Backend Test - FullThrottle Labs

**Task:**

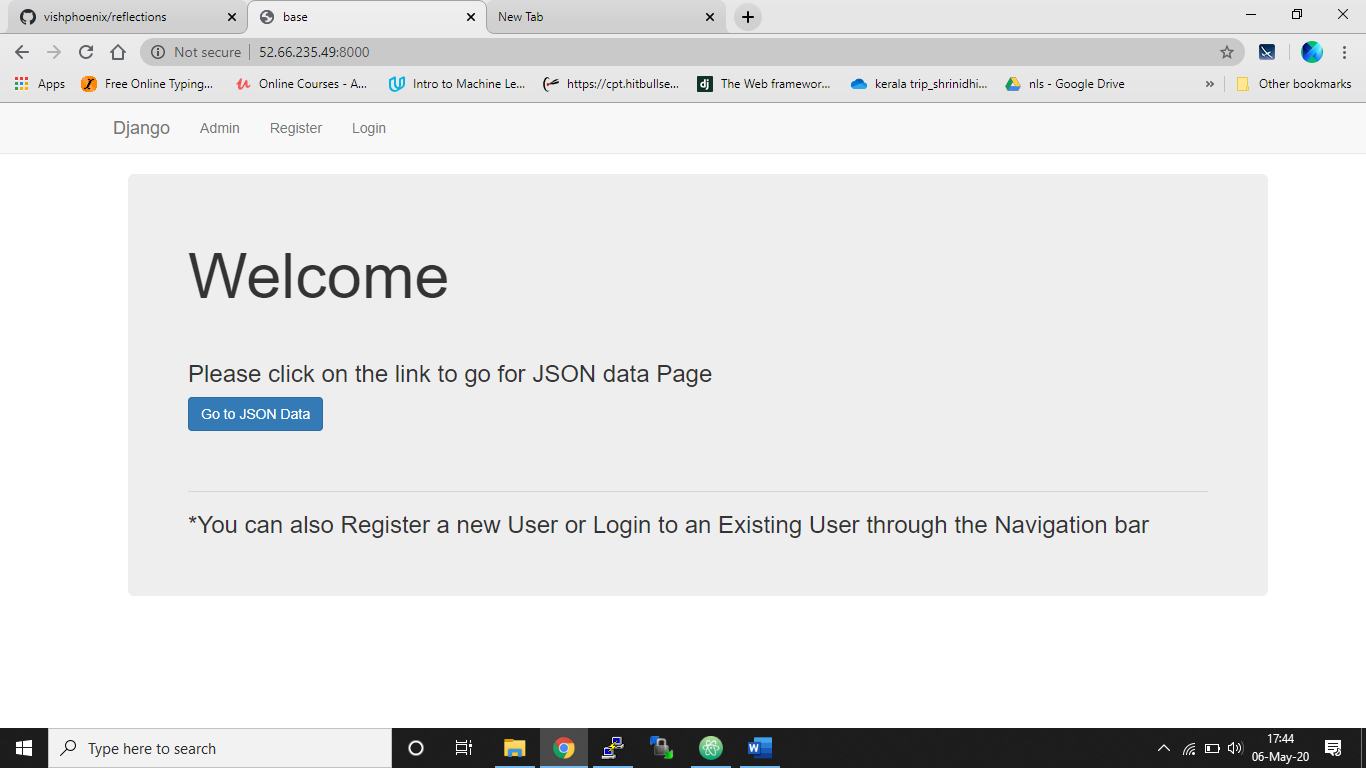
Go through the sample JSON file given here - <https://drive.google.com/open?id=1xZa3UoXZ3uj2j0Q7653iBp1NrT0gKj0Y>

This JSON file describes a list of users & their corresponding periods of activity across multiple months.

Now, design and implement a Django application with User and ActivityPeriod models, write a custom management command to populate the database with some dummy data, and design an API to serve that data in the json format given above.

