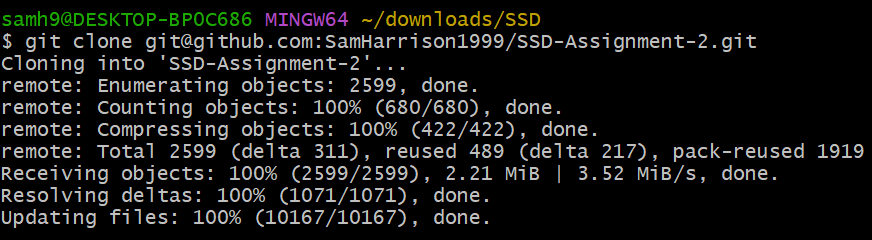
Course: MSc Computer Science

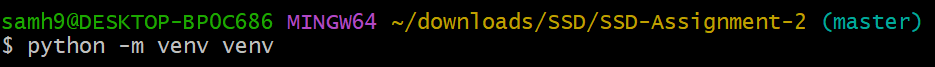
Module: Secure Software Development March 2024

Assignment: Coding Output and Evidence of Testing

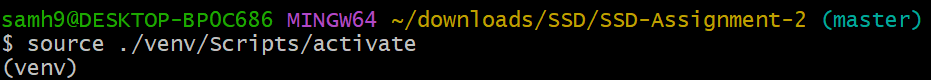
# Cloning the project



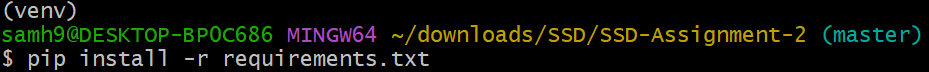
# Creating a virtual environment



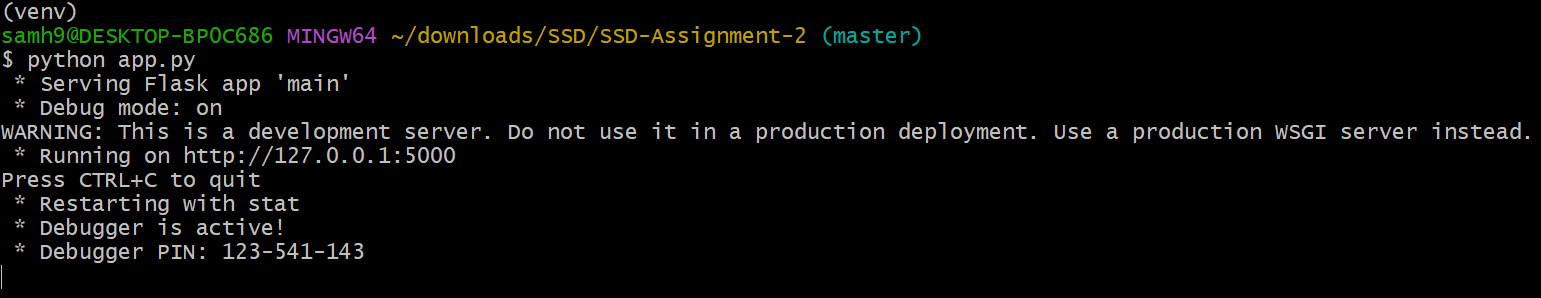
# Activating the virtual environment



# Installing the project requirements



# Starting the flask application



# Sign up

## Steps for signing up

1. Go to the [register page](http://127.0.0.1:5000/register)
2. Fill in the form
3. Click create account

## [Successful signup](https://1drv.ms/v/c/e08399976f28f5bf/EZwU-rkJXu9LvMxfsKTsQyUBp9a4hPUhlzSDxH5v94-lNg?e=qP0Kub)

## [Creating an account with a username that already exists](https://1drv.ms/v/c/e08399976f28f5bf/EVuYjNwH22tKsO2iQi6XLD4B1edXd_wlrww67FDY0Jig4g?e=i6NUQJ)

## [Creating an account with an email that is already associated with an account](https://1drv.ms/v/c/e08399976f28f5bf/EVOw45L4-HNAnAKUJ-xl5eABS4x6A7Ka3pXJjqFSvCqh9g?e=mc2cPZ)

## Creating an account with validation errors

### [Creating an account with no username](https://1drv.ms/v/c/e08399976f28f5bf/EXdhAFzPaMlJgzjUG4VAG00BZHSPUWdyvFmxdLsKPHf4dA?e=LTNwlQ)

### [Creating an account with an invalid email address](https://1drv.ms/v/c/e08399976f28f5bf/EUwp-NNylA9DubRwU1SnO6gBsFCq725b1xiBQ6NRWI4IHA?e=7XLgME)

### [Creating an account with passwords errors](https://1drv.ms/v/c/e08399976f28f5bf/Edh3tyX5BahJrMNHkR9AN-sB40qTWYAnivb-S-hba_MQhw?e=yj2bHr)

# Login

## Steps for logging in

1. Go to the [login page](http://127.0.0.1:5000/login)
2. Enter credentials
3. Click login

## [Login as an administrator](https://1drv.ms/v/c/e08399976f28f5bf/EVKxpbznTJhNr_CyCEN1qUoBaLBbEDk8lL_05WB6YyUYWw?e=w2uCZ6)

## [Login in as a customer](https://1drv.ms/v/c/e08399976f28f5bf/EdxzzRa1_ThFhDJH3XMRx9wBqm9_hdSVXZ_SQwAT4N8aSQ?e=7sMumy)

## [Failed login attempt](https://1drv.ms/v/c/e08399976f28f5bf/EQdn6-qwK31KhN8iu6mKg74BaivJypjNYvUF4bRaEwUxIQ?e=dH4LXs)

# Logout

## Steps for logging out

1. Click Logout

## [Logging out as an administrator](https://1drv.ms/v/c/e08399976f28f5bf/EZPq1khTC_tCnn-MDXZZKoMBbZj41eRLeOsYaoR8dj-_oA?e=BoRub5)

## [Logging out as a customer](https://1drv.ms/v/c/e08399976f28f5bf/EUeKPmt3s0xItn-egttqzfABbTpM1M9cfuilY83yeKdI_A?e=3LpZH8)

# User Profile

## Steps for accessing your individual user profile

1. Login
2. Click profile

## [Accessing your profile](https://1drv.ms/v/c/e08399976f28f5bf/EdmDTQ5GkLVHiKyxHoKYS9wBnHQSUfXZbFm8YfIXQGHj6Q?e=UszGvd)

## [Access is denied when attempting to access another user’s profile](https://1drv.ms/v/c/e08399976f28f5bf/Eb56wvnimN5HuxAazWwAEzMBJ1z8vIhu6U9CHtI_G5AdNA?e=iWZxtn)

# [Page not found](https://1drv.ms/v/c/e08399976f28f5bf/EQmLWF8TzjtFhZqR9WTMe0MBVHMUU4H_xU__vH7aW3Tyzw?e=3Ei9pr)

# Change Password

## Steps for updating your password

1. Login
2. Click profile
3. Click change password
4. Fill in form
5. Click update password

## [Successfully updating password](https://1drv.ms/v/c/e08399976f28f5bf/EdHS8pb6C5pDmBqCTr-KMEgBWr-62x-A85xfM7VdkjnYEQ?e=3zEETV)

## Error when updating password

### [Current password is incorrect](https://1drv.ms/v/c/e08399976f28f5bf/Ebn1Q87wVodCkhyeeSW_BTIBlPnI4dk6ZkQIuD_c7YdOKA?e=QzApB4)

### [New passwords don’t match](https://1drv.ms/v/c/e08399976f28f5bf/ETt3O2icOJ5FmQ-UxfdFOcEBqFa85O_UqGNkBbCHYNqi2g?e=cApvTa)

### [Password minimum requirements not met](https://1drv.ms/v/c/e08399976f28f5bf/EUonR0TL7LZEqEAfKLAG-zQBfEJV3tzH1cyitywe1FNaCQ?e=eWQVzk)

## [Access is denied when updating another user’s password](https://1drv.ms/v/c/e08399976f28f5bf/EeKPt5PWaaNHljPOlgYNXjMBsSy0_nzxWt0iiFUV6WINAw?e=78jedr)

# Creating a product

## Steps for creating a product

1. Login
2. Click add new product
3. Fill in form
4. Click add product

## [Successfully creating a new product](https://1drv.ms/v/c/e08399976f28f5bf/EWlUS2YRpkVNiwfZ7f4o63ABx6YvSx0DvsbaYQPw3UHC3Q?e=6Pm9UJ)

