Principles of Database Systems Fall 21

Web-Based Vehicle Rental Management System

Project Report

Jianqiao Li

New York University, Tandon School of Engineering

2021/12/16

Contents

1 Execute Summary **3**
- 1.1 Business Case . . . 3
- 1.2 Approaches Adopted . . . 3
- 1.3 Logical Relational Models . . . 3

2 Technical Requirements **4**
- 2.1 Software And Database . . . 4
- 2.2 Programming Languages . . . 5

3 DDL File **5**

4 Schema **12**

5 Screenshots **15**

6 Security Implementation **19**

7 Reflection **20**

8 Queries **21**
- 8.1 Table Join . . . 21
- 8.2 Multi Row Subquery . . . 22
- 8.3 Correlated Subquery . . . 22
- 8.4 Set Operator . . . 23
- 8.5 Top-N Query . . . 24

1 Execute Summary

1.1 Business Case

This project aims to develop a web-based database system called VRMS specifically designed to provide vehicle rental service and information management. As a car fan myself, I've always wanted to develop a business website where people can booking the rental of different types of cars. Two distinctive features of this website that set itself from other existing car rental service websites are: first, this website can support all kinds of vehicles and not just cars. In this project, I added the two-wheeler vehicle type, or motorcycle, but more can be added in future usage; second, this website would show users detailed description of vehicle configurations, fro example, gearbox and ABS. Different cars from the same brand and model can have different configurations and may come with different packages, so those information can be useful when users are deciding on which specific car to rent, instead of only choosing the brand and model.

1.2 Approaches Adopted

The web-based management system is basically incorporating relational database with we server API. However in reality, administration and general users don't share the web interface, thus two separate user modules are introduced. In general, the approach adopted is to use PHP and HTML scripting language to develop web interface running on localhost and connect it with database server every time query is needed.

1.3 Logical Relational Models

There are mainly two modules in this system design: users and admins. The specific business case is detailed as following:

Admin

1. Dashboard: admin can see two wheeler and four wheeler vehicle detail in brief.
2. Brand: admin can manage brands (Add/Update).
3. Two Wheeler: admin can manage two wheeler vehicle (Add/Update).
4. Four Wheeler: admin can manage four wheeler vehicle (Add/Update).
5. Pages: the admin can manage about us and contact us pages.
6. Reg Users: admin can view detail of registered users.
7. Two/Four Wheeler Booking: admin can view total new booking, total approved booking, total unapproved booking and all booking. Admin has also right to approve and unapproved the booking.
8. Search: admin can search two wheeler and four wheeler vehicle booking by users name and Booking Id.

9. Profile Update: Admin can also update his profile, change the password and recover the password.

General Users

1. Home Page: Users can see the listed vehicles on the home page.
2. About Us: Users can view about us page.
3. Vehicle: User can view total two wheeler and four wheeler vehicles in details.
4. Contact us: Users can view the contact us page.
5. Login/Register: guest user can register himself/herself.

Registered Users (Additional functionalities)

1. My Booking: User can view the detail of booking like status of booking, how much cost will be taken and also take a print of invoice.
2. My Account: User can update his/her profile, change the password and recover the password.

The logical and relational model designed for this application is based on the original version in project part 1 with modifications and simplifications. During the designing process, I came to realize that the support from original model is over-fitting, so the finalized model is simplified with down to only 11 tables in total. Also, modifications is made on the super-type/sub-type relationships of web users. Due to the actual implementation, I decided to separate the website into two correlating modules for users and administrators, so the sub-type based on authority is removed, and registered/unregistered users are differentiated based on SQL queries instead of sub-typing. However, invoice record and payment record are kept, though the payment functionality is not implemented for simplicity.

Several assumptions are made:

1. Assume the website only offers two types of vehicles: 2-wheel and 4-wheel. Though admin can add more vehicle types but the database system need to be adjusted.
2. Assume one particular user can make multiple bookings on different vehicles.
3. Assume admin and users are separate entities but an admin can be a user as well.
4. Assume a user can have an account without any booking history.

2 Technical Requirements

2.1 Software And Database

In order to develop a web-based system, I decided to use XAMPP during implement. XAMPP is an open-source cross-platform web server solution stack package developed by Apache, and the reason I chose this package is that I can solve web developing and database connection in one bundle since XAMPP supports Apache HTTP server, MySQL, and scripting languages like

PHP. Furthermore, the database design is done through Oracle Data Modeler and I used online website "sqlines.com" to convert SQL DDL code into MySQL code. With the MySQL DDL code, I implemented the database connection through an open-source administration tool phpMyAdmin: creating the database and importing the script. In this manner, I don't have to edit MySQL through shell and all SQL queries made in section 8 can be done on web server.

2.2 Programming Languages

The programming language used in this project are PHP, HTML and CSS. The major dynamic content realization is done through PHP, while HTML determines the general structure of the web page and HTML markup is incorporated into all PHP files. As for the CSS, it is style sheet that used to describe content presentation. Though the actually code of CSS included in this project is limited, it plays a major role in ascetics of web page appearance.

