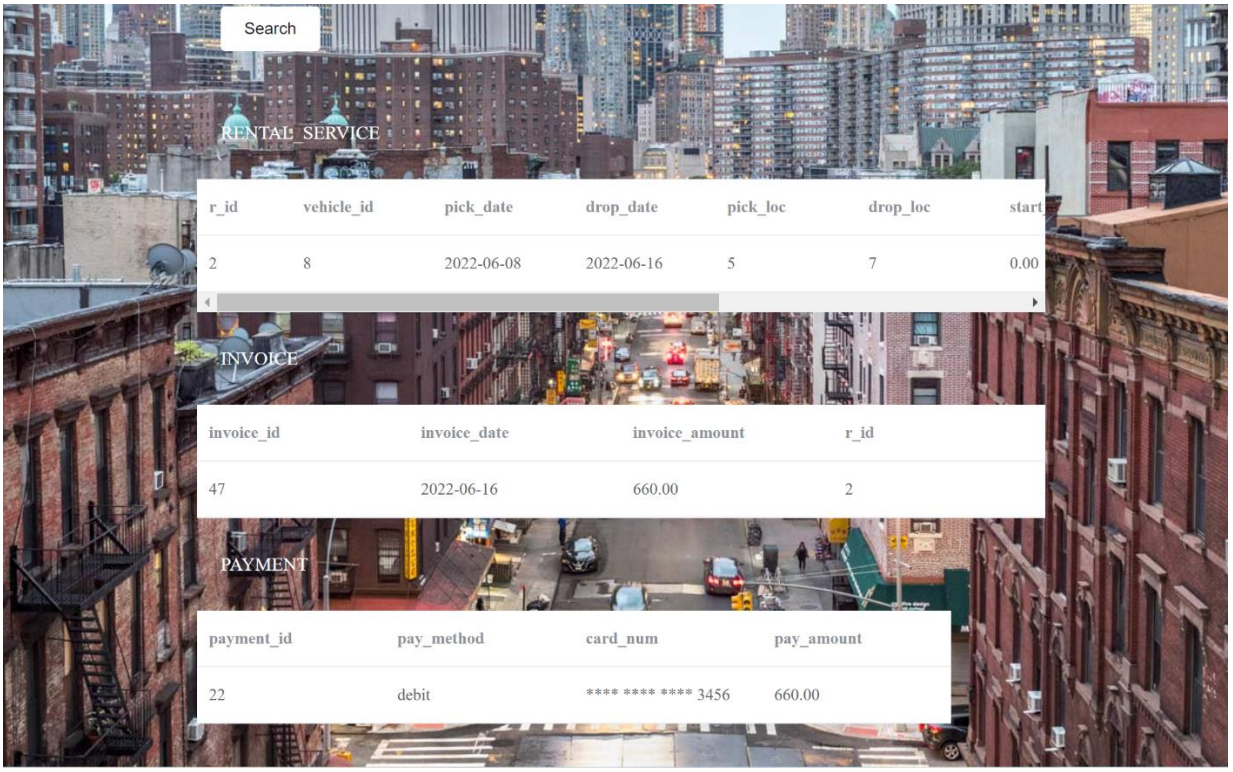


Select rental period:

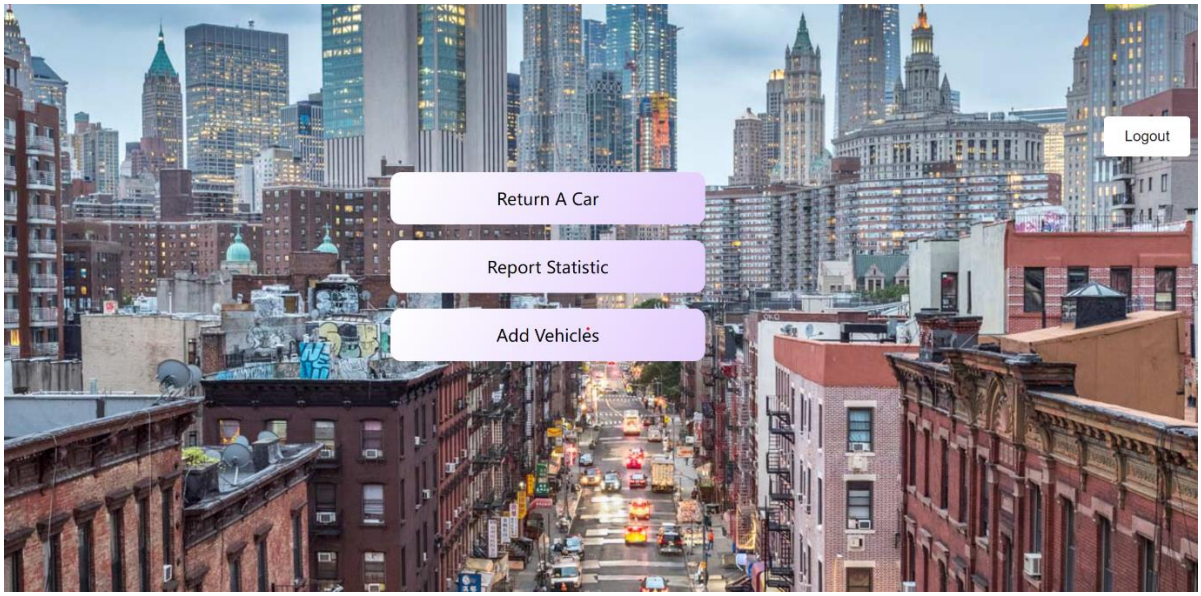
Complete:



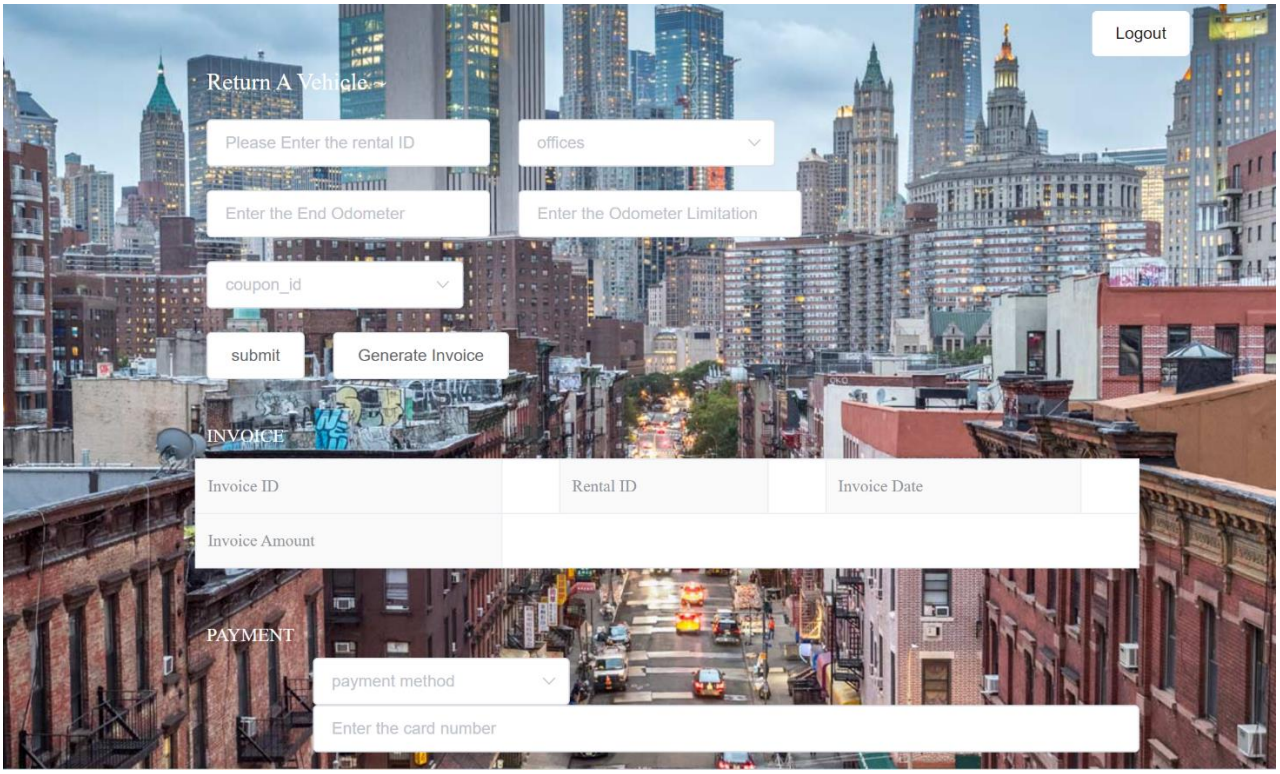
Rental History of Customers:



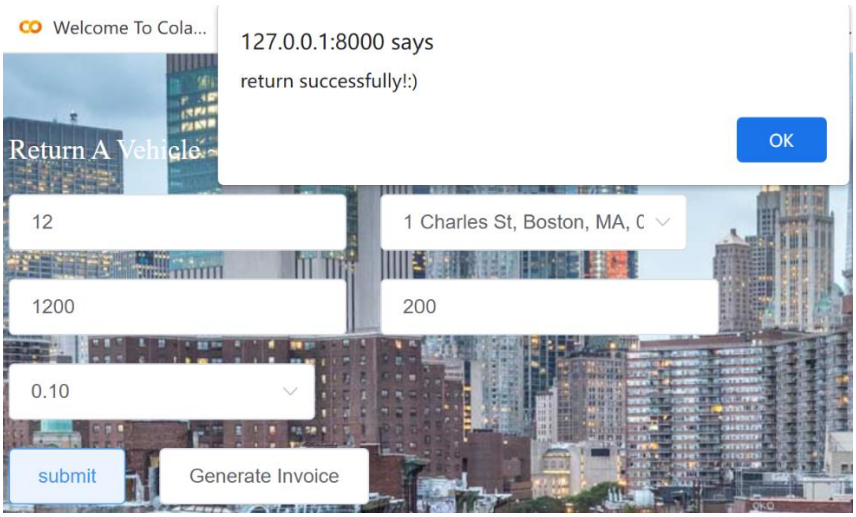
Officer Options:



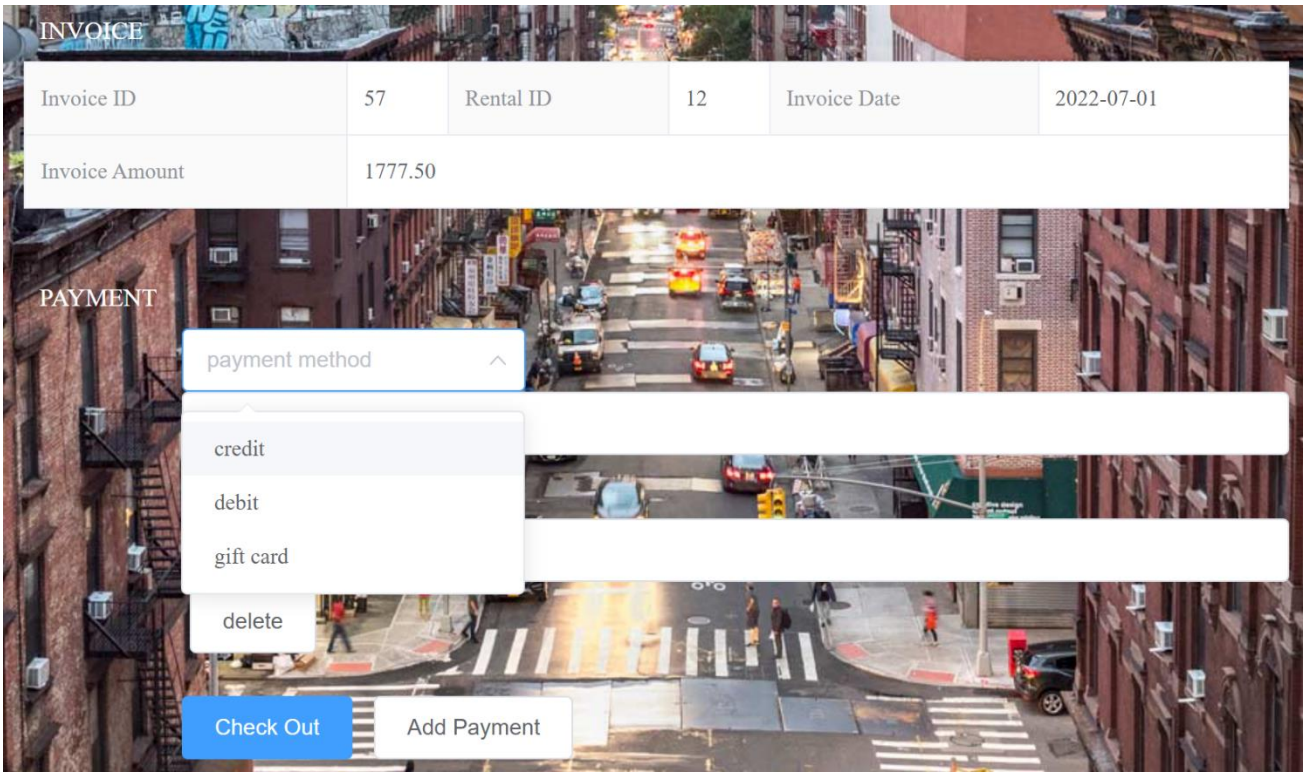
Return A Car:



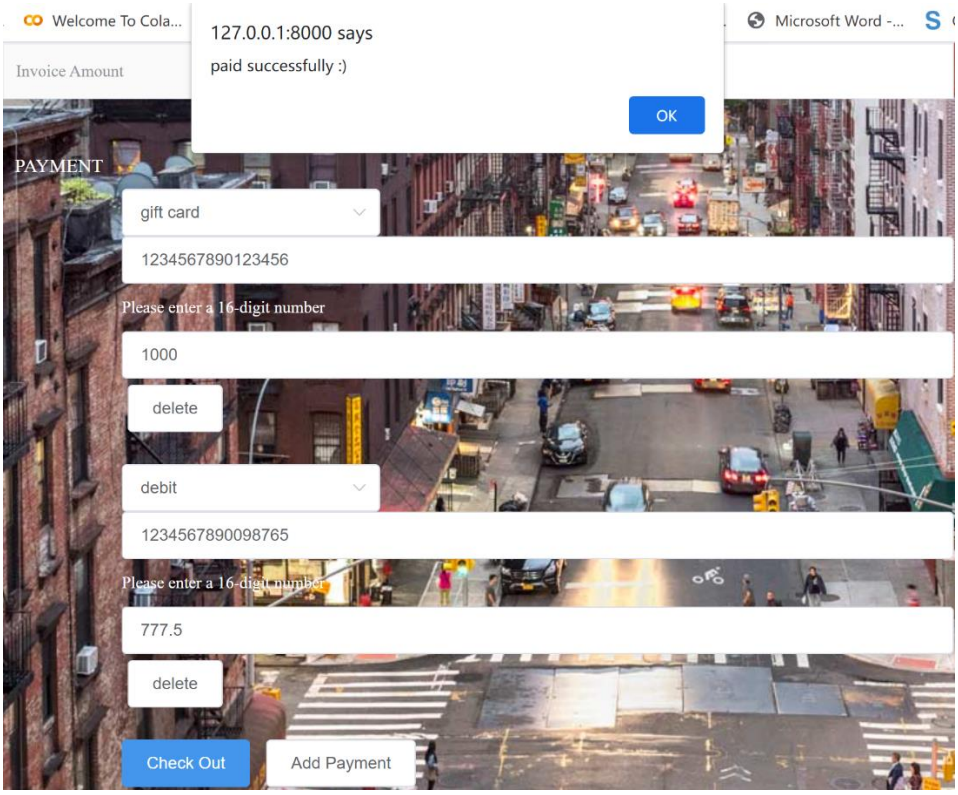
Return Successfully



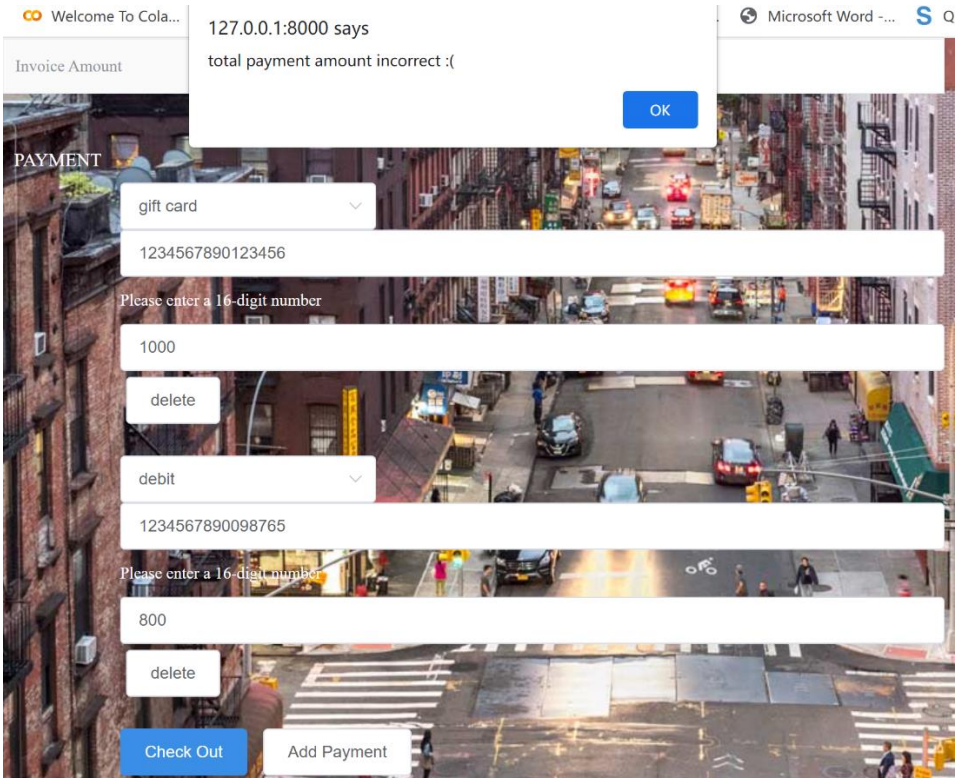
Generate Invoice



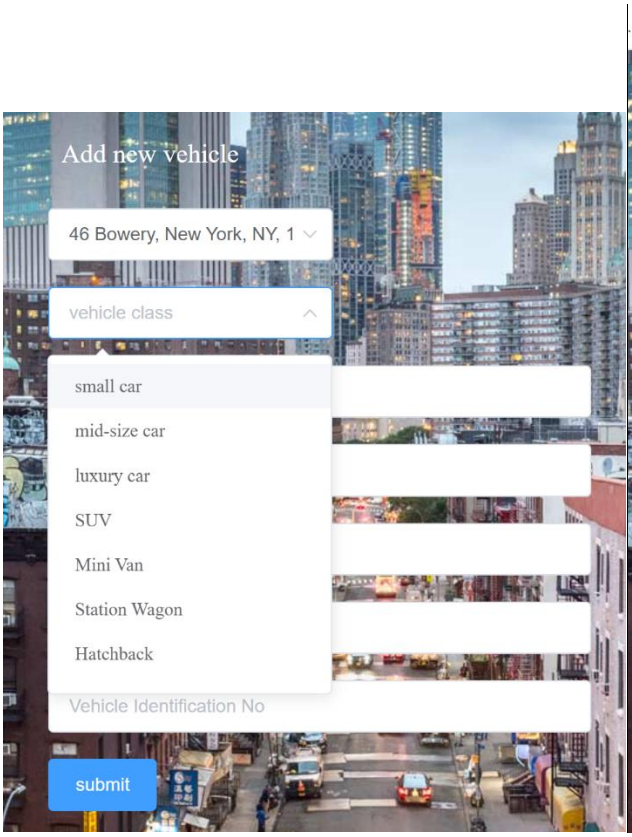
Multi-card Payment

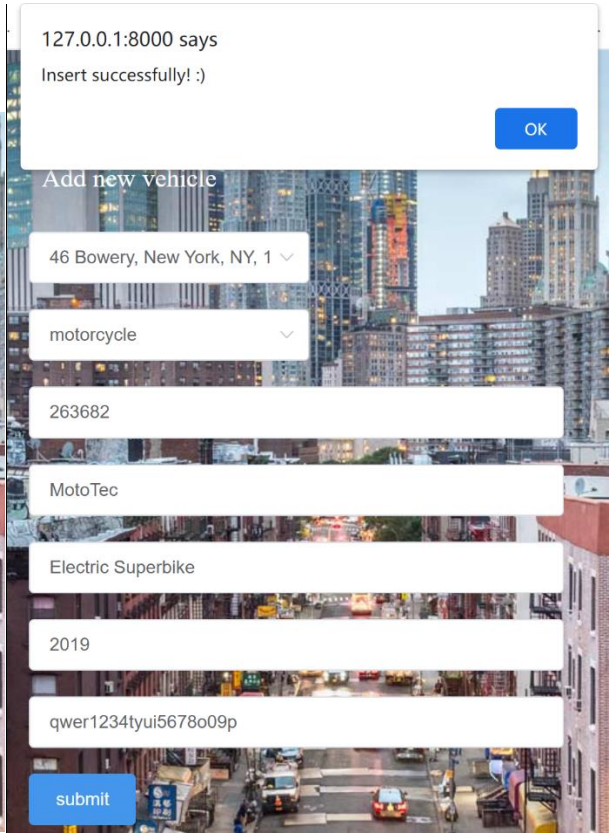


Payment failed

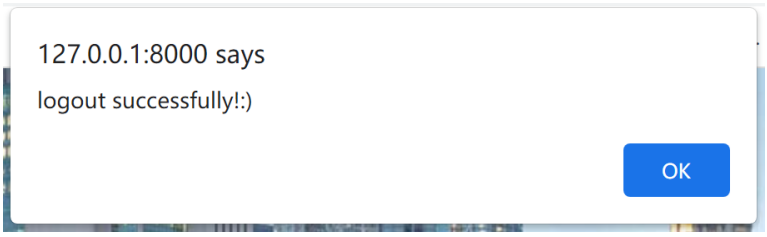


Add New Vehicles



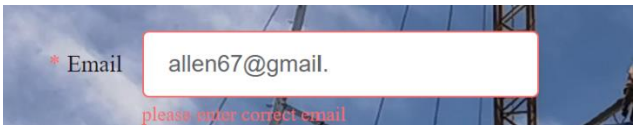


Log Out



6. Security Features

Email format check:



Special character check:

Empty input check:

Password Encryption:

C_ID	C_TYPE	C_STREET	C_CITY	C_STATE	C_ZIPCODE	C_EMAIL	C_PHONE_NUM	C_PASSWORD
1	1	605 Pavonia Ave	Jersey City	NJ	07306	kz1244@nyu.edu	646-750-2639	□Mw樑 K□?□□皐
2	1	605 Pavonia Ave	Jersey City	NJ	07306	ct1855@nyu.edu	551-998-1234	□Mw樑 K□?□□皐
3	1	605 Pavonia Ave	Jersey City	NJ	07306	yl8264@nyu.edu	646-750-3726	□Mw樑 K□?□□皐

Hidden card number:

7. Learning

From this project, I really practice and learn a lot about database. For example, I learnt how to analyze a business case and design a logical and relational model. During designing, more and more details, assumptions and constraints figured out. Then, we begin to combine web app with database designed. The most meaningful thing is that I learnt a lot of new knowledge, for example, I learnt how to encrypt important information of customers, how to keep login state of customers with session. The most important thing is that we learnt how to develop a project in a team. We always support each other when there're some confusing problems we often discuss and try to figure it out together. During solving all kinds errors and problems, I gained a lot. For example, I learnt the how do the delete policy work in database. If it is wrongly design, it might cause a lot of troubles in the after forwards development.