## [Error creating a product](https://1drv.ms/v/c/e08399976f28f5bf/EQbMJeKidq1OlLRzRCb4lUQBkudUFKVrmX7wpShO28lvRA?e=CLDp7r)

## [Access is denied when a non-admin user attempts to create a new product](https://1drv.ms/v/c/e08399976f28f5bf/EUauPzX4yENLhVFvzPu6QEYB5-4qy1w05Vzve2bn4oCcHg?e=oaiYwP)

# Viewing products

## Steps for viewing all products

1. Login
2. Click manage products

## [Administrator able to view products](https://1drv.ms/v/c/e08399976f28f5bf/EdRNsDnr-xtBokr548BHX3EBRMbVkDNs86F4xXbvDO9U1Q?e=3wk29Y)

## [Access is denied when a non-admin user attempts to view all products](https://1drv.ms/v/c/e08399976f28f5bf/EUwsakvdtwZCnyX2BFsMhaMB6hYbkUM91fYjsiBcaLkOOw?e=UOvpfp)

# Updating Products

## Steps for updating a product

1. Login
2. Click manage products
3. Click Update
4. Fill in the form
5. Click update product

## [Successfully updating a product](https://1drv.ms/v/c/e08399976f28f5bf/EQ-63ad5CYxImtEf_vE8vi4BKa3vAr0iiLmu7kmG54ZcUg?e=6qHDgP)

## [Error when updating a product](https://1drv.ms/v/c/e08399976f28f5bf/EU89yrmILy9An8zH6CRTdLwBp2LW9BmUthu4ExOScJ71rA?e=gnWAS4)

## [Access is denied when non-admin user attempts to update a product](https://1drv.ms/v/c/e08399976f28f5bf/EeGYUqYBgcxIg3_p-PEPGLwBALbPfOu6gdzUy5HK6t9T7Q?e=CNEWQF)

# Deleting a product

## Steps for deleting a product

1. Login
2. Click manage products
3. Click delete

## [Successfully deleting a product](https://1drv.ms/v/c/e08399976f28f5bf/EV1wmQWDBAJHsaHv-BQu2QABceDZxCZGrf9gz4vJuFDYdg?e=zOzkvw)

## [Access is denied when non-admin user attempts to delete a product](https://1drv.ms/v/c/e08399976f28f5bf/ESogKWqh9BFHvoAVH_SAUAEBXScI8VsdDbD9kH_AAIohpA?e=un8Txq)

# Add to cart

## Steps for adding an item to your cart

1. Login
2. Click add

## [Item not already in cart](https://1drv.ms/v/c/e08399976f28f5bf/Ee7b8A_XJqxPj-eQ-NuI1OwBYbRdCQS9IpqJYnwyS9Lv1w?e=ODbhaG)

## [Item already in cart](https://1drv.ms/v/c/e08399976f28f5bf/EUSev1sBE4NOh522R3s-kJQBSY0S_Nyhkbj_fxODsu5N1Q?e=AR8eOw)

# [Increasing the quantity of an item in your cart](https://1drv.ms/v/c/e08399976f28f5bf/EYzAmZ09qfNBvK6R8OPelpwBnt1bZWf9-GMgqvXFuRpMkw?e=2DObft)

## Steps for increasing the quantity of an item in your cart

1. Login
2. Add item to cart
3. Click cart
4. Click plus button

# [Decreasing the quantity of an item in your cart](https://1drv.ms/v/c/e08399976f28f5bf/EVI48Kg5rYxEr2W7UfuHN0EBgQ3IYrEx32HvM4xt1UsAeA?e=qMPBul)

## Steps for decreasing the quantity of an item in your cart

1. Login
2. Add item to cart
3. Click cart
4. Click minus button

# Removing an item from your cart

## Steps for removing an item in your cart

1. Login
2. Add item to cart
3. Click cart
4. Click remove button

### [Removing an item from your cart when it’s the only item in your cart](https://1drv.ms/v/c/e08399976f28f5bf/EQ_OJg5ad49LkwUd_I9XxxIB-llrCacsrPZwvrvnhZqJTQ?e=LBM2FB)

### [Removing an item from your cart when multiple items are in your cart](https://1drv.ms/v/c/e08399976f28f5bf/EXRx4TBC3YlBjxkCIxhaq6wBGDh9_c9YO3mKqFrk71uH-w?e=ptkaRD)

# View Users

## Steps for viewing all users

1. Login
2. Click manage users

## [Administrator can view all users](https://1drv.ms/v/c/e08399976f28f5bf/EXq900kRZiJGn1O7BBVbaHsBPnXfjPcyASmwYcW780wFCA?e=dTJpIT)

## [Access denied when non-admin user attempts to view all users](https://1drv.ms/v/c/e08399976f28f5bf/ESp-BdpVBWRCtk7RTlpt2CwBkZitAAIg5yb1G9Ws5e2pnw?e=D5H2rg)

# Deleting a customer

## Steps for deleting a user

1. Login
2. Click manage users
3. Click delete on the user you want to delete

## [Successfully deleting a customer](https://1drv.ms/v/c/e08399976f28f5bf/EU5aQYxeaU9Hn_RB6YB2QgkBnb9Ponzdp2_3_urdEFW8sA?e=yJOj2C)

## [Access is denied when a non-admin user attempts to delete a user](https://1drv.ms/v/c/e08399976f28f5bf/EY71U-BtSl5JtDl1VQPnSbcB0CHqAgOHp59_xKaeTXXFYQ?e=1mXjoR)

# Placing an order

## Steps for placing an order

1. Login
2. Add item to cart
3. Checkout

## [Successfully placing an order](https://1drv.ms/v/c/e08399976f28f5bf/Ea3HpM21nq1Ci-kkAmsDyTUB_7tvm_ssVrarasv-Ogo1qg?e=1dRJqB)

## [Failing to place an order if there is not enough stock available](https://1drv.ms/v/c/e08399976f28f5bf/EVEwHX3yYjRAhLlMQodSQ4sBk2Zi37CsOSfsIcwY102q_Q?e=ZtUnh0)

# Order history

## Steps for viewing your order history

1. Login
2. Click orders

## [Order history with no orders](https://1drv.ms/v/c/e08399976f28f5bf/EYrXswrZ52JGmbh1X7SMHzcBNA63uXQYY_b7knzMsSdvEQ?e=KXEfYJ)

## [Order history with orders](https://1drv.ms/v/c/e08399976f28f5bf/EY5GdsHkCtFLruTDH7m1vmEBrWuRuw8oi3LQxBGmS3x4mg?e=w1dF00)

# View orders

## Steps for viewing all orders

1. Login
2. Click manage orders

## [Administrator can view all orders](https://1drv.ms/v/c/e08399976f28f5bf/ER8lnD1pfepJkFeuSCWUQJoBwHZmMSBoF5yccjokWUojiw?e=wtaMZx)

## [Access is denied when a non-admin user attempts to view all orders](https://1drv.ms/v/c/e08399976f28f5bf/Efe-IEW9jtxEpgeG9TezZCsBOf7xyfD9H21MIoiVEM06Ow?e=fch4ZF)

# Updating order status

## Steps for updating the status of an order

1. Login
2. Click manage orders
3. Click update
4. Select updated order status
5. Submit form

## [Successfully updating an orders status](https://1drv.ms/v/c/e08399976f28f5bf/ETRmE2kyyMpPtonvWvU-wI8BUkc0Jkd-r7hVUG2TUjnfog?e=kHLrC4)

## [Access is denied when a non-admin user attempts to change an orders status](https://1drv.ms/v/c/e08399976f28f5bf/EZ1vmsFKtS9GvLHnR_IQo0kBX_uNyjg4qwdeE6kj1cOacg?e=Uzd15y)

# Search

## Steps for search

1. Enter search term
2. Click search button

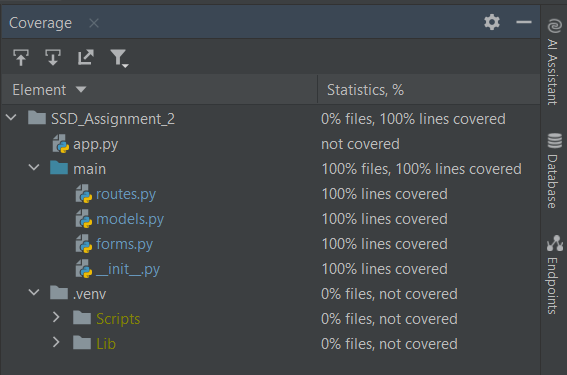
## [No matching items](https://1drv.ms/v/c/e08399976f28f5bf/EXBLEZS4pNVBkxj5XqjIfXgBvqDp4w5HjUmluXnOcr6SZw?e=zTESOS)