3 DDL File

The following Oracle DDL code is generated from the relational model using Oracle Data Modeler, and note that initial data and images are inserted as well:

```
-- Generated by Oracle SQL Developer Data Modeler 19.1.0.081.0911
--   at:        2021-12-14 14:55:25 EST
--   site:      Oracle Database 11g
--   type:      Oracle Database 11g

CREATE TABLE jl_admin (
    id             NUMBER(10) NOT NULL,
    adminname      VARCHAR2(120) NOT NULL,
    username       VARCHAR2(50) NOT NULL,
    mobilenumber   NUMBER(10) NOT NULL,
    email          VARCHAR2(200) NOT NULL,
    passcode       VARCHAR2(200) NOT NULL,
    adminregdate   TIMESTAMP WITH TIME ZONE NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE jl_admin ADD CONSTRAINT jl_admin_pk PRIMARY KEY ( id );

INSERT INTO jl_admin (id, adminName, username, mobilenumber, email, passcode, adminregdate)
VALUES
(1, Fredrick Li, admin, 9178645818, jl7136@nyu.edu, 104647745, 2021-12-13 04:01:29);

CREATE TABLE jl_bookingcar (
    id              NUMBER(10) NOT NULL,
    bookingdate     VARCHAR2(120) NOT NULL,
    returndate      VARCHAR2(120) NOT NULL,
    bookingnumber   VARCHAR2(120) NOT NULL,
```

```
    creationdate    TIMESTAMP WITH TIME ZONE NOT NULL,
    totalcost       NUMBER(10, 2) NOT NULL,
    remark          VARCHAR2(120) NOT NULL,
    status          VARCHAR2(120) NOT NULL,
    remarkdate      TIMESTAMP WITH TIME ZONE NOT NULL,
    user_id      NUMBER(10) NOT NULL,
    vehicle_id   NUMBER(10) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE jl_bookingcar ADD CONSTRAINT jl_bookingcar_pk PRIMARY KEY ( id );

CREATE TABLE jl_bookingtwowheeler (
    id                  NUMBER(10) NOT NULL,
    bookingdate         VARCHAR2(120) NOT NULL,
    returndate          VARCHAR2(120) NOT NULL,
    bookingnumber       VARCHAR2(120) NOT NULL,
    creationdate        TIMESTAMP WITH TIME ZONE NOT NULL,
    totalcost           NUMBER(10, 2) NOT NULL,
    remark              VARCHAR2(120) NOT NULL,
    status              VARCHAR2(120) NOT NULL,
    remarkdate          TIMESTAMP WITH TIME ZONE NOT NULL,
    vehiclevcar_id   NUMBER(10) NOT NULL,
    user_id          NUMBER(10) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE jl_bookingtwowheeler ADD CONSTRAINT jl_bookingtwowheeler_pk PRIMARY KEY ( id );

CREATE TABLE jl_brand (
    id             NUMBER(10) NOT NULL,
    brandname      VARCHAR2(200) NOT NULL,
    brandlogo      VARCHAR2(200) NOT NULL,
    creationdate   TIMESTAMP WITH TIME ZONE NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE jl_brand ADD CONSTRAINT jl_brand_pk PRIMARY KEY ( id );

INSERT INTO jl_brand (id, brandName, brandLogo, creationdate) VALUES
(1, 'Honda', '55ccf27d26d7b23839986b6ae2e447ab.jpg', '2019-09-18 06:00:11'),
(2, 'Kinectic', '41b586905e6233e72b076191f8bf9512.png', '2019-09-18 06:01:42'),
(3, 'Bajaj', '7fdc1a630c238af0815181f9faa190f5.jpg', '2019-09-18 06:02:16'),
(4, 'Yamaha', '5c642ec854a6a92a56d7ebf0b9648eea.jpg', '2019-09-18 06:03:35'),
(5, 'TVS', 'c26be60cfd1ba40772b5ac48b95ab19b.png', '2019-09-18 06:04:35'),
(6, 'Mahindra', 'b64810fde7027715e614449aff1d595f.png', '2019-09-18 06:05:41'),
(7, 'Hyundai', '37e2b52f19da778fba43ab3c1897f840.png', '2019-09-18 06:06:09'),
(8, 'Renault', 'e76de47f621d84adbab3266e3239baee.png', '2019-09-18 06:07:43'),
```

```
(9, 'Volvo', '2d99ae9e904f880eef8feb4e61882b79.jpg', '2019-09-18 06:08:42'),
(10, 'Skoda', '4389d9b5e3ba297410a11afc0b8e101b.png', '2019-09-18 06:09:23'),
(11, 'Maruti Suzuki', '1f868f31d7bb9b00f86cad27759faf95.png', '2019-09-18 06:10:10'),
(12, 'Fiat', 'e9db84d0e11b5c26723e9951e4f7204b.jpg', '2019-09-18 06:10:42'),
(13, 'Toyoto', '2122001dca390ac45fd38a57fa8c51ea.png', '2019-09-18 06:11:20')

CREATE TABLE jl_car_payment (
    id              NUMBER(10) NOT NULL,
    bookingcar_id   NUMBER(10) NOT NULL,
    paymentdate     TIMESTAMP WITH TIME ZONE NOT NULL,
    paymentamount   NUMBER(10, 2) NOT NULL,
    paymentmethod   VARCHAR2(10) NOT NULL,
    cardnumber      NUMBER(16) NOT NULL,
    cardfname       VARCHAR2(50) NOT NULL,
    cardlname       VARCHAR2(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE jl_car_payment ADD CONSTRAINT jl_car_payment_pk PRIMARY KEY
( id, bookingcar_id );

CREATE TABLE jl_category (
    id             NUMBER(10) NOT NULL,
    categoryname   VARCHAR2(50) NOT NULL,
    updatedate     TIMESTAMP WITH TIME ZONE NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE jl_category ADD CONSTRAINT jl_category_pk PRIMARY KEY ( id );

INSERT INTO jl_category (id, categoryname, updatedate) VALUES
(1, Two Wheeler, '2019-09-17 11:18:40'),

(2, Four Wheeler, '2019-09-17 11:18:52');

CREATE TABLE jl_page (
    id                NUMBER(10) NOT NULL,
    pagetype          VARCHAR2(200) NOT NULL,
    pagetitle         VARCHAR2(200) NOT NULL,
    pagedescription   VARCHAR2(200) NOT NULL,
    email             VARCHAR2(120) NOT NULL,
    mobilenumber      NUMBER(10) NOT NULL,
    updationdate      TIMESTAMP WITH TIME ZONE NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE jl_page ADD CONSTRAINT jl_page_pk PRIMARY KEY ( id );
```

```
INSERT INTO jl_page (id, pagetype, pagetitle, pagedescription, email,
                        mobilenumber, updationdate) VALUES

(1, 'aboutus', 'About Us', 'vrms is an imaginary online business designed
by current graduate student Fredrick Li at NYU. The purpose of this localhost
website is to demonstrate different types of vehicles available for rental and
generate business transactions for possible industrial usage.',
NULL, NULL, '2021-11-29 11:29:48'),

(2, 'contactus', 'Contact Us', '795 Columbus Ave, New York, United States',
'jl7136@nyu.edu', 9178645818, '2021-12-14 19:24:55');

CREATE TABLE jl_twowheel_payment (
    id                      NUMBER(10) NOT NULL,
    bookingtwowheeler_id    NUMBER(10) NOT NULL,
    paymentdate             TIMESTAMP WITH TIME ZONE NOT NULL,
    paymentamount           NUMBER(10, 2) NOT NULL,
    paymentmethod           VARCHAR2(10) NOT NULL,
    cardnumber              NUMBER(16) NOT NULL,
    cardfname               VARCHAR2(50) NOT NULL,
    cardlname               VARCHAR2(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE jl_twowheel_payment ADD CONSTRAINT jl_twowheel_payment_pk PRIMARY KEY
( id, bookingtwowheeler_id );

CREATE TABLE jl_user (
    id              NUMBER(10) NOT NULL,
    firstname       VARCHAR2(120) NOT NULL,
    lastname        VARCHAR2(120) NOT NULL,
    email           VARCHAR2(120) NOT NULL,
    mobilenumber    NUMBER(10) NOT NULL,
    passcode        VARCHAR2(120) NOT NULL,
    regdate         TIMESTAMP WITH TIME ZONE NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE jl_user ADD CONSTRAINT jl_user_pk PRIMARY KEY ( id );

INSERT INTO jl_user (id, firstname, lastname, email, mobilenumber, passcode, regdate)
VALUES
(1, 'Rishi', 'Bhardwaj', 'rishi@gmail.com', 5646545645, '202cb962ac59075b964b07152d234b70',
'2019-09-21 07:13:47'),
(2, 'Meenakashi', 'Singh', 'singh@gmail.com', 9878775464, '202cb962ac59075b964b07152d234b70',
'2019-09-21 07:16:30'),
(3, 'Devesh', 'Gupta', 'devesh@gmail.com', 5656565656, '202cb962ac59075b964b07152d234b70',
```

```
'2019-09-21 07:17:00'),
(4, 'Abir', 'Malhotra', 'abir@gmail.com', 2556444557, '202cb962ac59075b964b07152d234b70',
'2019-09-21 07:17:35'),
(5, 'Perth', 'Kaushal', 'perth@gmail.com', 3232565690, '202cb962ac59075b964b07152d234b70',