The strength of our system is that our system care about security, since we encrypt customers' password with key so that hackers can't easily steal customers' information. Also, beyond required SQL query, we also carefully think about business cases and generate many meaningful statistics using SQL I learnt from lessons.

The constraint for this project is that the time is very tight we only spent a week developing this project and try our best to make it better. Hence, I think we might improve the system. For example, we could create functions to send coupons to customers through email or text, set some constraints on wrongly input to improve the system security, display more images to make front-end looks colorful and attractive and also make system more convenient to use.

8. Business Analysis

Q1 Three Table Join

When a customer wants to return a car, system should select from table according to his/her customer id to find whether there're some valid coupons for this customer. Since coupon and customer have many-to-many relationship, we need to join three join table (cky_coupon, cky_coup_cust and cky_customer) to find coupon according to customer id.

```
14 WITH THREE AS(SELECT * FROM cky_coupons NATURAL JOIN cky_coup_cust NATURAL JOIN cky_customer)
15 SELECT THREE.COUPON_ID,THREE.DISCOUNT
16 FROM THREE, (SELECT DROP_DATE FROM cky_rental WHERE R_ID = 6) AS dropdate
17 WHERE   (C_ID = (SELECT C_ID FROM cky_rental WHERE R_ID = 6) AND
18         (((SELECT COUNT(*) FROM THREE)=0) OR ((SELECT COUNT(*)FROM THREE) >0) AND ((VALID_FROM IS NULL)
19          OR (VALID_FROM<=dropdate.DROP_DATE AND VALID_TO>=dropdate.DROP_DATE))))
```

信息 结果1 概况 状态

COUPON_ID	DISCOUNT
7	0.1

Q2 Multi-row Subquery

Compare all the car rental consumption with every average consume of each office, to find the higher rental sale service. It can evaluate the business level of the office, the more rental service in an office found, the better the business ability of this office is. Company could focus on these higher sale rental service to figure why this office and why this rental service could have such high sale, which might be beneficial for future business decisions.

```
SELECT a.PICK_LOCATION, a.R_ID, a.C_ID, c.PAY_AMOUNT
FROM cky_rental a NATURAL JOIN cky_invoice b NATURAL JOIN cky_payment c
WHERE c.PAY_AMOUNT>= ALL (SELECT avg(f.PAY_AMOUNT)
                          FROM cky_rental d NATURAL JOIN cky_invoice e NATURAL JOIN cky_payment f
                          GROUP BY d.PICK_LOCATION)
Order by a.PICK_LOCATION;
```

Q3 Correlated Subquery

Find customers whose spend is lower than average turnover of the office they consume. WOW may give these customers more coupons to encourage them to consume.

Find customers whose spend is higher than average turnover of the office they consume. WOW may let these customers become VIP who would get exclusive customer service. VIP can also get priority to rent new cars and high-level cars to encourage them to consume.

```
SELECT a.C_ID, c.C_EMAIL, a.PICK_LOCATION, b.INVOICE_AMOUNT
FROM cky_rental a natural join cky_invoice b NATURAL JOIN cky_customer c
WHERE b.INVOICE_AMOUNT < (SELECT AVG(e.INVOICE_AMOUNT)
                          FROM cky_rental d natural join cky_invoice e
                          WHERE d. PICK_LOCATION = a.PICK_LOCATION)
Order by a.PICK_LOCATION;

SELECT a.C_ID, c.C_EMAIL, a.PICK_LOCATION, b.INVOICE_AMOUNT
FROM cky_rental a natural join cky_invoice b NATURAL JOIN cky_customer c
WHERE b.INVOICE_AMOUNT > (SELECT AVG(e.INVOICE_AMOUNT)
                          FROM cky_rental d natural join cky_invoice e
                          WHERE d. PICK_LOCATION = a.PICK_LOCATION)
Order by a.PICK_LOCATION;
```

Customer with Low Consumption

customer_id	email	office	consumption
6	st67@zinc.com	2	1777.50
2	ct1855@nyu.edu	3	140.00
16	romeror@gmail.com	5	400.00

Customer with High Consumption

customer_id	email	office	consumption
20	sf671@gmail.com	2	4256.00
1	kz1244@nyu.edu	3	1800.00
3	yl8264@nyu.edu	5	660.00

