CISC-327 group-51 readme first

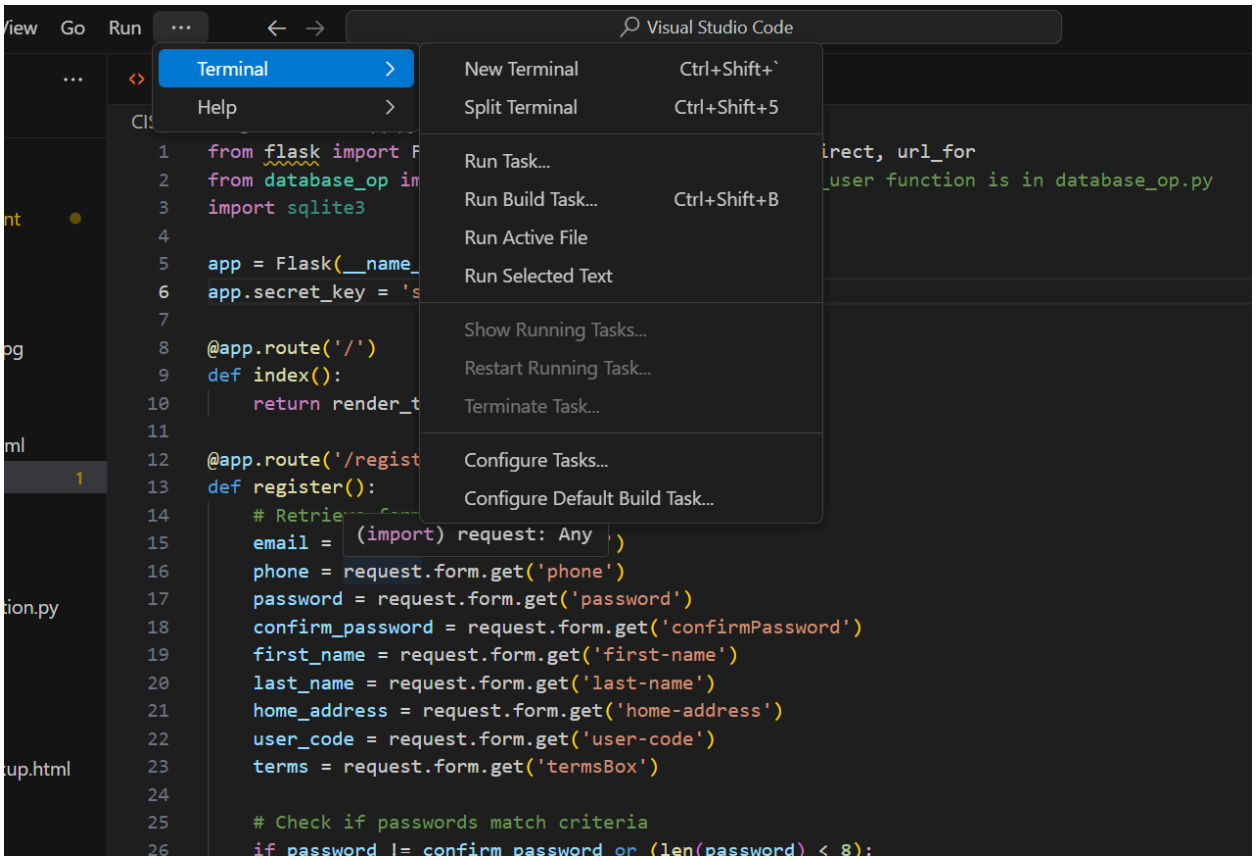
http://127.0.0.1:5000 leads to sign up page

http://127.0.0.1:5000/login leads to log in page

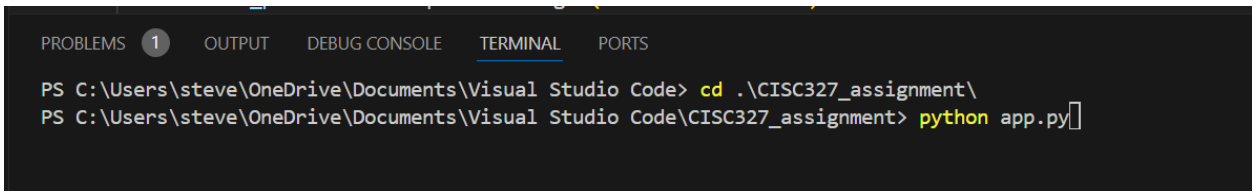
http://127.0.0.1:5000/payment leads to payment page

How to run the flight booker page

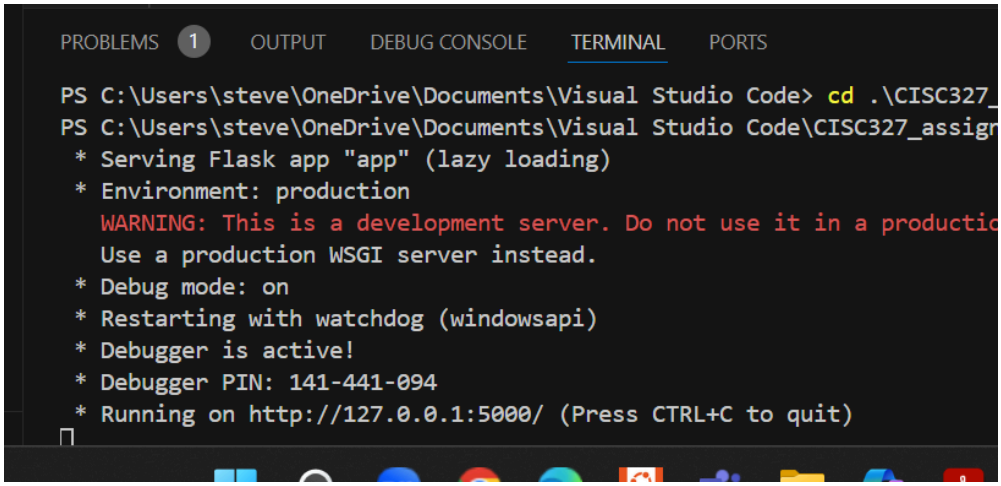
- Start a new terminal



- Make sure that Flask is installed, if not run “pip install Flask”
- Make the current directory to where the app.py is found. Run the app.py by running “python app.py”