## [Matching items](https://1drv.ms/v/c/e08399976f28f5bf/ETPk49T-egZKqQM2LLV-FAEBFwPqu2jsgQ3NpZvcV9amTA?e=Ph352G)

# [Dictionary attack](https://1drv.ms/v/c/e08399976f28f5bf/ESfnwS2l5kBPm0_eDaDrI7MBELjYRmF84lpC2WIA4YuNCw?e=FrmaRa)

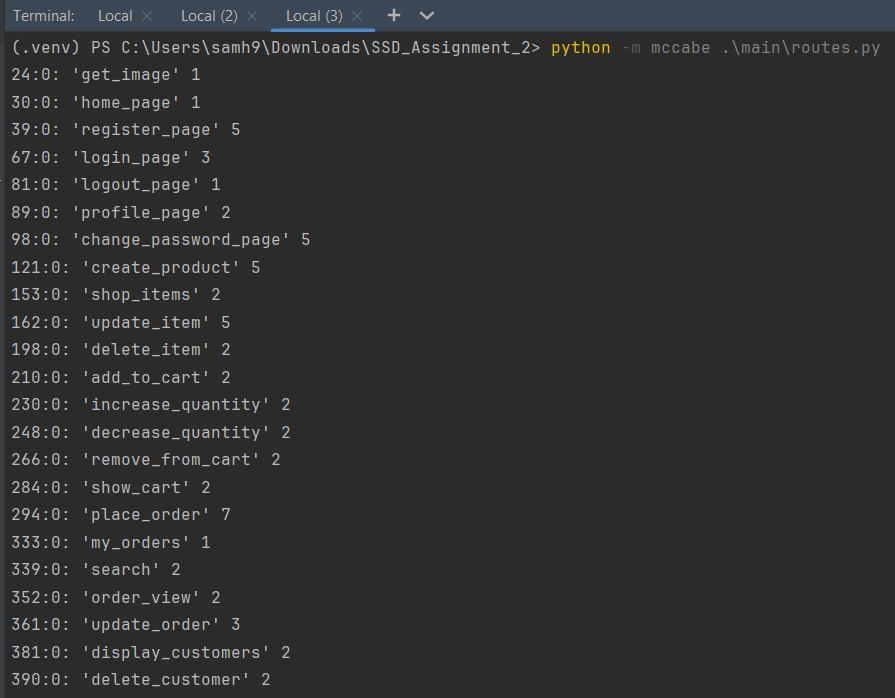
# Coverage Results

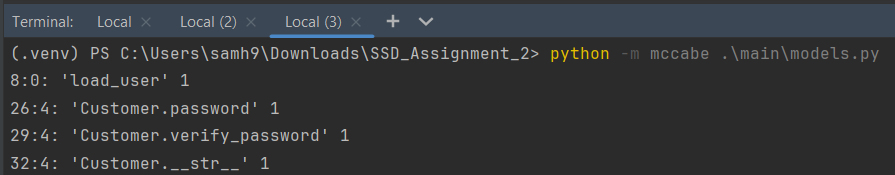
My unit tests covered 100% of the python code for my application. The benefit of this is it allows me to easily add new features without breaking the code or introducing any bugs.

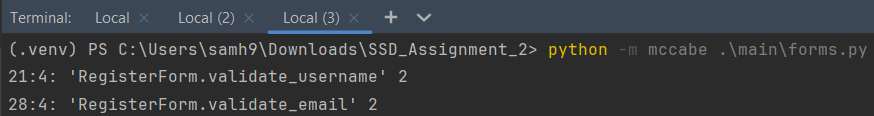


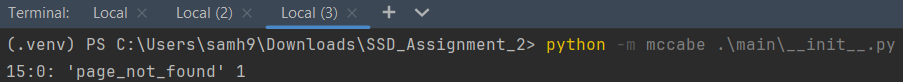
# McCabe results

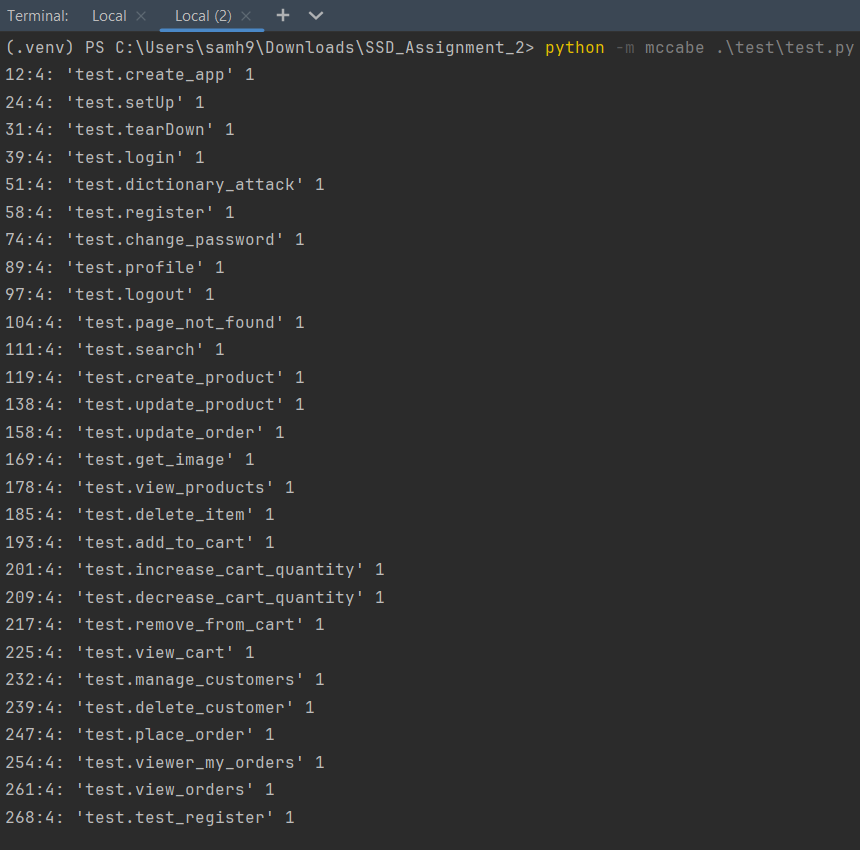
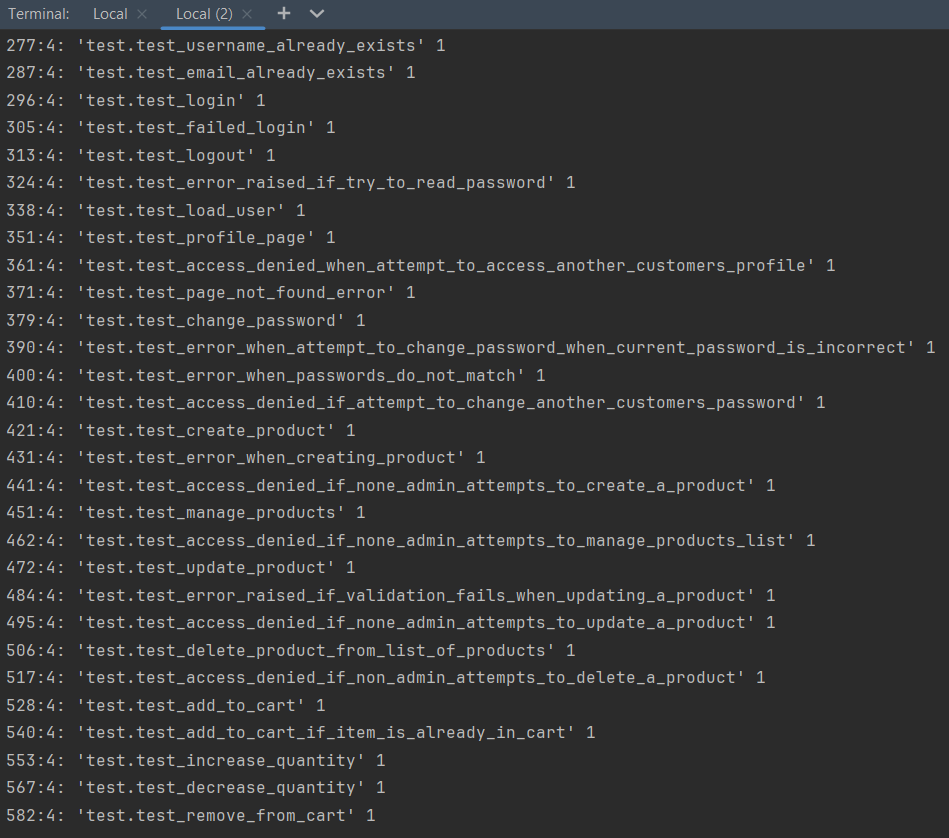
McCabe is a code complexity checker tool used to detect overly complex code based on cyclomatic complexity. The default maximum complexity in flake 8 is 7 so I used this as a guide when writing my code. The lower the cyclomatic complexity the less likely it is that the code contains any defects. I followed the single responsibility principal in an attempt to keep the cyclomatic complexity as low as possible.