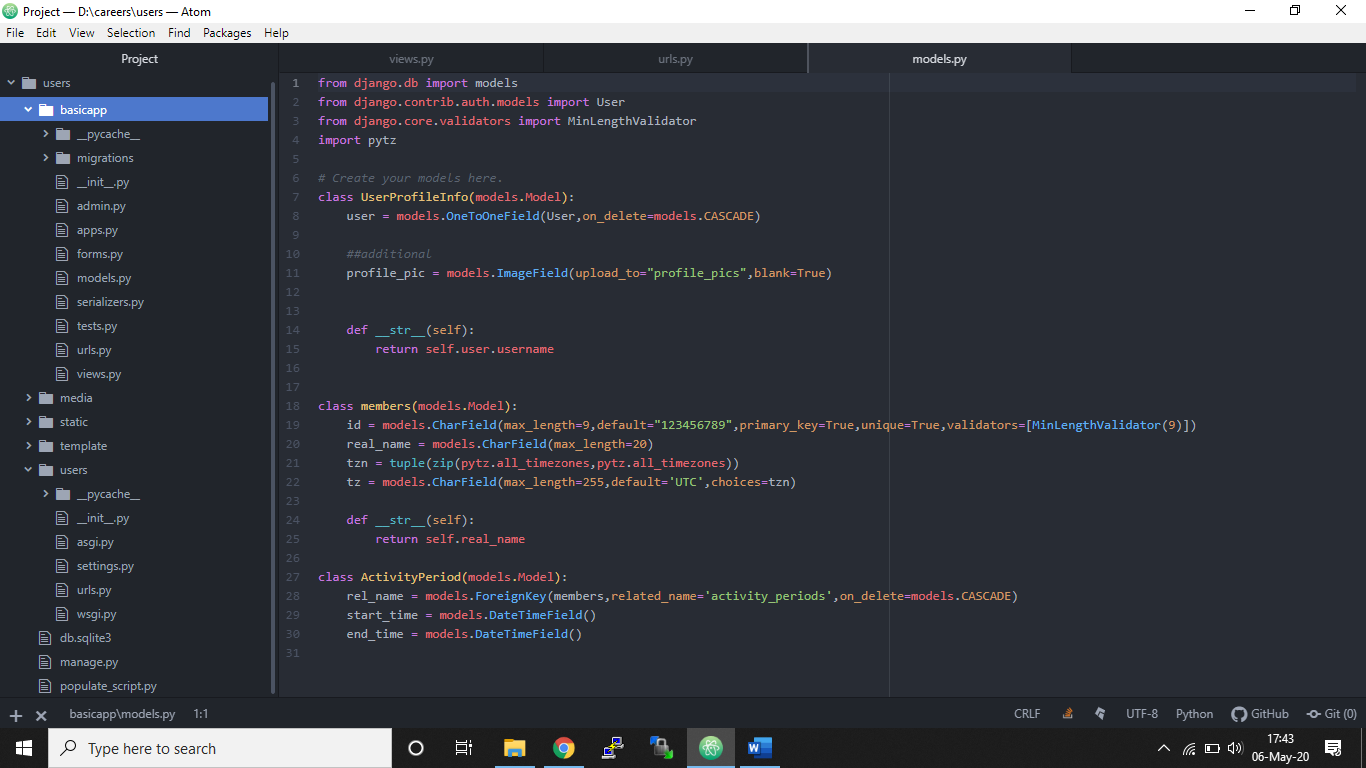
**Solution**

* The assignment is completed using Python & Django, and used the Faker library in order to populate the database with some dummy data.
* Rest API (Serializer) is used to convert the models into JSON format.
* The code and API endpoint is production ready and hosted on a publicly accessible location on AWS (<http://52.66.235.49:8000/>)