'2019-09-21 07:18:10'),
(6, 'Shagun', 'Mishra', 'shagun@gmail.com', 7897978798, '202cb962ac59075b964b07152d234b70',
'2019-10-14 12:48:28'),
(7, 'Anuj', ' kumar', 'anc@test.com', 1234567809, 'f925916e2754e5e03f75dd58a5733251',
'2019-12-14 05:02:04');

CREATE TABLE jl_vehicle (
    id                   NUMBER(10) NOT NULL,
    vehiclename          VARCHAR2(120) NOT NULL,
    registrationnumber   VARCHAR2(120) NOT NULL,
    rentalprice          NUMBER(10, 2) NOT NULL,
    vehiclemodel         VARCHAR2(120) NOT NULL,
    vehicledescription   CLOB NOT NULL,
    seatingcapacity      NUMBER(2) NOT NULL,
    image                VARCHAR2(200) NOT NULL,
    image1               VARCHAR2(200),
    image2               VARCHAR2(200),
    image3               VARCHAR2(200),
    image4               VARCHAR2(200),
    image5               VARCHAR2(200),
    creationdate         TIMESTAMP WITH TIME ZONE NOT NULL,
    brand_id          NUMBER(10) NOT NULL,
    category_id       NUMBER NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE jl_vehicle
    ADD CONSTRAINT ch_inh_jl_vehicle CHECK ( jl_category_id IN (
        1,
        jl_vehiclevcar
    ) );

ALTER TABLE jl_vehicle ADD CONSTRAINT jl_vehicle_pk PRIMARY KEY ( id );

CREATE TABLE jl_vehiclevcar (
    id              NUMBER(10) NOT NULL,
    class           VARCHAR2(50) NOT NULL,
    fuel            VARCHAR2(50) NOT NULL,
    doors           VARCHAR2(50) NOT NULL,
    gearbox         VARCHAR2(50) NOT NULL,
    aircondition    VARCHAR2(50) NOT NULL,
    abs             VARCHAR2(50) NOT NULL,
```

```
    airbags         VARCHAR2(50) NOT NULL,
    bluetooth       VARCHAR2(50),
    carkit          VARCHAR2(50),
    gps             VARCHAR2(50),
    music           VARCHAR2(50),
    centerlocking   VARCHAR2(50)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE jl_vehiclevcar ADD CONSTRAINT jl_vehiclevcar_pk PRIMARY KEY ( id );

INSERT INTO jl_vehicle (id, vehiclename, registrationnumber, rentalprice,
vehiclemodel, vehicledescription, seatingcapacity, image, image1, image2,
image3, image4, image5, creationdate, brand_id, category_id) VALUES

(1, , 'Yamaha', 'Yamaha YZF R15 V3', 'DEL-5676', '500', '2018', '\"\"The
YZF-R15 changed the 150cc segment in the Indian market the way the CBZ
did when it was launched. It was an everyday motorcycle that could genuinely
be used as a trackday tool. The second version of the R15 traded practicality
for more focused performance, but the advent of the KTM RC200 meant that a far
better performance was available for the sportbike enthusiast at a similar price.
The R15 Version 3.0 reduces that gap significantly with technology. On the
list is an engine with a few more ccs, but featuring variable valve timing
which takes the maximum power output to nearly 19bhp with a nominal decrease
in torque which is now spread over a wider rev range. It also gets all-LED
lamps and an all-digital LCD instrument cluster that displays a wealth of
information, including when the Variable Valve Actuation switches to the
different camshaft profile.\"\"', 2,
'af457c29a6bce1b40f45386173b51064.jpg',
'efc1a80c391be252d7d777a437f868701568817086.jpg',
'cff8ad28cf40ebf4fbdd383fe546098d1568817086.jpg',
'19c10f4e66067da4b5eb1dac874e46721568817086.jpg',
'74375080377499ab76dad37484ee7f151568817086.jpg',
'b9fb9d37bdf15a699bc071ce49baea531568817086.jpg', NULL, 4, 1),

(2, 'Honda Activa 5G', 'DEL-7676876', '200', '2018', 'The very popular Honda
Activa 5G received a major update at the Auto Expo 2018 in February, and now
the company has listed the scooter on its website at a starting price of
Rs 52,460 (ex-showroom) for the standard variant and Rs 54,325 (ex-showroom)
for the Deluxe variant. Bookings were already underway for the Activa 5G since
it was unveiled in February. The 2018 Honda Activa 5G comes with subtle changes
to its design with all-LED headlamp with a position lamp along with new color
options and minor changes to its styling. The mechanics of the Activa 5G,
however, remains the same.', 2, '31bc5410ae14c6f60a47e9edd913b4d7.jpg',
'941e62e1e18f45f7a4108c7bc692bc9e.jpg',
'e0ac3c5d2c5bcd60234e29de9477328e1568868021.jpg',
```

```
'553d1f568dcda350727b015978f2456e1568868021.jpg',
'f490adaea0d324c2fc638b4162af61291568868021.jpg',
'fbd8be3d214221f1a4394fbfac8064ca1568868021.jpg', NULL, 1, 1);

(3, 'Hyundai Elite i20', 'Del-7889', '500', '2019', '\"Popular Korean
car manufacturer, Hyundai has updated the Elite i20 range in India.
The premium hatchback is available in four variants - Era, Magna Plus,
Sportz Plus and Asta (O). The latest model continues to be powered
by existing petrol and diesel engine options while changes are limited
to cosmetic upgrades.  The Hyundai Elite i20 competes against the
Maruti Suzuki Baleno, the Honda Jazz and the Volkswagen Polo in the
premium hatchback segment in the Indian car market.\"', 5, 'Compact',
'Petrol and Diesel', '5', 'Manual and Automatic (CVT)', 'No', 'Yes',
'Yes', 'Yes', 'Yes', 'Yes', 'Yes', 'Yes',
'85521b836186ca3632770483a48a10db.jpg',
'85521b836186ca3632770483a48a10db.jpg',
'1fbf40517cdc7517629e1b62e15f0b161568964746.jpg',
'dab1051b997261c9f130a15c2bc6fe8f1568964746.jpg',
'fc6c1f4a82dc3123554208854de80ffa1568964746.gif',
'91d295aa31f046a4aeb48704d6b94a1c1568964746.jpg',
'2019-09-20 07:32:26', 7, 2),

(4, 'Renault Kwid', 'Del-78907', '600', '2019', 'The entry-level
product from Renault India, the Kwid, is available at an extremely
competitive price and offers a good blend of practicality and efficiency.
Named after the concept car that was shown at the 2014 Auto Expo,
the Kwid finds itsâ€™ underpinnings on the Renaultâ€™s CMFA platform
and is being produced at their Chennai factory.', 5, 'Compact',
'Petrol', '5', 'Manual and AMT', 'Yes', 'Yes', 'Yes', 'Yes', 'Yes',
'Yes', 'Yes', 'Yes',
'63c5ee8e7434754583cb02196d07ad141568965368.jpg',
'938077061daafd39ed8fb52ae810f2311568965368.jpg',
'938077061daafd39ed8fb52ae810f2311568965368.jpg',
'1ff95e755485face77a49fe8e09a3d161568965368.jpg',
'012228c710a8fc6500c92a7369d639a41568965368.jpg',
'63c5ee8e7434754583cb02196d07ad141568965368.jpg',
'2019-09-20 07:42:48', 8, 2);


ALTER TABLE jl_car_payment
    ADD CONSTRAINT bookingcar_fk FOREIGN KEY ( bookingcar_id )
        REFERENCES jl_bookingcar ( id );

ALTER TABLE jl_bookingcar
    ADD CONSTRAINT jl_bookingcar_jl_user_fk FOREIGN KEY ( jl_user_id )
```

```
        REFERENCES jl_user ( id );

ALTER TABLE jl_bookingcar
    ADD CONSTRAINT jl_bookingcar_jl_vehicle_fk FOREIGN KEY ( jl_vehicle_id )
        REFERENCES jl_vehicle ( id );

ALTER TABLE jl_twowheel_payment
    ADD CONSTRAINT jl_bookingtwowheeler_fk FOREIGN KEY ( bookingtwowheeler_id )
        REFERENCES jl_bookingtwowheeler ( id );

ALTER TABLE jl_bookingtwowheeler
    ADD CONSTRAINT jl_user_fk FOREIGN KEY ( jl_user_id )
        REFERENCES jl_user ( id );

ALTER TABLE jl_vehicle
    ADD CONSTRAINT jl_vehicle_jl_brand_fk FOREIGN KEY ( jl_brand_id )
        REFERENCES jl_brand ( id );

ALTER TABLE jl_vehicle
    ADD CONSTRAINT jl_vehicle_jl_category_fk FOREIGN KEY ( jl_category_id )
        REFERENCES jl_category ( id );

ALTER TABLE jl_vehiclevcar
    ADD CONSTRAINT jl_vehiclevcar_jl_vehicle_fk FOREIGN KEY ( id )
        REFERENCES jl_vehicle ( id );

ALTER TABLE jl_bookingtwowheeler
    ADD CONSTRAINT vehiclevcar_fk FOREIGN KEY ( jl_vehiclevcar_id )
        REFERENCES jl_vehiclevcar ( id );


COMMIT;
```