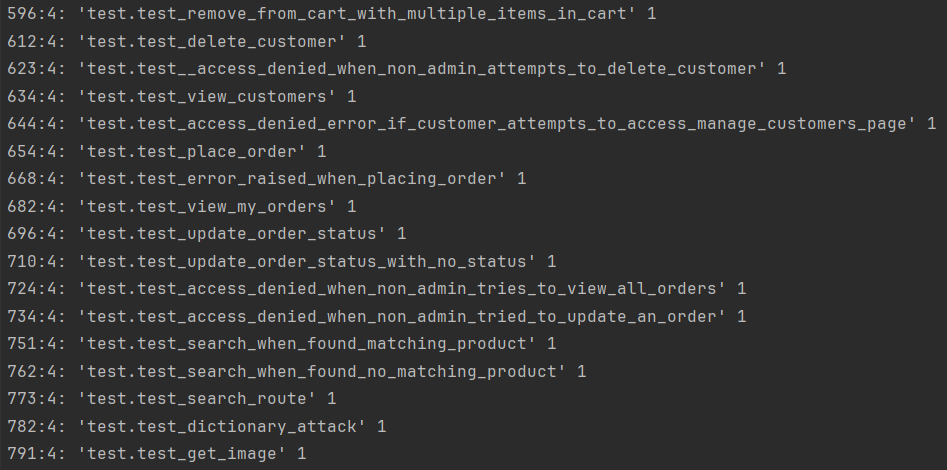


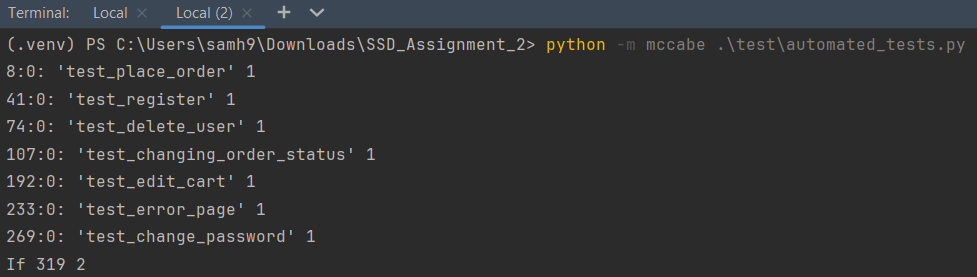






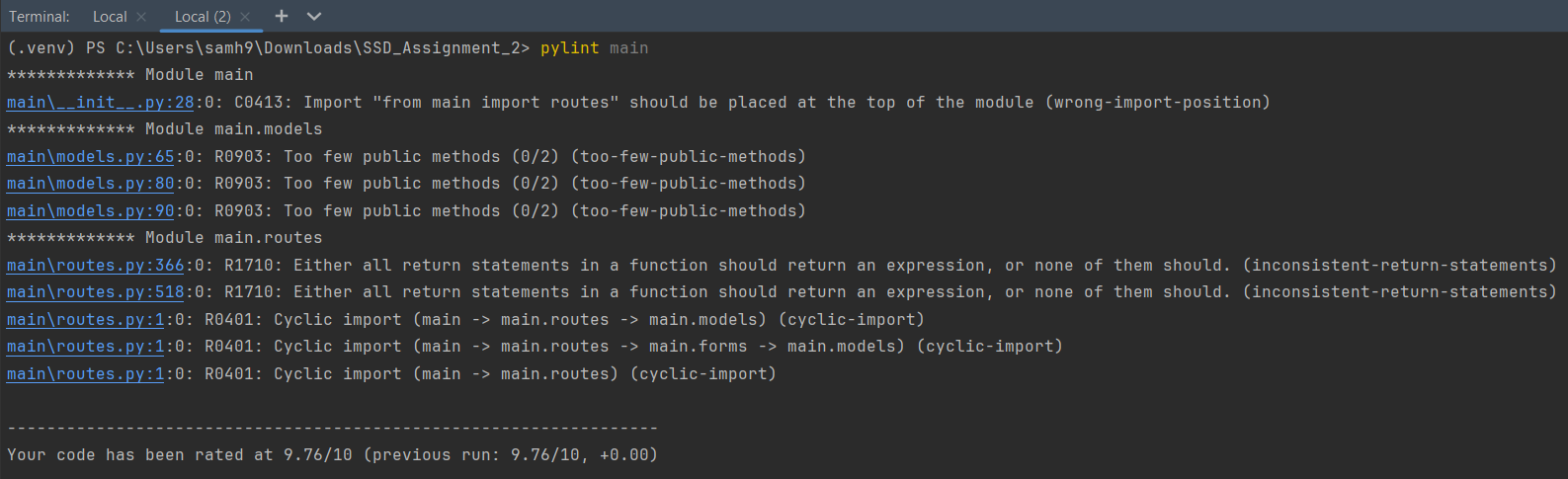
  






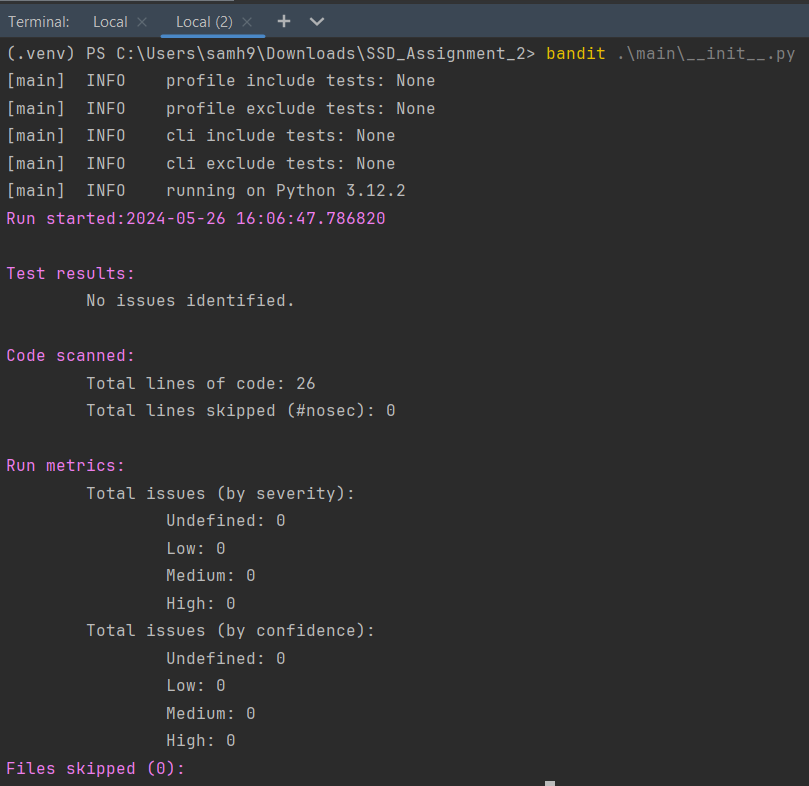
# Pylint Results

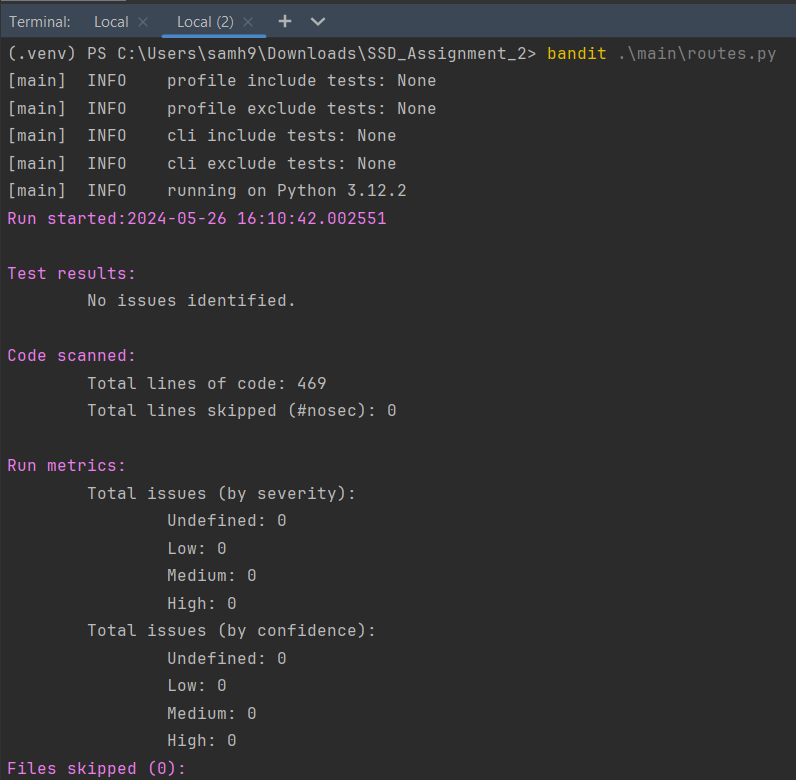
Pylint is a static code analysis tool for python code. It checks that your code doesn’t violate the PEP styling guidelines. The purpose of Pylint is to ensure the