CS551QEnterprise Software Development Solo Assignment

Student ID: 52212596 Name: ZISHUO LIU

This is a report on the development of a django-based web app, which focuses on how to implement various functions in a web page through django, mainly focusing on the following areas: connecting tables, displaying all data, adding shopping carts, validation, error handling, distinguishing identities and connection handling images

1.The first is associated with two tables, in the associated product details in the models within the product details table using ForeignKey foreign key associated with the goods, the essence of one-to-one can also use OneToOneField, and generate orders is a multi-to-many relationship. An order can have more than one product, and a product can also appear in more than one order, so first created a Part table to store the number of goods and order goods, in the order table with ManyToManyField associated Part table. Django will automatically create a third table to store this relationship, and in the generation of orders, still, use Order.objects.create and Parts.objects.create to create, and then through the order instance. add to add multiple Parts can be.

2.The second point is how to display all the data and use the paginator. The paginator uses Django's built-in pagination method Paginator, which generates the required data by passing in the corresponding data and page size, and then iterates through it in the front-end.

3.The third point is how to add a shopping cart. The essence of the add-to-cart function is to store the corresponding product id (key) and quantity (value) through the session, the shopping cart will be emptied when the session is cleared, and the way to get the shopping cart data is to iterate through all the product data. The method of getting the shopping cart data is to iterate through all the product data, match the session with the product id to get the quantity, and add the final returned data through the dictionary, at which point each piece of data in the front end will have its own quantity.

4.The fourth point is how to validate the user information. When the user logs in, the page send a POST request, the backend receives the username and password and uses User.objects.filter(username=username, password=password) to check if the data exists in the database.If it exists, the product list page is redirected, if not, an error message is returned

5.The fifth point is how to do method error handling, where the errors are mainly login information errors and registration errors. This is achieved by using messages.error(request, ''), a standard method provided by Django to add messages, while in the front end, it is only necessary to determine the existence of messages and pop up the message through the script alert popup.

6.The sixth point is how to distinguish the identity of the user. The essential difference between the identification of ordinary users and administrators is the identity field, the identity of the administrator is admin when querying the data through User.objects.filter(username=username, password=password), by judging the identity field of the data The distinction can be made

7.The seventh point is how to relate the tables to the database and display the order quantity in real-time. The first step is to use the calendar library to get the first and last day of the month, and then iterate through to complete the middle days, using all the dates as the x-axis, and passing the time into the order at the same time as iterating through to find out all the orders for the day.then through the order.electronics.all() can get the order within the commodity information, traversing this information can then be superimposed on the number of this day's sales volume, at this time the sales volume for the y-axis, then each day has the corresponding number, the front-end in access through the Axios access interface, this method will return the corresponding json format data and give the front end a value.

The last point is how to use the behave test, which consists mainly of testing the availability of links and functions on a web page, for example links and functions on the home page, such as login.