4 Schema

The following page shows both logical and relational models:

Figure 1: Logical Model

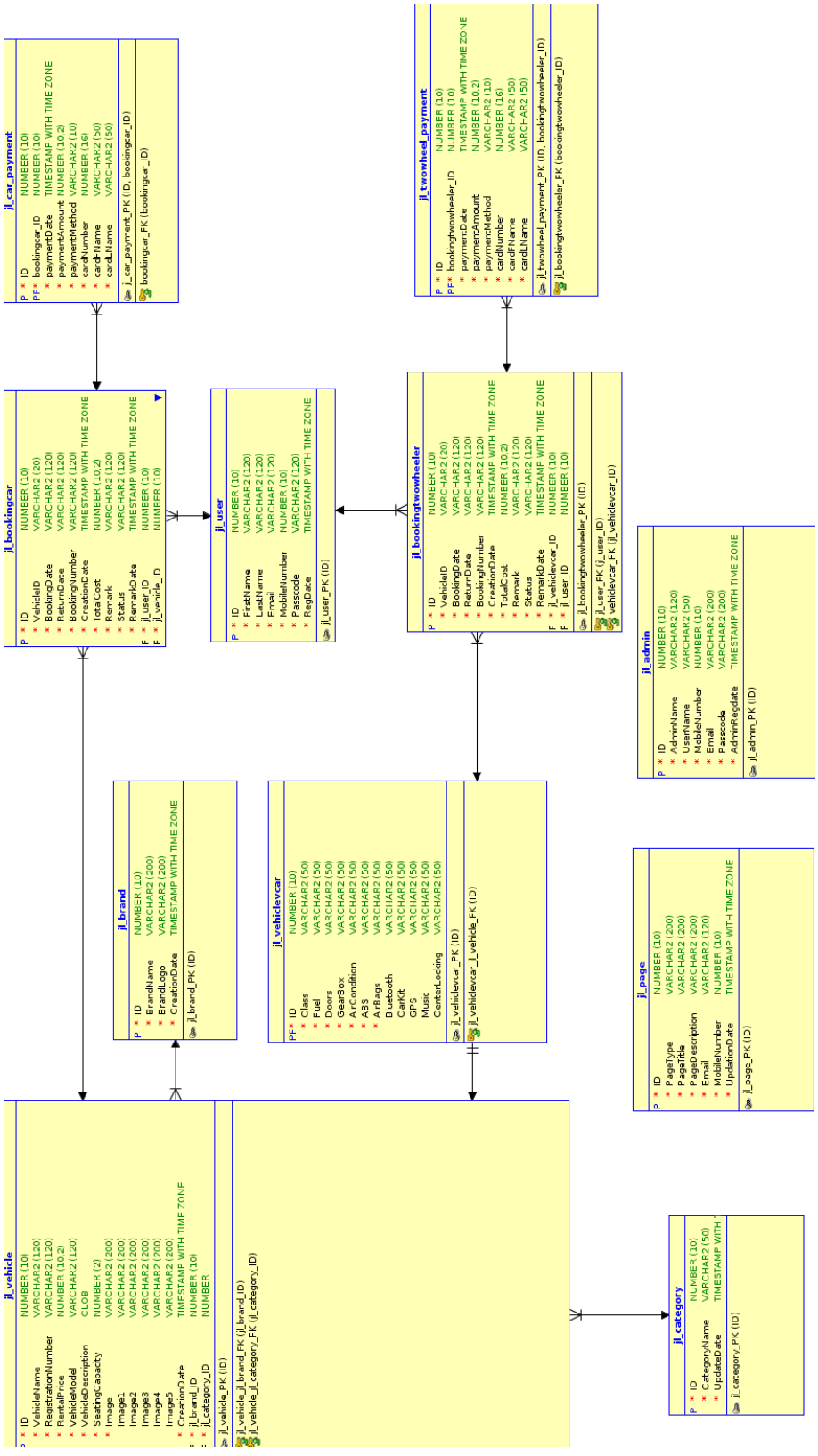

Figure 2: Relational Model

There are total of 11 tables in the databse model and some are pre-populated with initial data. The following table shows detailed description:

Table Name	Number of Entries
jl_admin	1
jl_user	7
jl_page	2
jl_category	2
jl_vehicle	4
jl_vehiclecar	2
jl_bookingcar	0
jl_bookingtwowheeler	0
jl_brand	13
jl_twowheel_payment	0
jl_car_payment	0

5 Screenshots

Below are several screenshots of user/admin interface and functionalities:

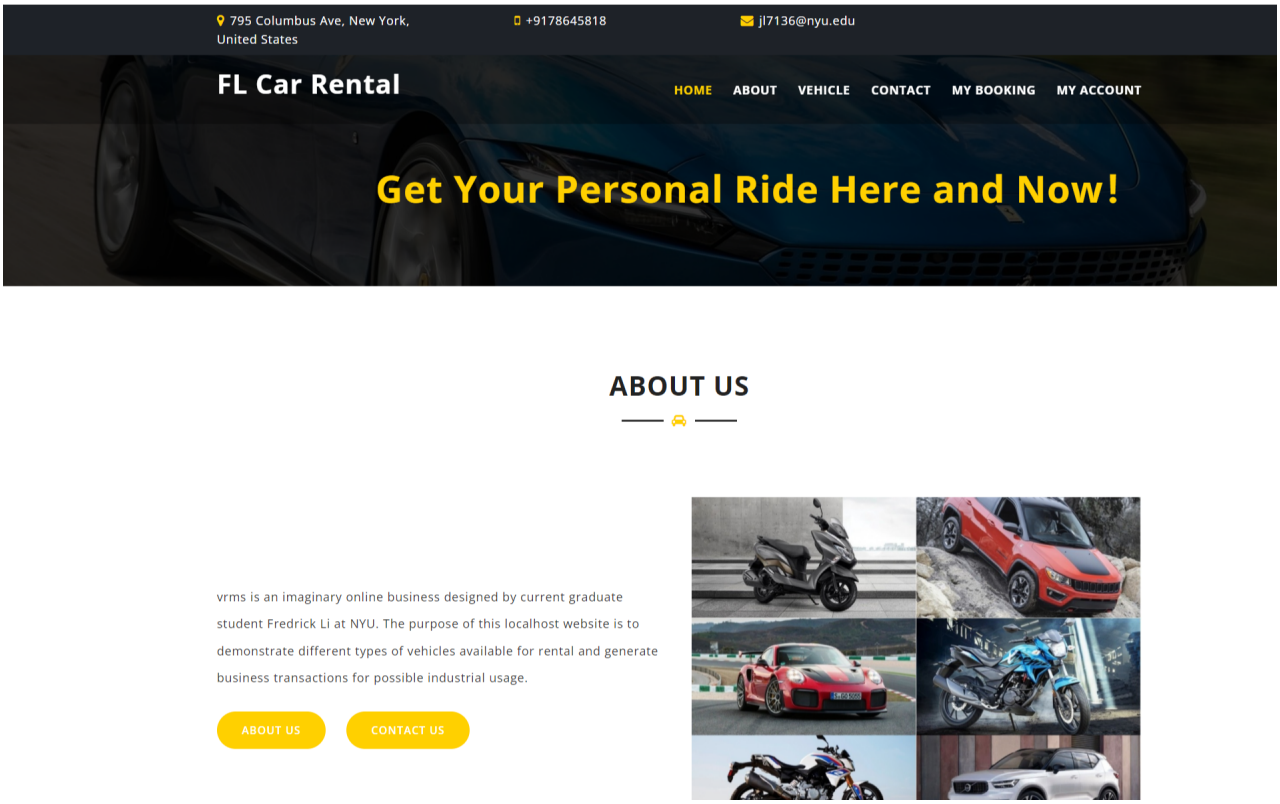

Figure 3: Home Page

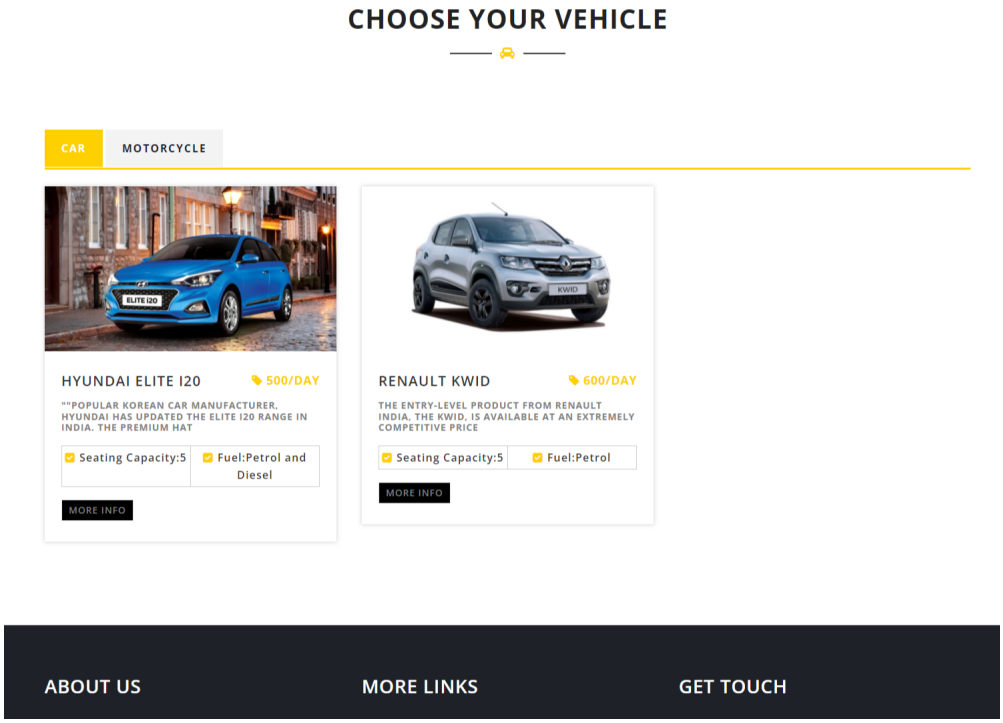

Figure 4: Home Page

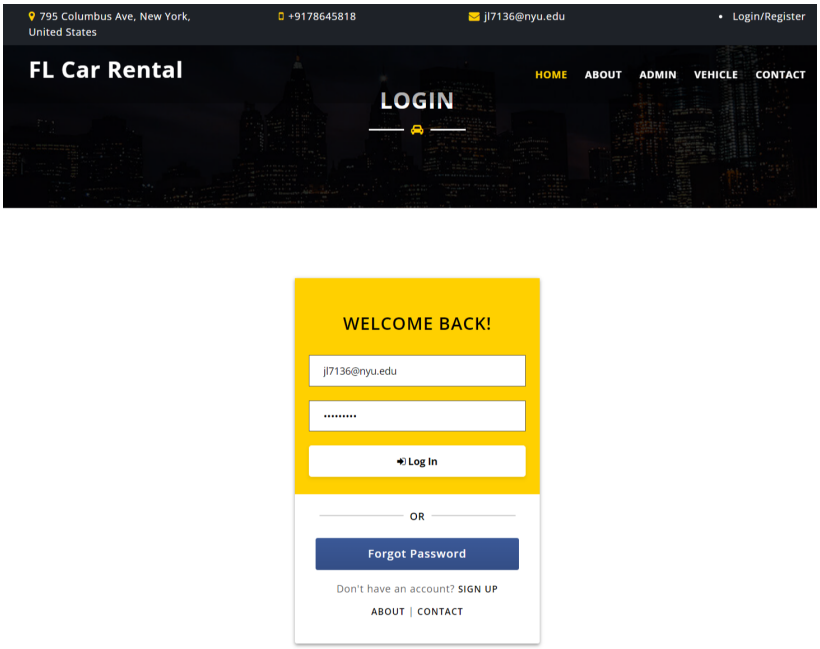

Figure 5: User Login

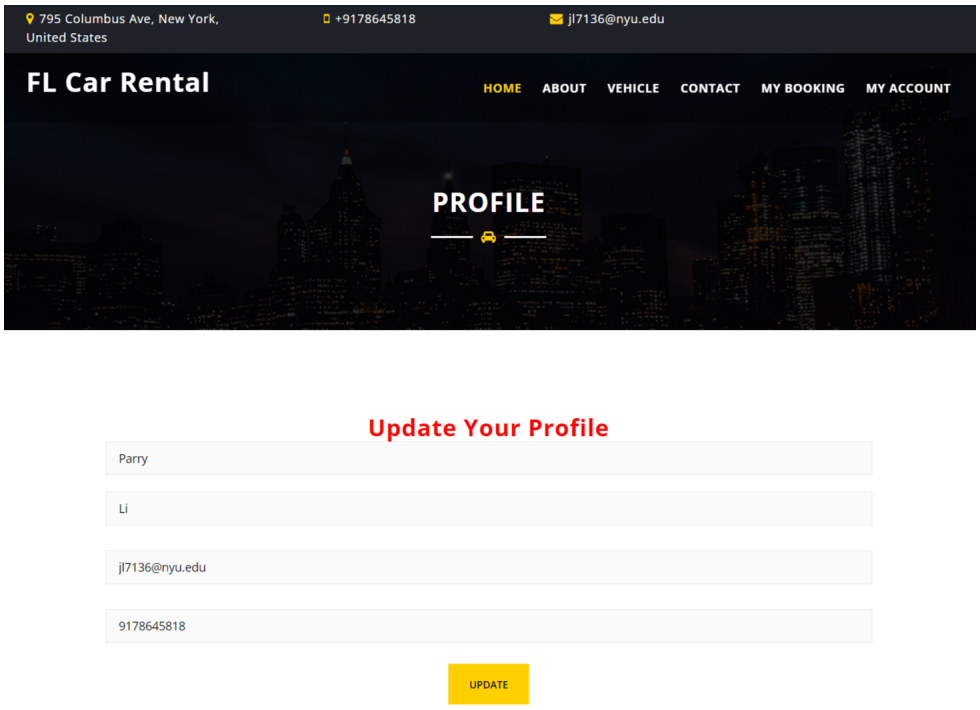

Figure 6: User Profile

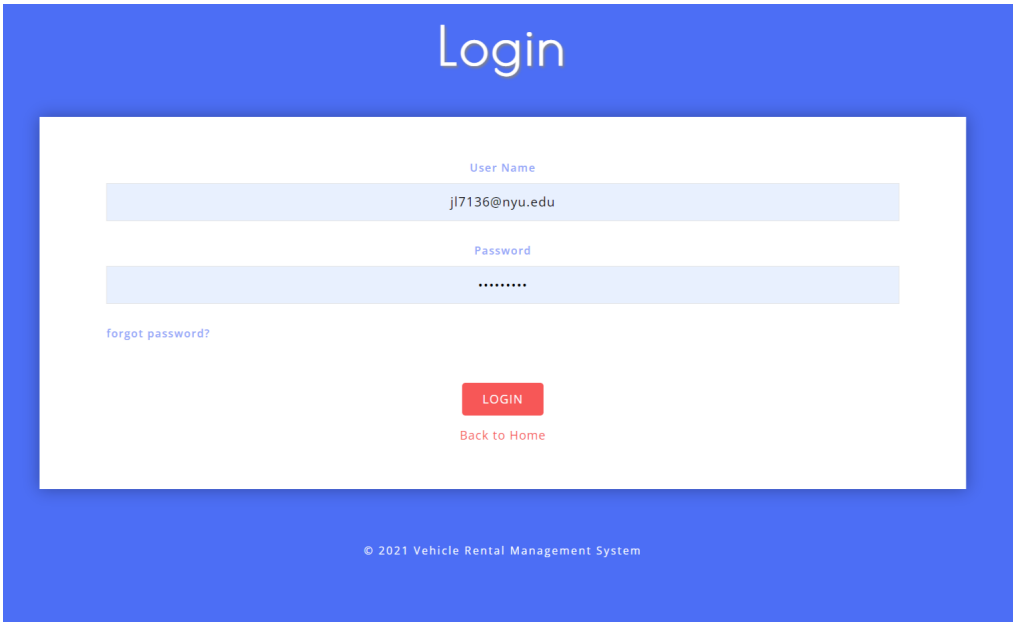

Figure 7: Admin Login

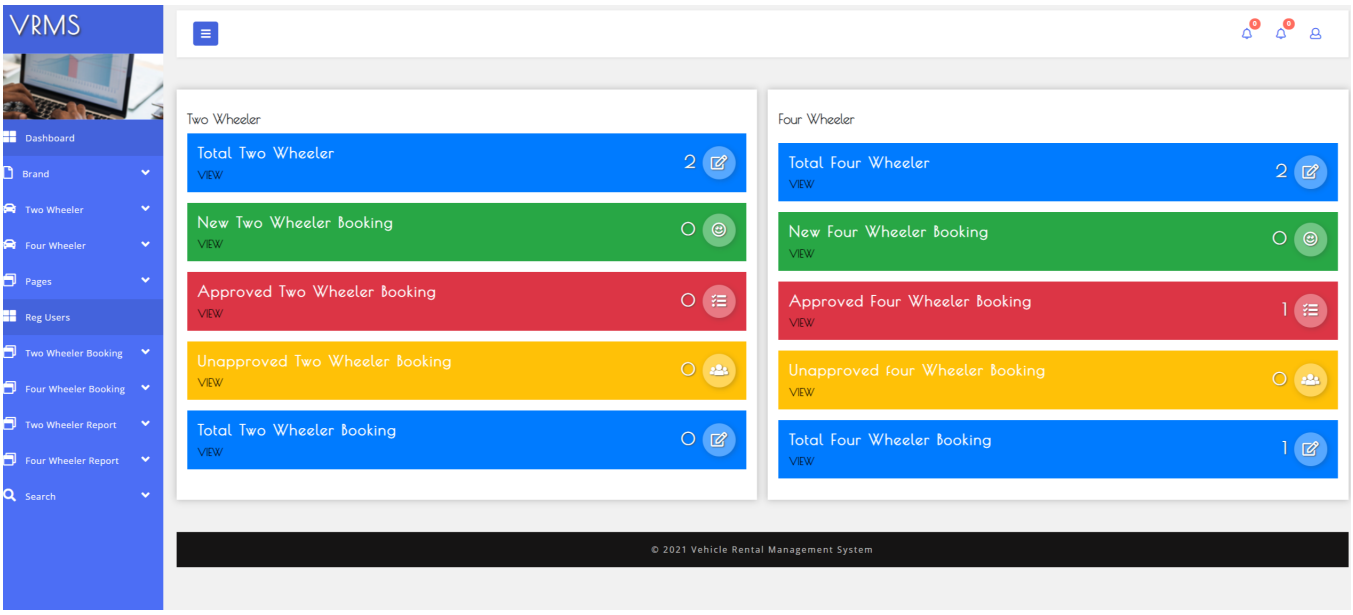

Figure 8: Admin Dashboard

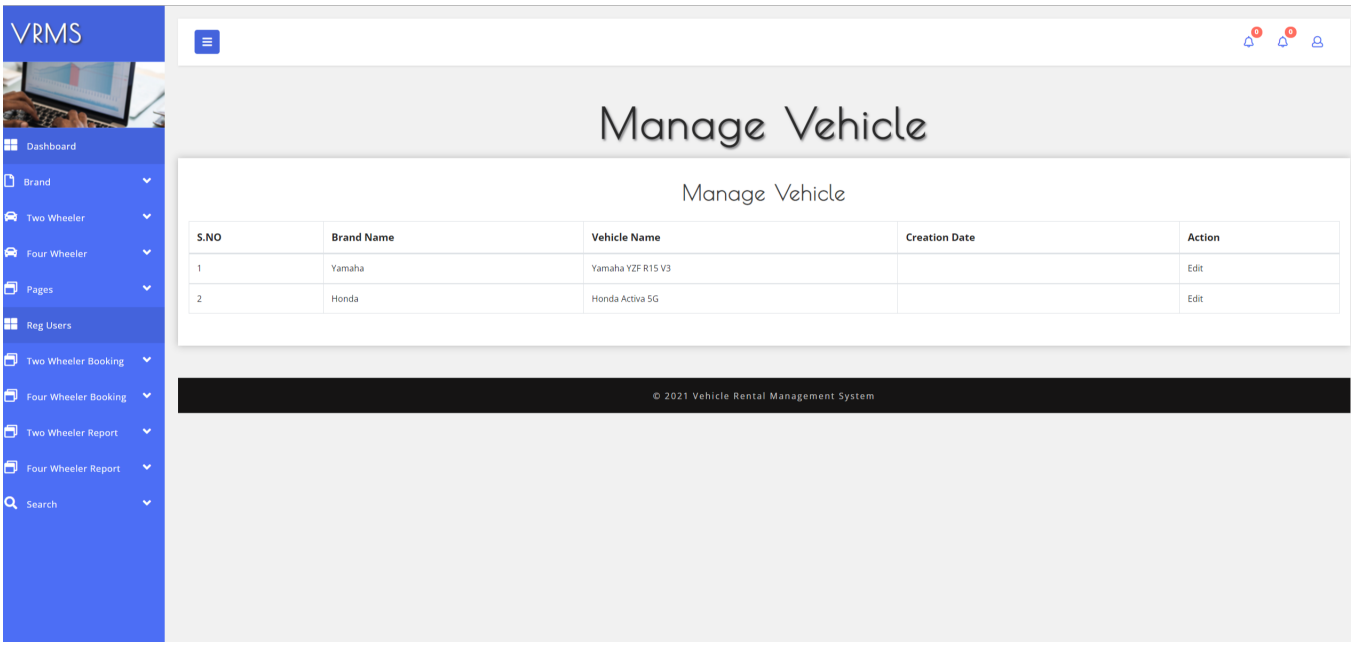

Figure 9: Vehicle Manage Page

6 Security Implementation

One of the greatest and yet most prevalent threats to web application development is SQL Injection. What SQL injection does is to use malicious SQL code inserted into URL for backend database manipulation, and could, in result, access and most likely compromise sensitive information.

There are many frequently adopted and powerful tools like parameterized queries and stored procedures that could be used against SQL injection, but in this project, I utilized a simple, low-cost, yet quite effective approach to prevent SQL injection attacking: User Data Escaping. Since the basic idea of SQL injection is to confuse DBMS between user input and SQL code, but if we escape all user input, this confusion would be impossible to arise. Let's look at an example:

```
$adminuser=$_POST['username'];
$password=$_POST['password'];
$query=mysqli_query($con,"select ID from jl_admin where
UserName='$adminuser' && Password='$password' ");

$ret=mysqli_fetch_array($query);
```

This block of code is the insecure version of login checking part for admin users. Note that the user inputs username and password are incorporated into the query conditions, and what hackers can do is to type specific SQL condition codes as username and password to circumvent credential checking, for example:

```
UserName: admin OR 1=1'
Password: admin OR 1= 1'
```

Now the query condition becomes:

```
$query=mysqli_query($con,"select ID from jl_admin where
UserName='admin OR 1=1' && Password='admin' OR 1=1'");
```

Since 1=1 is always true, the SQL query would always execute and the where clause is essentially useless. As a result, the hacker would successfully bypass the login checking. Now let's see how escaping user input works to prevent this from happening:

```
$adminuser=mysqli_real_escape_string($con, $_POST['username']);
$password=mysqli_real_escape_string($con,$_POST['password']);
$query=mysqli_query($con,"select ID from jl_admin where
UserName='$adminuser' && Password='$password' ");

$ret=mysqli_fetch_array($query);
```

I added a built-in escaping function *mysqli_real_escape_string*() for all user inputs, so that the variables called in the query condition will be always treated as a string instead of SQL codes, thus preventing from any outside manipulation.

Another security implementation is using MD5 function to ensure password integrity. MD5 (message-digest algorithm) is a hash function developed for cryptography, and what it dose is to convert user password into a 128-bit hash value. Though MD5 is as encryption algorithm, it has many weakness in security, but still suitable for checking partitioned database integrity due to its low computation cost and fast algorithm speed. An example is as following:

```
$password=md5(mysqli_real_escape_string($con, $_POST['password']));
```

7 Reflection

Developing a web-based database management system as an individual project can be challenging and throughout the whole process, it is indeed. Before beginning this project, my original expectations on possible sources challenges are those from physical database design: a properly functioning relational model; while designing an easy-to-manage database is difficult, the most challenging part of the design process actually emerges from developing web interface through API, and the difficulties are two-folded.

