



# UNIVERSITY OF MALAYA

## WOC7014

### FRAMEWORK-BASED SOFTWARE DESIGN AND DEVELOPMENT

### Final Alternative Assessment

<b>Name:</b>	Kaha Mohamud Mohamed Haid
<b>Matric Number:</b>	23069029

**Question 1:**

In this question four models will be created and designed for CarHub Online Showroom which are:

**1. User Model:**

1.	username
2.	password
3.	email
4.	is_dealer
5.	is_customer
6.	first_name
7.	last_name
8.	date_joined

**2. Dealer Model:**

1.	user (OneToOneField to User)
2.	image (ImageField, upload_to='dealer_images/', blank=True, null=True)
3.	email (EmailField)

**3. Car Model:**

1.	dealer (ForeignKey to Dealer)
2.	name (CharField, max_length=100)
3.	model (CharField, max_length=100)
4.	year (IntegerField)
5.	price (DecimalField, max_digits=10, decimal_places=2)
6.	image (ImageField, upload_to='car_images/')

**4. Wish List Model:**

1.	user (ForeignKey to User)
----	---------------------------

2.	car (ForeignKey to Car)
----	-------------------------

Below the picture will explain tables, fields, and fields' datatype through code.

```
main > models.py > ...
1  from django.contrib.auth.models import AbstractUser
2  from django.db import models
3
4  class User(AbstractUser):
5      is_dealer = models.BooleanField(default=False)
6      is_customer = models.BooleanField(default=False)
7
8  class Dealer(models.Model):
9      user = models.OneToOneField(User, on_delete=models.CASCADE)
10     image = models.ImageField(upload_to='dealer_images/', blank=True, null=True)
11     email = models.EmailField()
12
13     def __str__(self):
14         return self.user.username
15
16     class Car(models.Model):
17         dealer = models.ForeignKey(Dealer, on_delete=models.CASCADE)
18         name = models.CharField(max_length=100)
19         model = models.CharField(max_length=100)
20         year = models.IntegerField()
21         price = models.DecimalField(max_digits=10, decimal_places=2)
22         image = models.ImageField(upload_to='car_images/')
23
24         def __str__(self):
25             return self.name
26
27     class Wishlist(models.Model):
28         user = models.ForeignKey(User, on_delete=models.CASCADE)
29         car = models.ForeignKey(Car, on_delete=models.CASCADE)
30
31         def __str__(self):
32             return f"{self.user.username} - {self.car.name}"
33
```

### Question 2:

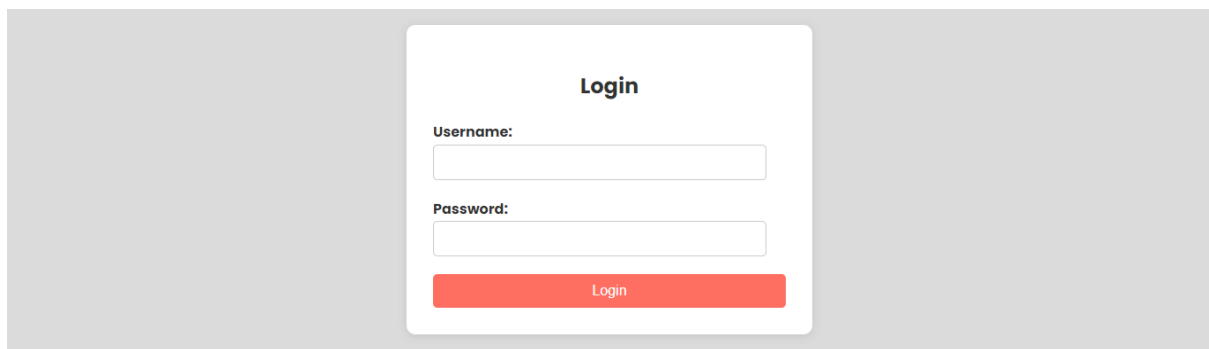
In this question it asked to display templates that will be used for CarHub, *the base template will be created with header and footer* ( © CarHub – Kaha 23069029 ). Other templates are displayed below:

1.	Car List
2.	Car Detail
3.	Dealer List
4.	Add Car
5.	Login
6.	Dealer Detail

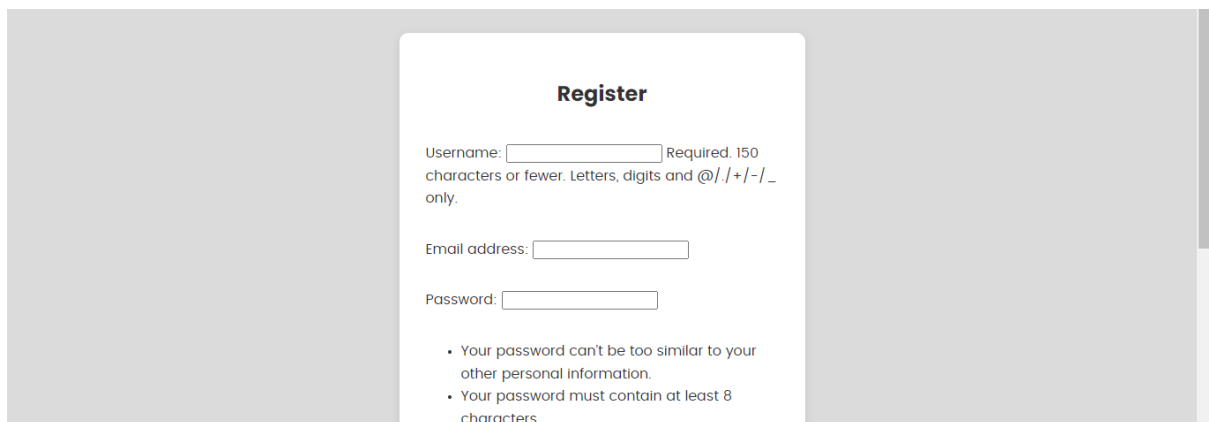
7.	Welcoming Page
8.	Register
9.	Manage Cars
10.	Update Profile
11.	Wish List

Here is the designed UI for better understanding, you can view the code through attached folder in Specturm:

### Login and Register:



The image shows a login form centered on a light gray background. The form is a white rounded rectangle with the title "Login" in bold black text at the top. Below the title, there are two labels: "Username:" and "Password:". Each label is followed by a white rectangular input field with a thin gray border. At the bottom of the form is a solid red rectangular button with the word "Login" in white text.



The image shows a register form centered on a light gray background. The form is a white rounded rectangle with the title "Register" in bold black text at the top. Below the title, there are three labels: "Username:", "Email address:", and "Password:". Each label is followed by a white rectangular input field with a thin gray border. Below the "Username:" input field, there is a line of text: "Required. 150 characters or fewer. Letters, digits and @/./+/-/\_ only." Below the "Password:" input field, there is a bulleted list of two requirements: "Your password can't be too similar to your other personal information." and "Your password must contain at least 8 characters."

other personal information.

- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

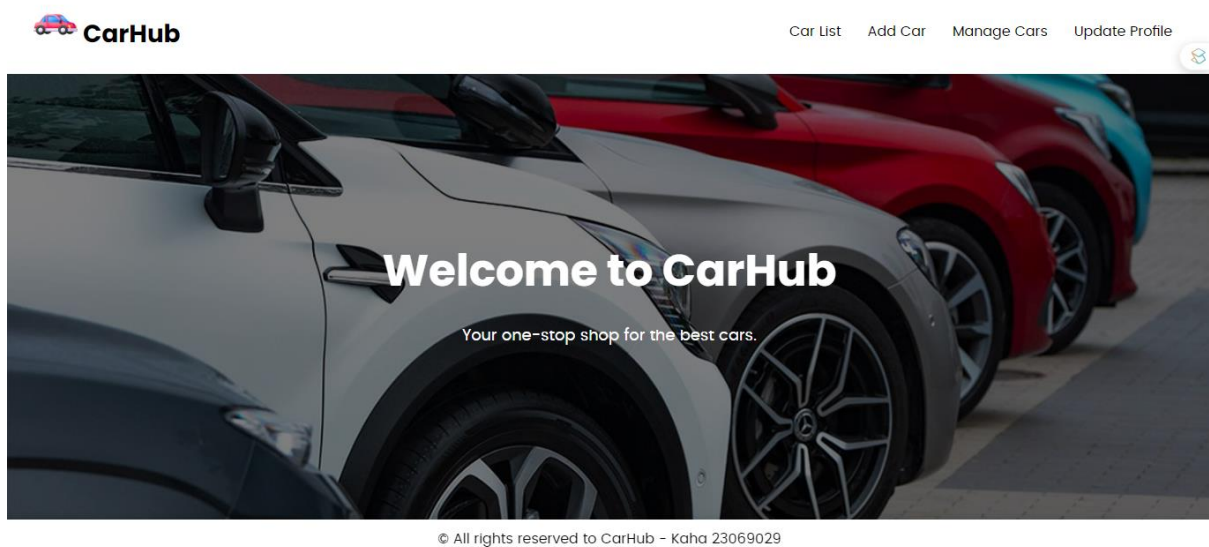
Password confirmation:   
Enter the same password as before, for verification.

Is dealer: ☐

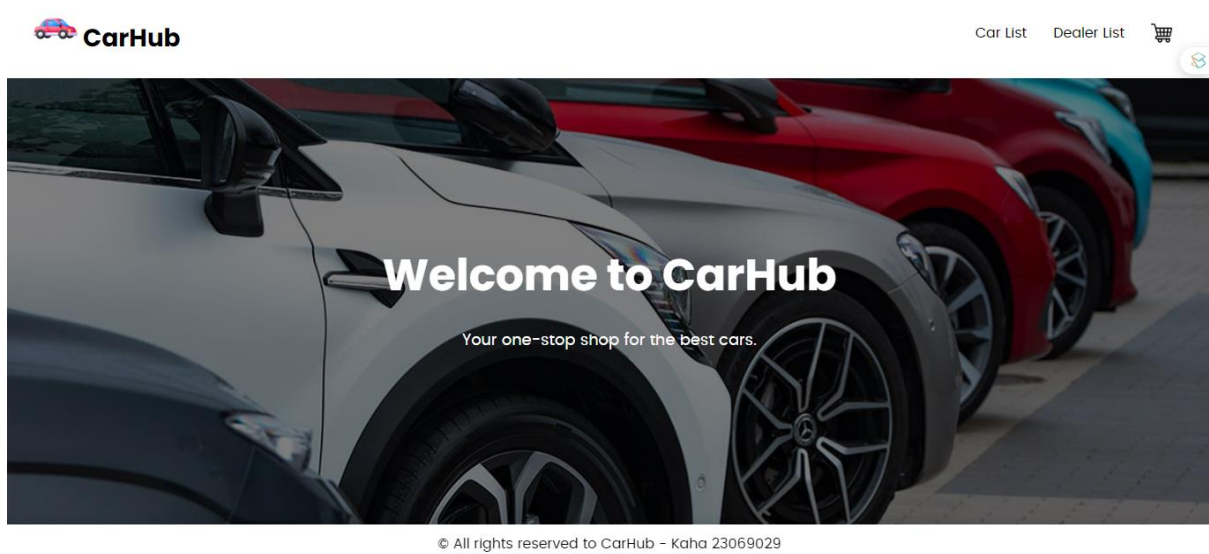
Is customer: ☐

Register


## Welcome Dealer:



## Welcome Customer:



Car List Dealer:

 CarHub

Car ListAdd CarManage CarsUpdate Profile


LIST OF AVAILABLE CARS

Search Car/Model...