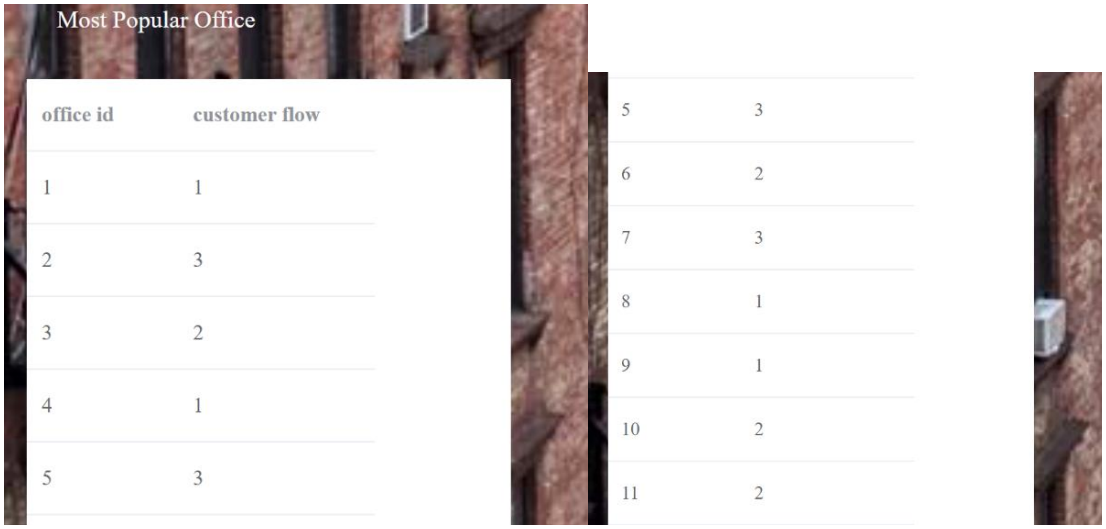
Q4 SET operator query

Check the number of rental services associated with an office, which means the number of records of renting or returning a car at this office. If the office has a high number of records (either renting or returning cars), it means that this office is popular, and WOW can consider opening more offices in the area where the office is located.

```
SELECT PICK_LOCATION AS Office, COUNT(*) AS Times FROM (SELECT * FROM cky_rental WHERE PICK_LOCATION= 2
UNION
SELECT * FROM cky_rental WHERE DROP_LOCATION=2) AS UN
GROUP BY UN.PICK_LOCATION;
```

信息 结果1 概况 状态

Office	Times
2	2
3	1



Q5 WITH clause

When a customer wants to return a car, system should select from table according to his/her customer id to find whether there're some valid coupons for this customer. Beyond that, system should also filter the valid coupons (dropout date should fall in coupon valid period). Hence, we use with clause.

```
WITH THREE AS(SELECT * FROM cky_coupons NATURAL JOIN cky_coup_cust NATURAL JOIN cky_customer)
SELECT THREE.COUPON_ID,THREE.DISCOUNT
FROM THREE, (SELECT DROP_DATE FROM cky_rental WHERE R_ID = 6) AS dropdate
WHERE  (C_ID = (SELECT C_ID FROM cky_rental WHERE R_ID = 6) AND
       ((((SELECT COUNT(*) FROM THREE)=0) OR ((SELECT COUNT(*)FROM THREE) >0) AND ((VALID_FROM IS NULL)
        OR (VALID_FROM<=dropdate.DROP_DATE AND VALID_TO>=dropdate.DROP_DATE))))
```

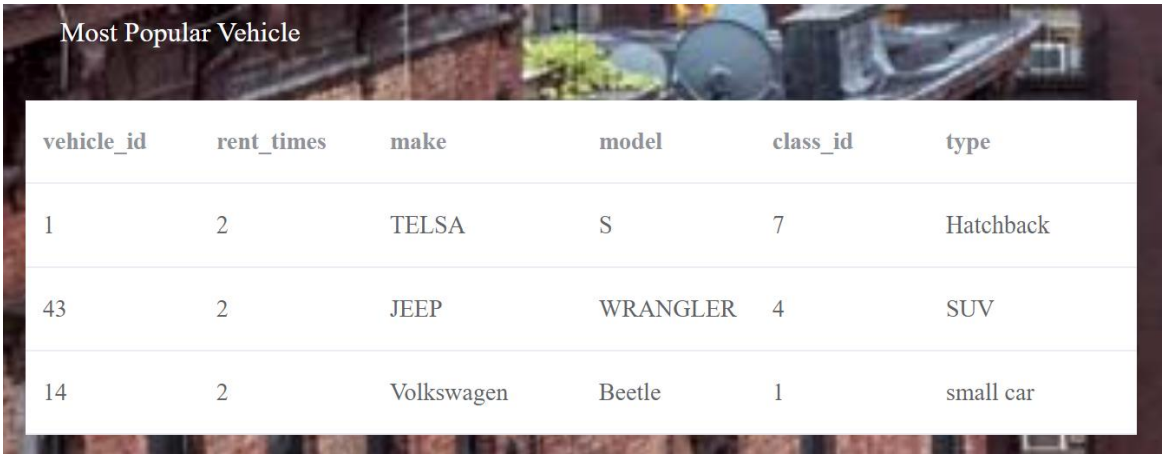
信息 结果1 概况 状态

COUPON_ID	DISCOUNT
7	0.1

Q6 TOP-N query

Find the three vehicles that have been rented the most times, which means that these types of cars are the most popular. Thus, WOW may buy more cars of these types for rental.

```
SELECT c.VEHICLE_ID, count(c.VEHICLE_ID) as Times, b.MAKE, b.V_MODEL, a.CLASS_ID, a.CLASS_NAME
FROM cky_vehicle_class a NATURAL JOIN cky_vehicle b NATURAL JOIN cky_rental c
GROUP BY c.VEHICLE_ID
ORDER BY Times DESC
limit 3;
```