PHP as a scripting language is easy to learn and can be readily incorporated into HTML documents, thus gives me abundant time to design a much better and realistic-looking user interface with various css plugins. However, as I was trying to add more functions and page hyperlinks, the code volume and complexity started to get out of control. For example, the administrator of the website should have the access to manage and modify almost all information available in the database system, but in my particular cases, there are many attributes of tables regarding information of cars and booking logs, and the vehicle itself also has subtypes with even more attributes; in the "edit_fourwheelevehicle.php" file within the admin interface, if the administrator wishes to change the information of a particular type of car, the code length to implement and validate the update submission form is over 200 lines because there are 26 fields that can be updated, and one can easily make loads of mistakes during coding. As I proceeded with designing functionalities of the admin interface, I decided to drop several functions like querying count reports, payment options, and sales summary, etc. to make the system more manageable. Nevertheless, in reality, this final version of vehicle rental system is still far from application.

Another major obstacle is reconciling the database design and website's actual needs as more functions are added. For instance, when I was coding the booking schedule module, my original design blueprint has a constraint check on available rental slots for all vehicles. To do that, I should include an entity called "schedule" that stores all information of existing booked time-slots to prevent from overlapped booking time. However, this database design would become too complicated to maintain if there are, say, more than 1000 cars in inventory. Fortunately, I was able to streamline the design and made simplifications on the original database design in project part one. what I did was to add a query condition when the user is submitting a booking confirmation:

```
bdate=_POST['bookingdate'];
rdate=_POST['returndate'];
vid=_GET['viewid'];

query_check=mysqli_query(con,
SELECT * FROM jl_bookingcar where
(bdate BETWEEN date(BookingDate) and date(ReturnDate) ||
rdate BETWEEN date(BookingDate) and date(ReturnDate) ||
date(BookingDate) BETWEEN bdate and rdate) and
Vehicle_ID=vid);

num=mysqli_num_rows(query_check);

if num != 0:
   echo <script>alert("Vehicle not available on these days")</script>;
```

In this way, I managed to ensure the proper functioning of the booking system while maintaining the simplicity of the database design.

Though there were lots of difficulties of designing a proper functioning web-based management system, I was able to pick up different development toolkit and kept pushing myself to perfection. One notable feature that I added to this project is sidebar plugins realized through css that enables really beautiful appearance of the website. I spent time digging into the extensive use of Cascading Style Sheets language, one of the cornerstone languages of web development, so that I can make a real website instead of a merely functioning one.

In conclusion, the time and efforts invested in this project allowed me to reflect on the whole web-developing process in first-hand experience. Though designing relational database stands on its own, one should also constantly revisit the design in making sure that the database can meet the requirements of predetermined web functionalities and make adjustments accordingly.

8 Queries

8.1 Table Join

```
SELECT
  jl_bookingtwowheeler.BookingDate AS Booking_Date,
  jl_user.FirstName AS First_Name,
  jl_user.LastName AS Last_Name,
  jl_vehicle.BrandName AS Brand_Name
FROM jl_bookingtwowheeler
JOIN jl_user ON jl_bookingtwowheeler.Userid=jl_user.ID
JOIN jl_vehicle ON jl_vehicle.ID= jl_bookingtwowheeler.VehicleID;
```

Booking_Date	First_Name	Last_Name	Brand_Name
2021-12-17	alice	kled	Yamaha
2021-12-31	alice	kled	Honda
2022-01-18	ashe	king	Yamaha

Figure 10: Join 3 Tables

What the query returns is the list of total bookings for motors. Four columns are representing booking starting date, user's first name, user's last name, and motor's brand name.

8.2 Multi Row Subquery

```
SELECT jl_bookingcar.ID AS Car_Booking_ID, jl_bookingcar.TotalCost
AS Total_Cost, jl_bookingcar.VehicleID AS Vehicle_ID
FROM jl_bookingcar
WHERE VehicleID IN(
 SELECT ID FROM jl_vehiclecar
 WHERE BrandName='Hyundai')
AND jl_bookingcar.Status = 'Approved';
```

Car_Booking_ID	Total_Cost	Vehicle_ID
1	1500	1
2	2500	1
7	1500	1

Figure 11: Multi Row Subquery

What the query returns is the list of approved car bookings that specifically for car brand Hyundai. Three rows represent booking ID, total rental cost and vehicle ID.

8.3 Correlated Subquery

```
SELECT a.Userid, a.TotalCost, b.BrandName
 FROM jl_bookingcar a
 JOIN jl_vehiclecar b ON a.VehicleID = b.ID
```

```
WHERE a.TotalCost > (SELECT AVG(jl_bookingtwowheeler.TotalCost)
                FROM jl_bookingtwowheeler
                WHERE jl_bookingtwowheeler.Status = 'Approved');
```

Userid	TotalCost	BrandName
5	2500	Hyundai
6	2400	Renault
10	7800	Renault
10	16000	Hyundai

Figure 12: Correlated Subquery

The above correlated subquery returns the user ID, Total rental cost, and brand name of approved car bookings that costs more than the average rental cost for motor booking.

```
SELECT AVG(tblbookingtwowheeler.TotalCost) AS Average_Cost_Motor
     FROM tblbookingtwowheeler
     WHERE tblbookingtwowheeler.Status = 'Approved';
```

Average_Cost_Motor
1760

Figure 13: Average Cost of Motor Booking

Above outputs the average cost of motor booking which is $1760 , clearly less than all 4 entries returned by the correlated query.

8.4 Set Operator

```
SELECT tblbookingtwowheeler.Location AS Pickup_Location,
    tblbookingtwowheeler.BookingDate As Booking_Date
FROM tblbookingtwowheeler
WHERE tblbookingtwowheeler.Location = 'ca'

UNION
SELECT tblbookingcar.Location AS Pickup_Location,
tblbookingcar.BookingDate As Booking_Date
```

```
FROM tblbookingcar
WHERE tblbookingcar.Location = 'ca';
```

Pickup_Location	Booking_Date
ca	2021-12-17
ca	2021-12-31
ca	2022-01-18
ca	2021-12-18
ca	2021-12-30
ca	2022-01-28

Figure 14: Set Operator on Pickup Location

Above SQL query returns all booking's dates where the pickup location is in California.

8.5 Top-N Query

```
SELECT tblbookingcar.Userid AS User_ID,
    tblbookingcar.TotalCost AS Total_Cost
FROM tblbookingcar

ORDER BY tblbookingcar.TotalCost DESC LIMIT 5;
```

User_ID	Total_Cost
10	16000
10	13200
10	7800
5	2500
6	2400

Figure 15: Top 5 Highest Car Booking Cost