Min price

Max price


Search



Honda

Model: Civic


Year: 2024



Honda

Model: Civic

Year: 2023




Mercedes-Benz


Model: S-Class

Year: 2024

© All rights reserved to CarHub - Kaha 23069029

 CarHub

Car ListAdd CarManage CarsUpdate Profile



Honda


Model: Civic

Year: 2024

Price: \$23950.00

Dealer: Wonchang

View Details



Honda


Model: Civic

Year: 2023

Price: \$22400.00

Dealer: Wonchang

View Details



Mercedes-Benz

Model: S-Class

Year: 2024


Price: \$110100.00

Dealer: Wonchang

View Details

© All rights reserved to CarHub - Kaha 23069029

## Add Car:



Car ListAdd CarManage CarsUpdate Profile

Add Car


Name:

Model:

Year:

Price:

© All rights reserved to CarHub - Kaha 23069029



Car ListAdd CarManage CarsUpdate Profile

Name:

Model:

Year:

Price:


Image:  No file chosen

Description:  


Add Car

© All rights reserved to CarHub - Kaha 23069029


## Manage Cars:

 **CarHub**


Car ListAdd CarManage CarsUpdate Profile



**Honda**  
Model: Civic  
Year: 2024  
Price: \$23950.00  
Dealer: Wonchang  
Update Delete




**Honda**  
Model: Civic  
Year: 2023  
Price: \$22400.00  
Dealer: Wonchang  
Update Delete



**Mercedes-Benz**  
Model: S-Class  
Year: 2024  
Price: \$110100.00  
Dealer: Wonchang  
Update Delete

© All rights reserved to CarHub - Kaha 23069029

## Update Profile:

 **CarHub**


Car ListAdd CarManage CarsUpdate Profile

### Update Profile

**Username:**

**Email:**


**Profile Image:**  
 No file chosen




© All rights reserved to CarHub - Kaha 23069029





Car List Customer:

 CarHub

Car ListDealer List







Honda

Model: Civic


Year: 2024

Price: \$23950.00

Dealer: Wonchang

View Details

Add to Wishlist



Honda

Model: Civic


Year: 2023

Price: \$22400.00

Dealer: Wonchang

View Details

Add to Wishlist



Mercedes-Benz

Model: S-Class

Year: 2024

Price: \$110100.00


Dealer: Wonchang

View Details


Add to Wishlist


© All rights reserved to CarHub - Kaha 23069029


View Car Detail:

 CarHub

Car ListDealer List







Model: Civic

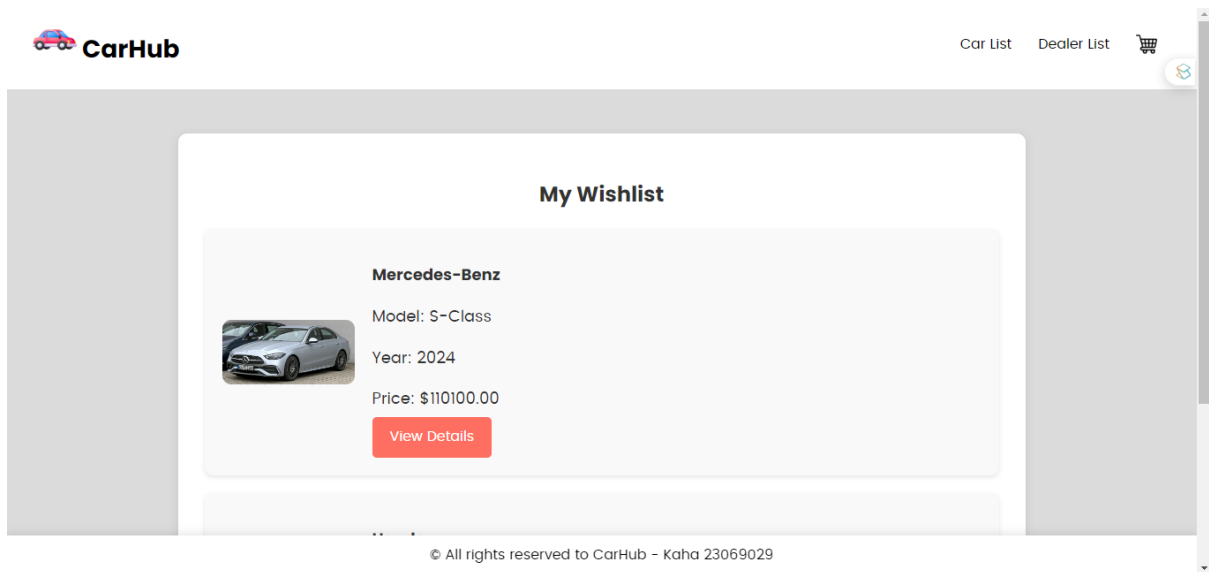
Year: 2024

Price: \$23950.00

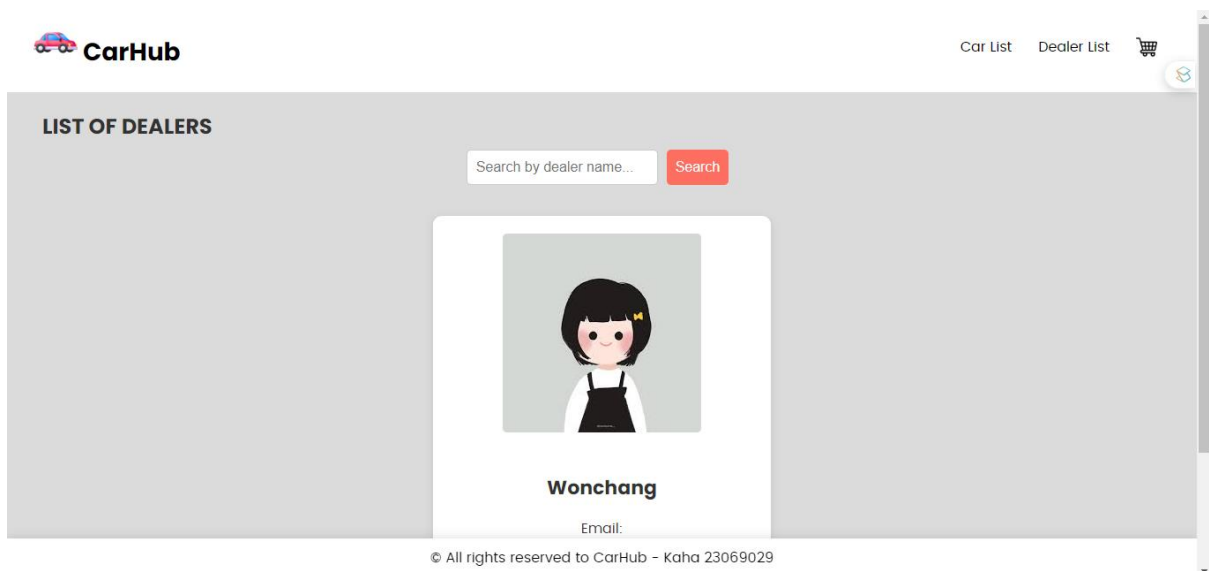
Dealer: Wonchang

© All rights reserved to CarHub - Kaha 23069029

## Wishlist (Also Accessible through cart icon in header):



## Dealer List:





**Wonchang**

Email:

Selling: Honda, Honda, Mercedes-Benz

[View Details](#)