vehicle_id	rent_times	make	model	class_id	type
1	2	TELSA	S	7	Hatchback
43	2	JEEP	WRANGLER	4	SUV
14	2	Volkswagen	Beetle	1	small car

9. Extra Features

Index

Build BTree index on column “C_TYPE” from table CKY_CUSTOMER, because there are only two values in “C_TYPE” which is suitable for creating index to accelerate the speed of searching a customer.

```
1 CREATE INDEX idx_ctype ON cky_customer(C_TYPE) USING BTREE;
2 SHOW INDEX FROM cky_customer FROM cky_scheme;
```

信息 结果1 概况 状态

Table	Non_unique	Key_name	Seq_in_index	Column_nam	Collation	Cardinality	Sub_pa	Packe	Null	Index_type	Com	Inde	Visible	Expression
cky_custome	0	PRIMARY	1	C_ID	A	17	(Null)	(Null)		BTREE			YES	(Null)
cky_custome	0	PRIMARY	2	C_TYPE	A	17	(Null)	(Null)		BTREE			YES	(Null)
cky_custome	1	C_ID	1	C_ID	A	17	(Null)	(Null)		BTREE			YES	(Null)
cky_custome	1	idx_ctype	1	C_TYPE	A	2	(Null)	(Null)		BTREE			YES	(Null)

Build BTree index on column “PAY_METHOD” from table CKY_PAYMENT, because there are only three values in “PAY_METHOD” which is suitable for creating index to accelerate the speed of searching a method of payment. Customer may pay the invoice with the combination of multiple payment methods.

```
1 CREATE INDEX idx_method ON cky_payment(PAY_METHOD) USING BTREE;
2 SHOW INDEX FROM cky_payment FROM cky_scheme;
```

信息 结果1 概况 状态

Table	Non_unique	Key_name	Seq_in_index	Column_nam	Collation	Cardinality	Sub_pa	Packe	Null	Index_type	Com	Inde	Visible	Expression
cky_payment	0	PRIMARY	1	PAYMENT_ID	A	18	(Null)	(Null)		BTREE			YES	(Null)
cky_payment	1	INVOICE_ID	1	INVOICE_ID	A	11	(Null)	(Null)		BTREE			YES	(Null)
cky_payment	1	idx_method	1	PAY_METHO	A	3	(Null)	(Null)		BTREE			YES	(Null)

Build index on column “C_ID”, “vehicke_id”, “pick_loc” and “drop_loc” from cky_rental since they are the columns that used to join with other tables frequently. Hence, create index on them could accelerate joining and searching speed.

```sql
CREATE INDEX idx_type ON cky_rental(C_TYPE) USING BTREE;
CREATE INDEX idx_vehicleid ON cky_rental(VEHICLE_ID);
CREATE INDEX idx_coupon_id ON cky_rental(COUPON_ID);
CREATE INDEX idx_pickloc ON cky_rental(PICK_LOCATION);
CREATE INDEX idx_droploc ON cky_rental(DROP_LOCATION);
SHOW INDEX FROM cky_rental FROM cky_scheme;
```

信息 结果1 概况 状态

Table	Non_unique	Key_name	Seq_in_index	Column_nam	Collation	Cardinality	Sub_part	Packe	Null	Index_type	Com	Index_	Visible	Expression
cky_rental	0	PRIMARY	1	R_ID	A	17	(Null)	(Null)		BTREE			YES	(Null)
cky_rental	1	idx_cid	1	C_ID	A	12	(Null)	(Null)		BTREE			YES	(Null)
cky_rental	1	idx_type	1	C_TYPE	A	2	(Null)	(Null)		BTREE			YES	(Null)
cky_rental	1	idx_vehicleid	1	VEHICLE_ID	A	12	(Null)	(Null)		BTREE			YES	(Null)
cky_rental	1	idx_coupon_i	1	COUPON_ID	A	5	(Null)	(Null)	YES	BTREE			YES	(Null)
cky_rental	1	idx_pickloc	1	PICK_LOCATI	A	9	(Null)	(Null)		BTREE			YES	(Null)
cky_rental	1	idx_droploc	1	DROP_LOCA	A	12	(Null)	(Null)	YES	BTREE			YES	(Null)