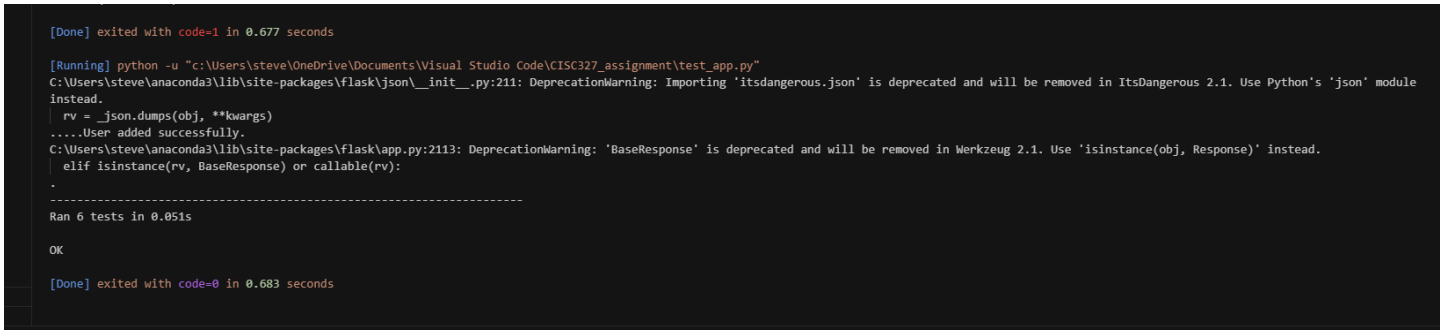
- Copy the link given and paste it onto your local browser



- The link will take you to the registration page, to test the other pages:
- http://127.0.0.1:5000/login will take you to the login page
- http://127.0.0.1:5000/payment will take you to the payments page
- You can test the registration page by adding information as you would normally and clicking register
- To create a test case for registration, copy the skeleton code and fill in or remove information as needed.
    - Keep in mind that the assert message must match the one given in the app.py as well. I.E depending on the testing case.

```
def test_case_registration(self):
        result = self.app.post('/register', data={
            'email': self.generate_random_email(),
            'phone': self.generate_random_phone(),
            'password': 'password123',
            'confirmPassword': 'password123',
            'first-name': 'John',
            'last-name': 'Doe',
            'home-address': '123 Queen Street',
            'user-code': '123456',
            'termsBox': 'on'
        })
        self.assertIn(b'Error/Success message', result.data)
```

- After running the test_app_registration.py, the results should look like this



- After successful registration, you'll be taken to the login page
- To test the login page, you can try logging in with the information you just registered with. Another option is to log in using some of the information that has already been saved in the database:
  email: "hassan@gmail.com" password: "password123"
  email: "steven@gmail.com" password: "password123"
  email: "anwar@gmail.com", password: "password123"
- After successfully logging in, you'll be taken to the payments page
- To test payments: all the payment methods are saved in payments.db, if for any reason payment database is compromised or it doesn't exist, delete the corrupted db file and run init_payments.py by running the code "python init_payments.py" to recreate it and in the database, the following payment methods are saved (note that name is written in lowercase with no space, this is to make sure that names are interpreted properly when submitted, therefore, names such as John Smith, JohnSmith, john smith are all accepted): use these payment methods in the payment page are used to test cases, and real users can use these to simulate purchasing the ticket

method	card number	expiration date	CVC	name	country	balance
debit	11111111 11111111	0125	001	johnsmith	CA	1300
credit	22222222 22222222	0226	002	johnsmith	CA	1500
visa	33333333 33333333	0327	003	johnsmith	US	2000
master	44444444 44444444	0428	004	johnsmith	US	3000
debit	55555555	0529	005	johnsmith	CA	900

	55555555					
credit	22222222 22222222	0226	002	johnsmith	CA	1500
visa	33333333 33333333	0327	003	johnsmith	US	2000
master	44444444 44444444	0428	004	johnsmith	US	3000
debit	55555555 55555555	0529	005	johnsmith	CA	900

-

To run the test cases for flight booker

- Make sure you're in the parent folder of the directory (should be assignment-3 in this case)
- Run the command: **python -m unittest discover -s tests** to test all the test cases in one go
- Or to run them individually for each feature, use these commands:
    - **python -m unittest discover -s tests -p "test_login.py"**
    - **python -m unittest discover -s tests -p "test_payment.py"**
    - **python -m unittest discover -s tests -p "test_registration.py"**

```
(base) hassan@MacBookAir Assignment-3 % python -m unittest discover -s tests
User added successfully.
User with email unique_user@example.com deleted successfully.
.User added successfully.
User with email unique_user@example.com deleted successfully.
.User added successfully.
User with email unique_user@example.com deleted successfully.
.User added successfully.
User with email unique_user@example.com deleted successfully.
.User added successfully.
User with email unique_user@example.com deleted successfully.
.User added successfully.
User with email unique_user@example.com deleted successfully.
.................User added successfully.
.
----------------------------------------------------------------------
Ran 22 tests in 0.033s

OK
```