readability of your code. Ideally, I would have liked to have received a score of 10 however due to time constraints I was unable to achieve this.

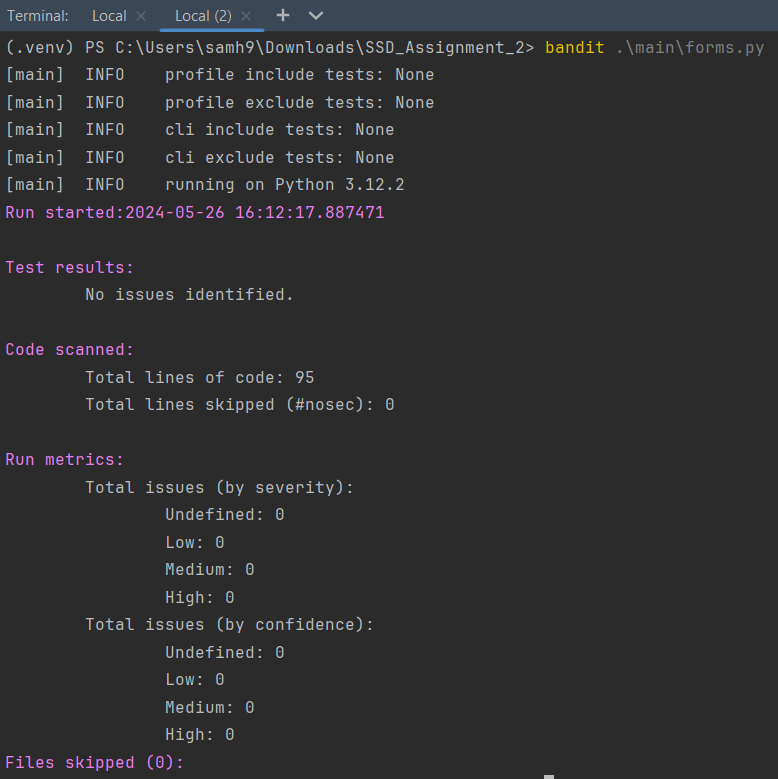


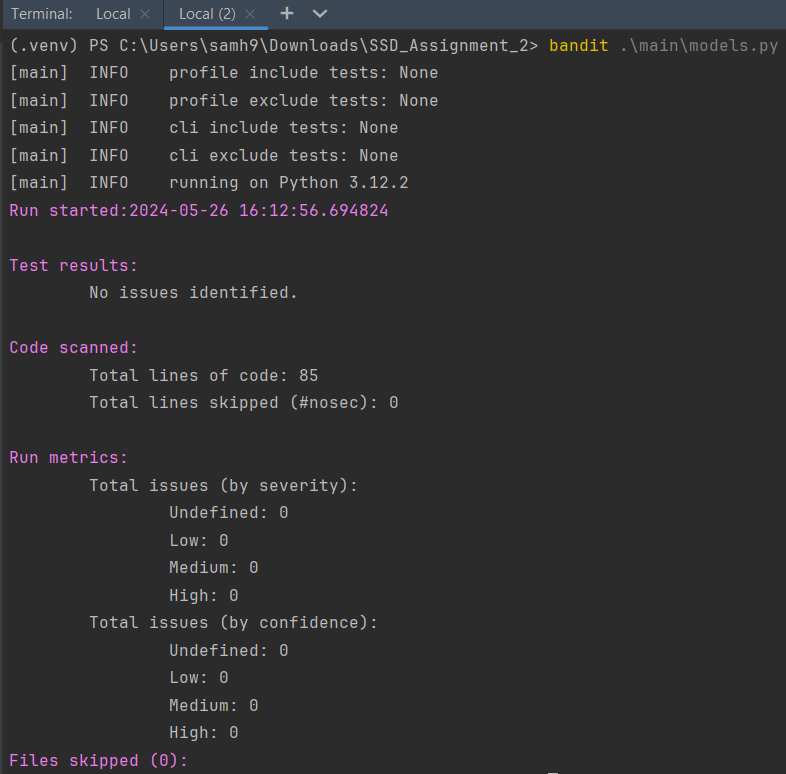
# Bandit Results

Bandit is a tool used to find common security issues in Python code. Based on the results I can determine there aren’t any common security issues in my code.