## Dealer Detail:



**Wonchang**



Email: wonchang@gmail.com

**Cars Selling:**



Email: wonchang@gmail.com

**Cars Selling:**

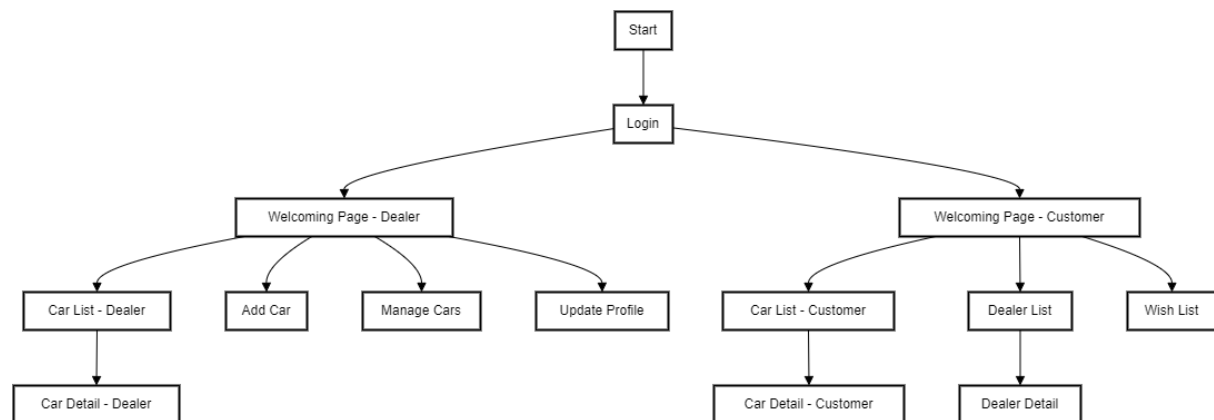
Honda - Civic - \$23950.00

Honda - Civic - \$22400.00

Mercedes-Benz - S-Class - \$110100.00

### Question 3:

Below is the is the navigation diagram to showcase the flow of the CarHub website



### Justification and explanation:

There are two user types - dealers and customers. Dealers will land on the welcoming page after logging in, where they can access the Car List and Add Car pages in the header. The Car List page displays all cars listed by dealers on the website, and the Add Car feature allows dealers to upload new cars for sale. The Car Detail page provides detailed information about individual cars where they can access it through Car List by clicking on one car. They can also Manage their cars through Manage Cars in the header as well as updating their profiles.

Customers also land on the welcoming page after logging in, where they can access the Car List, Wish List and Dealer List. The Car List page displays all cars listed by dealers, and clicking on a specific car directs them to the Car Detail page they can also add the car into the wish list. The Dealer List page allows customers to view information about different dealers on the website.

#### Question 4:

**A.** No, I will not split into multiple apps for two reasons:

**1. Project Size and Scope:** Given the current size and scope of the given website project, managing it within a single Django app is better and practical. The functionalities I'm implementing are cohesive and interrelated, making it efficient to keep them unified under one app.

**2. Simplicity and Maintainability:** Choosing for a single app structure simplifies my development process significantly. It streamlines code maintenance, reduces complexity, and enhances overall maintainability.

My plan is to work on a single app.py (main) and this is the outcome:

```
main > apps.py > ...
1  from django.apps import AppConfig
2
3
4  class MainConfig(AppConfig):
5      default_auto_field = 'django.db.models.BigAutoField'
6      name = 'main'
```

**B.** No, I have not used a RESTful API in this project. Instead, I used traditional Django views and templates to render HTML pages and handle form submissions. RESTful APIs typically involve JSON responses and specific endpoints, which I did not implement here. The project focused on server-rendered HTML suitable for a multi-page application.

**C.** For this project, the best testing procedures are:

#### 1. Unit Testing:

- Tests individual components to ensure each part works independently.
- Crucial for catching early bugs in functions and methods.

#### 2. Integration Testing:

- Ensures different parts of the application work together.
- Important for verifying interactions between views, models, and forms.

### **3. Functional Testing:**

- Checks the entire application from the user's perspective.
- Ensures user-centric features like login, registration, and adding cars to a wishlist work as intended.

These procedures provide comprehensive coverage and ensure the application is robust and user-friendly

#### **Notes:**

Accounts used are:

Wonchang as dealer (Password: wasd1312)

Wonchang\_Friend as customer (Password: wasd1312)

Dajuna as the admin (Password: wasd1312)