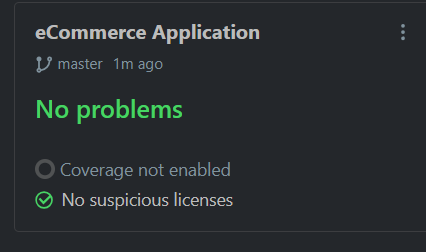


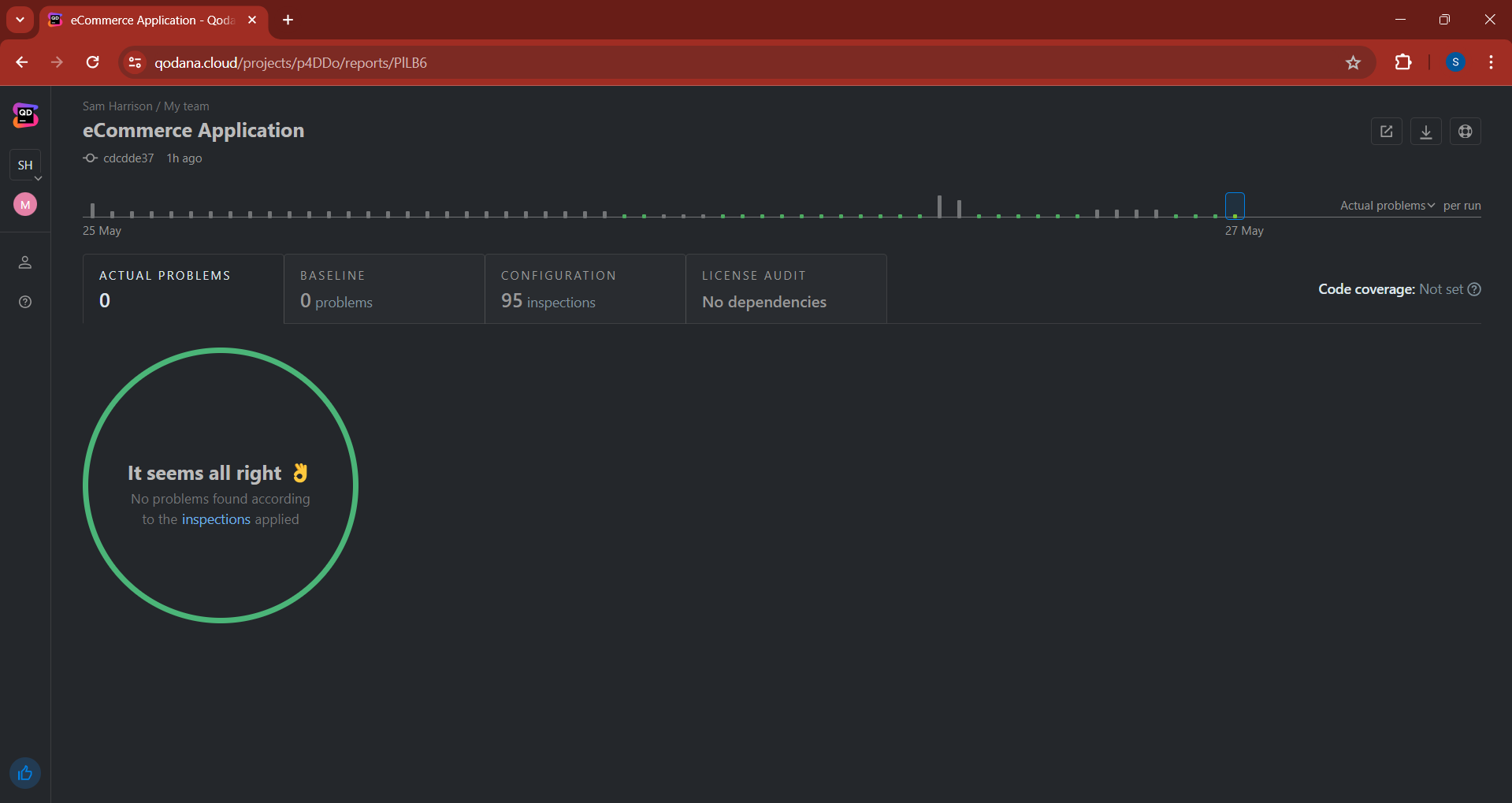




# Qodana Results

Qodana is a static code analysis tool created by JetBrains, it is an alternative to SonarQube. I created a GitHub action so that anytime new code was pushed to the master branch the code is scanned by Qodana for any issues.





# [Unit Tests](https://1drv.ms/v/c/e08399976f28f5bf/EQ9f-vWCGlFLgwzEwlSiQu8BwP8Duphyl56GWW1QjkJXhg?e=lg1Gua)

# [Automation Tests](https://1drv.ms/v/c/e08399976f28f5bf/Ec0_Jvkg3S1IuuCIy6DNMa0BdqUFjaBFZjiJmY2X-ha8LA?e=pMwUgR)

# Explaining the code

## Front End

The front end of my application was written in HTML the reason for this was I had prior knowledge of HTML so could get a basic website designed relatively quickly.

### HTML

I wrote separate HTML files for each of the pages of my application. I used Jinja templating throughout the HTML files so that the website would be loaded dynamically. I extended the base HTML file to ensure consist design throughout my application.

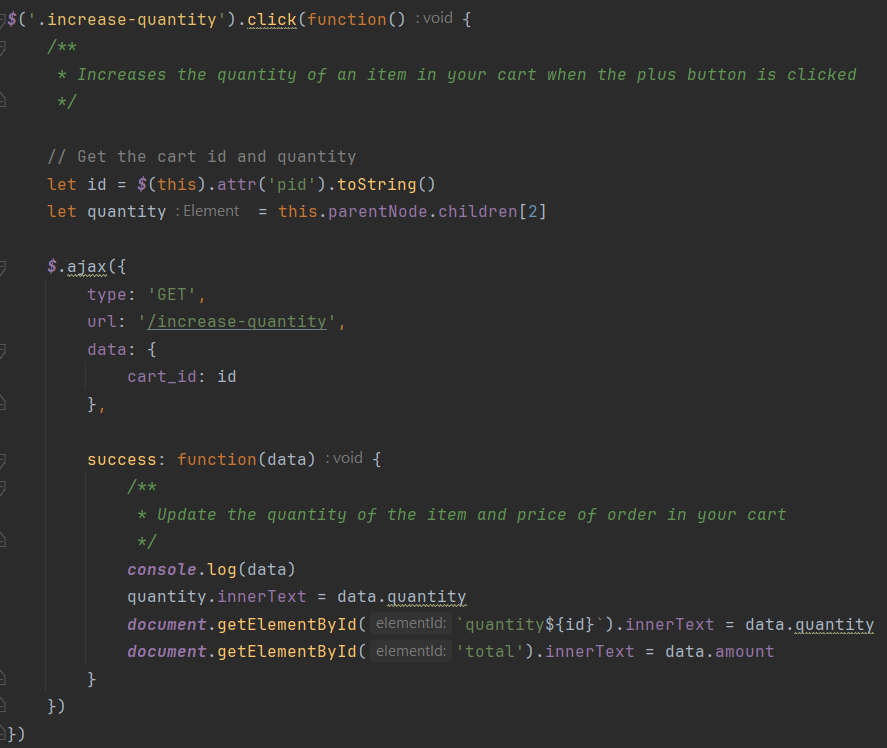
### CSS

I wrote a small amount CSS code to style my application instead mainly using bootstrap for styling my application.

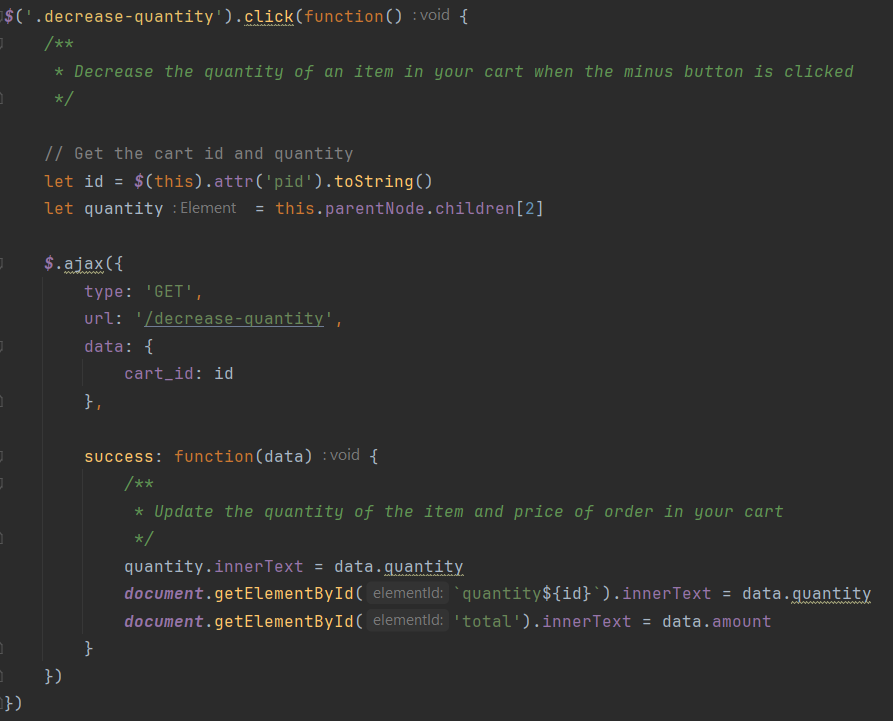
### Java Script

The JavaScript in my application was used to alter the quantity of an item in your cart.

#### Increase Quantity Function



#### Decrease quantity function



#### Remove from cart function



## Back End

### Routes

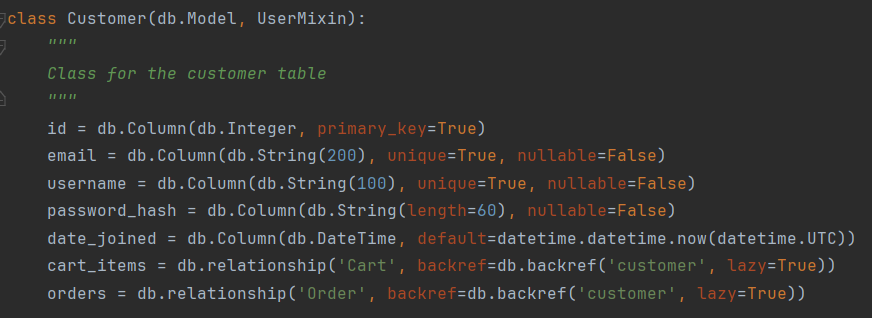
My application has 23 routes. Each route is associated with a particular endpoint and contains the functionality for each feature and is responsible for connecting the front and back end of my application. The reason I have designed my app like this is to limit the attack surface. I used the render template function from flask to render the appropriate HTML page for each end point. More information about each individual endpoints can be found on the API HTML file in the doc’s folder. I used sphinxes auto doc feature to generate the documentation based off the doc strings in my project.

The specific routes for my application are the home route, register route, login route, logout route, profile route, change password route, create product route, manage products route, update products route, delete products route, add to cart route, increase quantity route, decrease quantity route, remove for cart route, cart route, place order route, orders route, search route, manage orders route, update orders route, manage users’ route, delete user route and dictionary attack route.

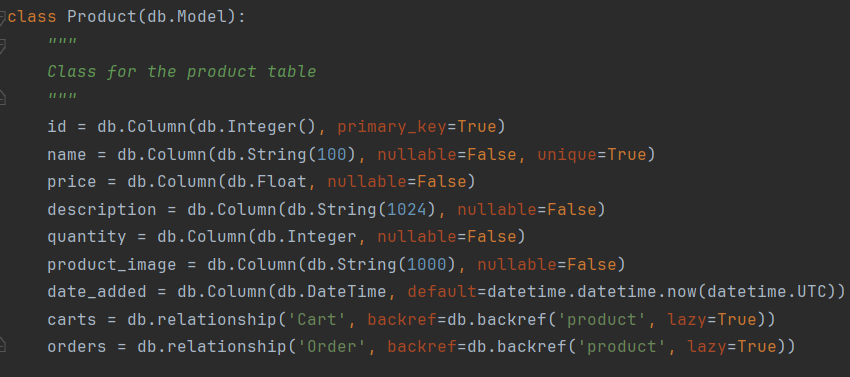
### Models

The model’s class was designed to use SQL alchemy to automatically generate the tables for my application.

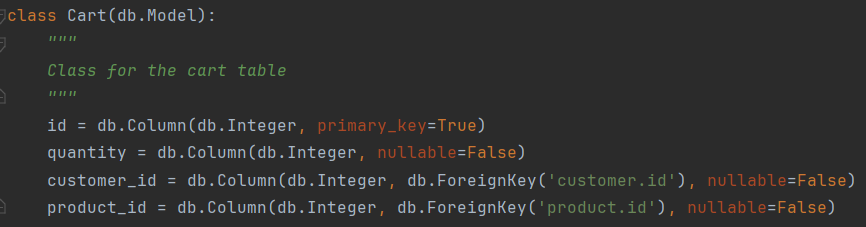
#### Customer Model



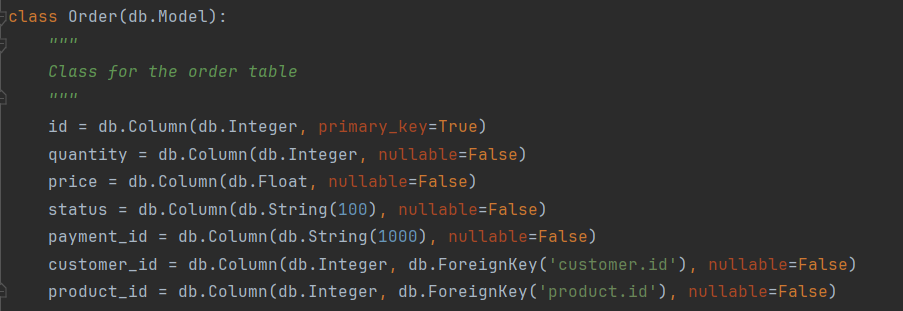
#### Product Model



#### Cart Model



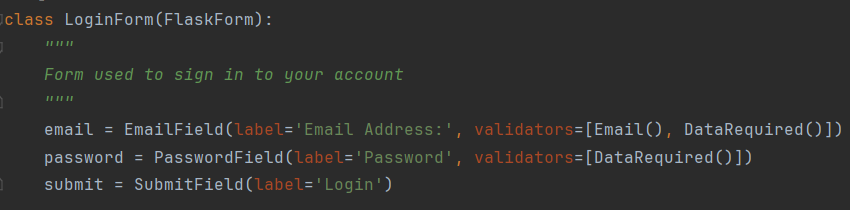
#### Order Model



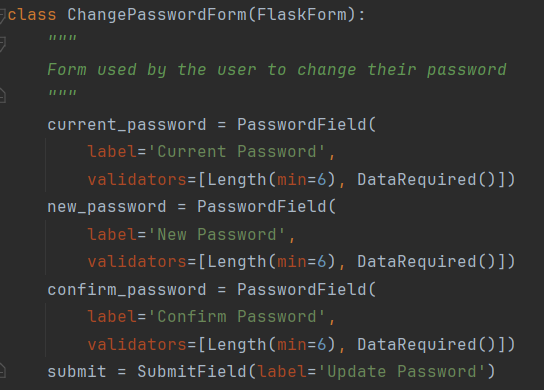
### Forms

The forms class was used to retrieving the information from the front end and pass it to the backend.

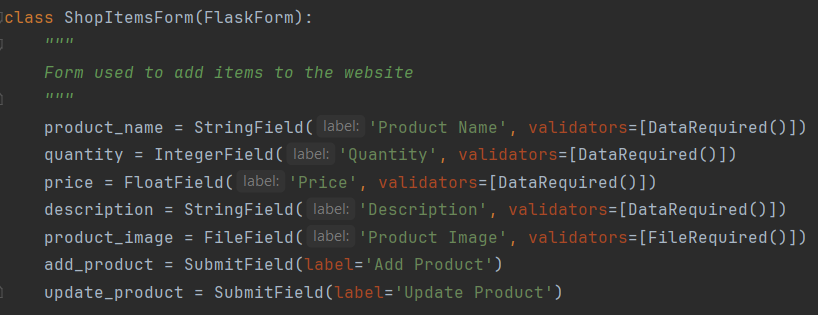
#### Login form



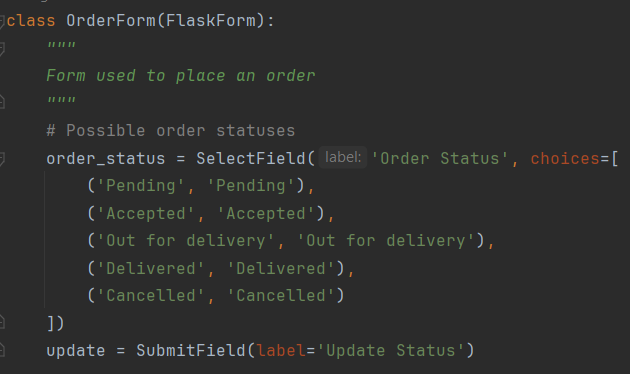
#### Change Password form



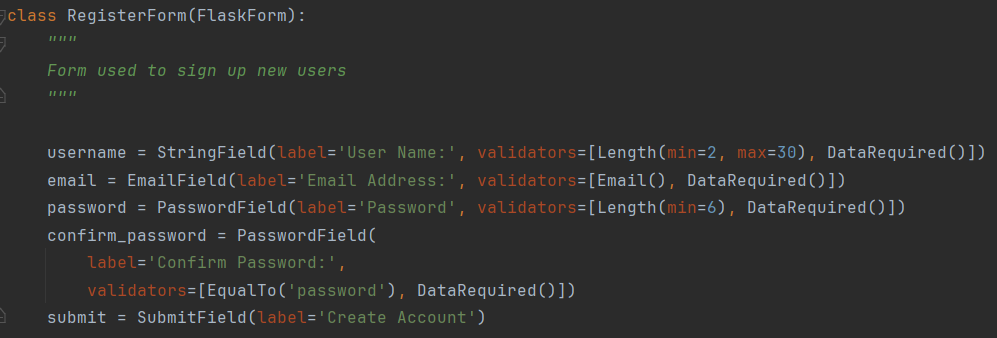
#### Shop items form



#### Order form

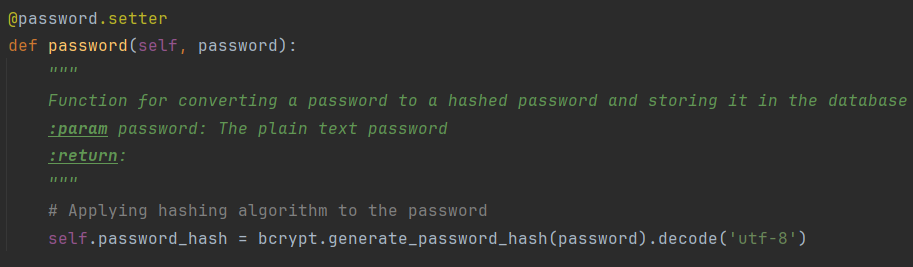


#### Register Form

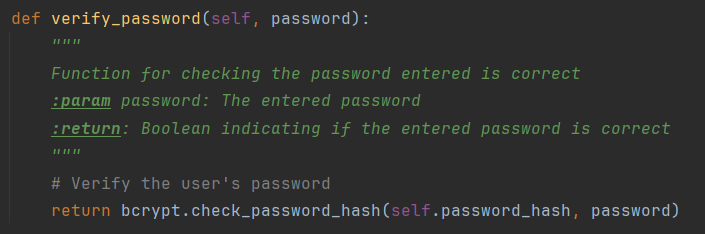


### Password Hashing

I used bcrypt to has the plain text password before storing it in the database. The reason I did this was so in the event a hacker gains access to the database they can’t access customers passwords.

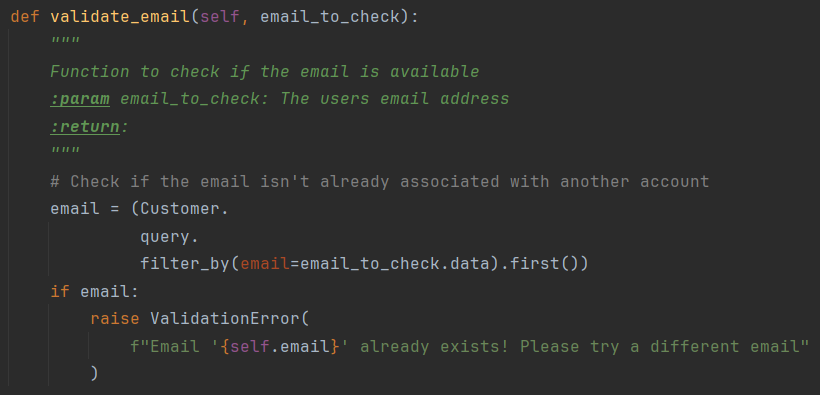


### Verifying credentials

This function checks the password again the password hash to verify a user credentials. 

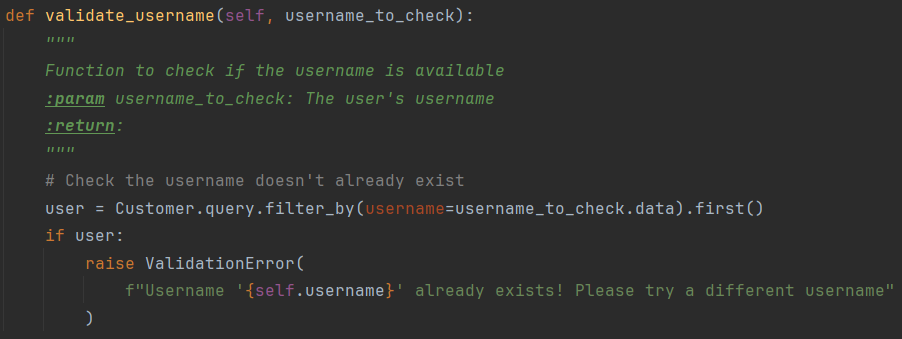
### Email Validation

This function checks the email isn't linked to another account



### Username Validation

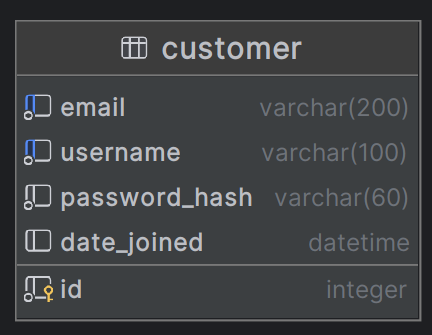
This function checks the username is linked to another account



## Database

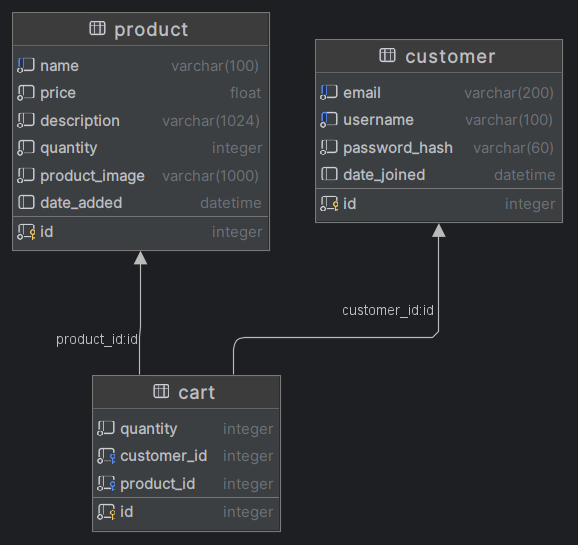
### Customer

The customer table is made up of 5 columns; the customer id, email, username, password hash and date joined. The customer id is an integer used to identify a customer and is the customer tables primary key. The customer’s email is a string no longer 200 characters long. The customer’s username is a string no longer than 100 characters long. The customer’s password is stored as a hash and is 100 characters long.



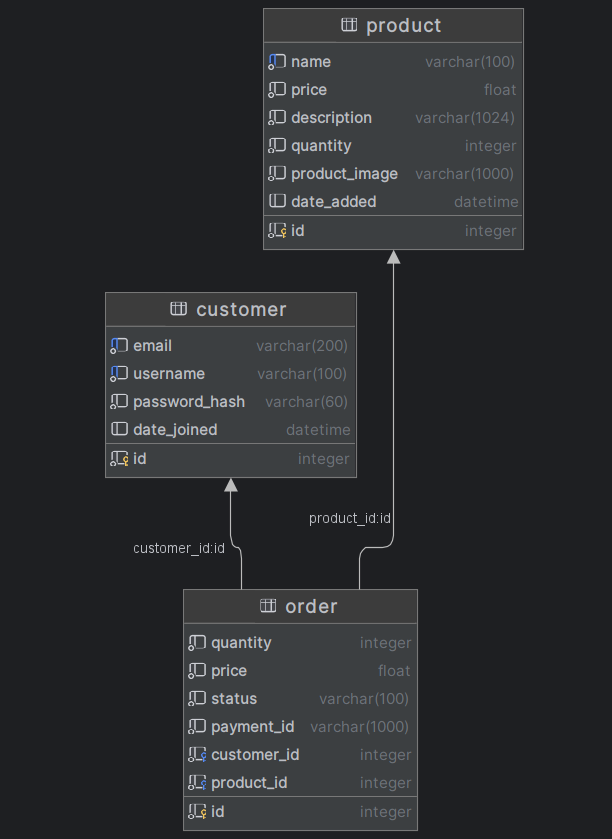
### Cart

The cart table is made up of 4 columns; the cart id, customer id, product id and quantity. The cart id is the cart tables primary key it is used to identify each user’s cart. The customer id is a foreign key used to link the cart to a customer. The product id is a foreign key used to link the item to the customer cart. The quantity is an integer representing how many of a particular product the user is attempting to buy.



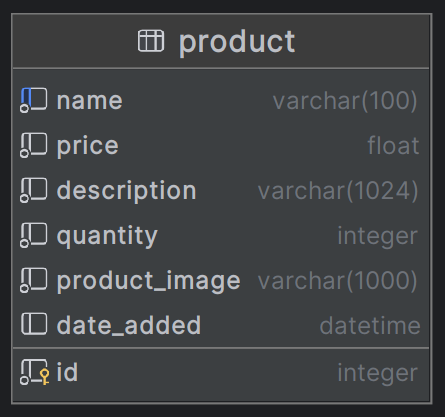
### Order

The order table is made up of 7 columns; the order id, customer id, product id, payment id, status, price and quantity. The order id is the primary key and is used to identify the individual order. The customer id is a foreign key used to link the order to a customer. The product id is a foreign key used to link the item to the order. The status is a string no longer than 100 characters long. The price is a numeric value representing the order total.



### Product

The product table is made up of 7 columns; the product id, name, price, description, quantity, product image and date added. The product id is the primary key and is used to identify each product. The name is a string no longer than 100 characters long. The description is a string no longer than 1024 characters long. The price is a numeric value representing the price of a product. The quantity is a number representing the available stock.



# Future Enhancements:

* Containerise the application with Kubernetes and Docker
* Host the App on Heroku to offer DDoS protection
* Create a build Pipeline in Harness for CI/CD
* Write more automation tests
* Rewrite front end in react
* Fix remaining Pylint issues