Home Page

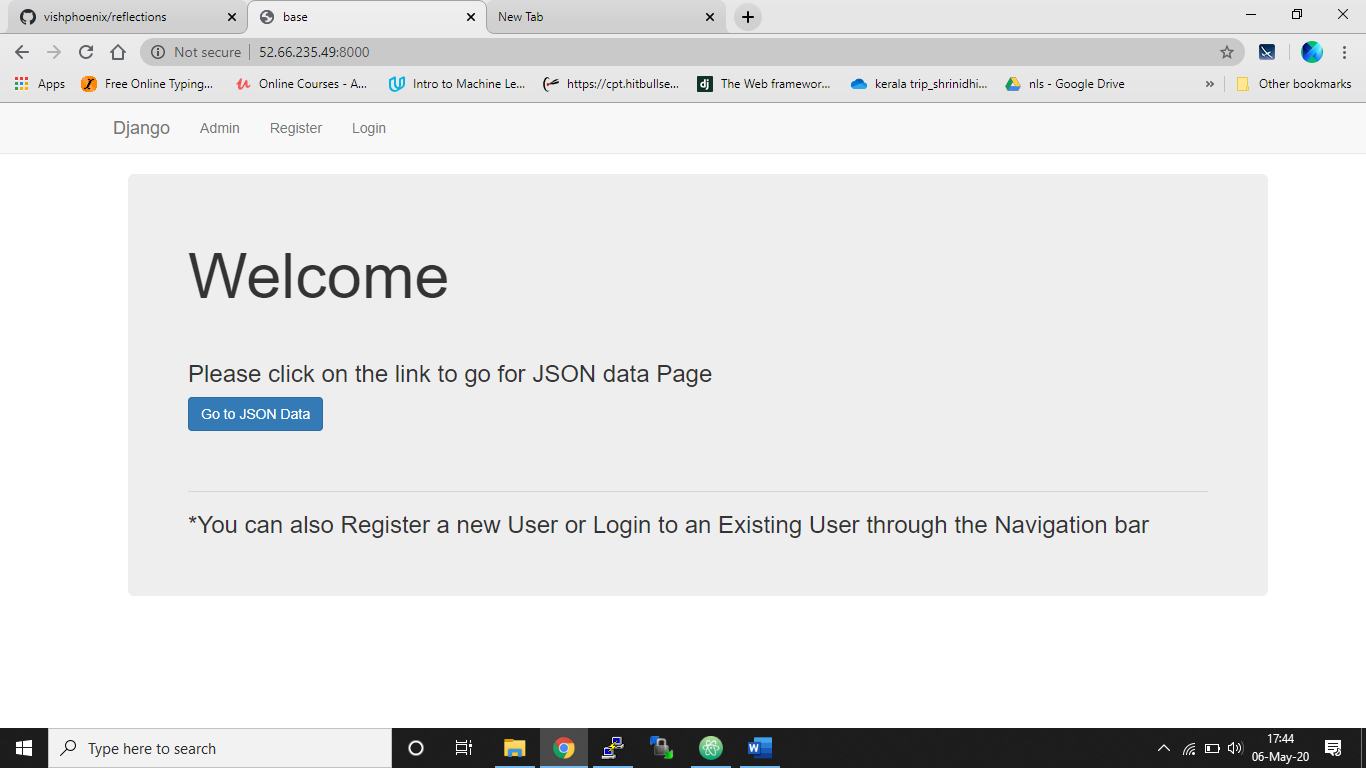
**Code:**



Code Screen

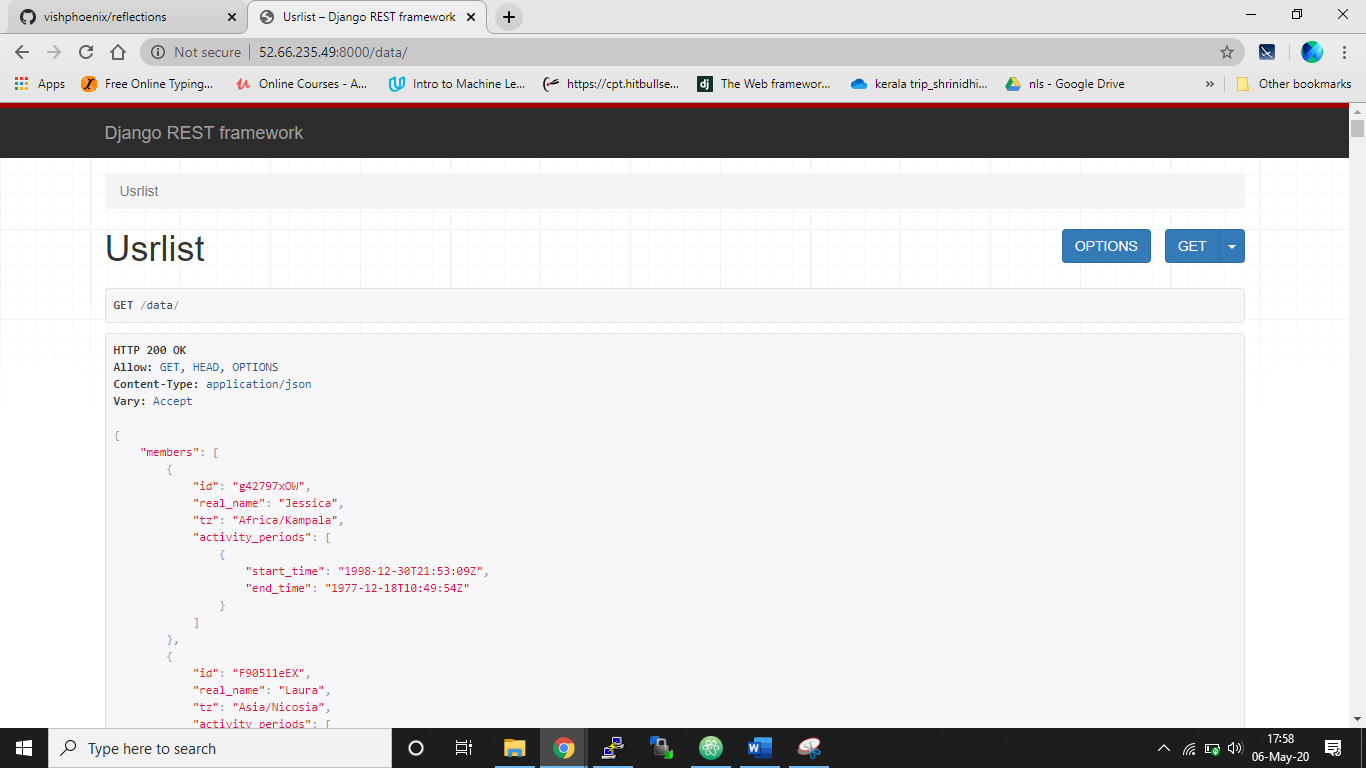
* The code is entirely developed in Python and Django.
* As it can be seen from the above screen users is the main project folder.
* basicapp : All the application related code such as model.py, views.py, admin.py, serializers.py of which are the few.
* Media : Basically for storing media related file.
* Static : Basically for storing static files such as css scripts, js, image files etc.
* Template : All the html files are stored here.
* Users : Project related files such as setting.py, urls.py, wsgi.py
* Manage.py : Most import file in the Django command line utility for administrative tasks.
* Populate\_script : pyhton script for populating the data with Faker library.

**UI**

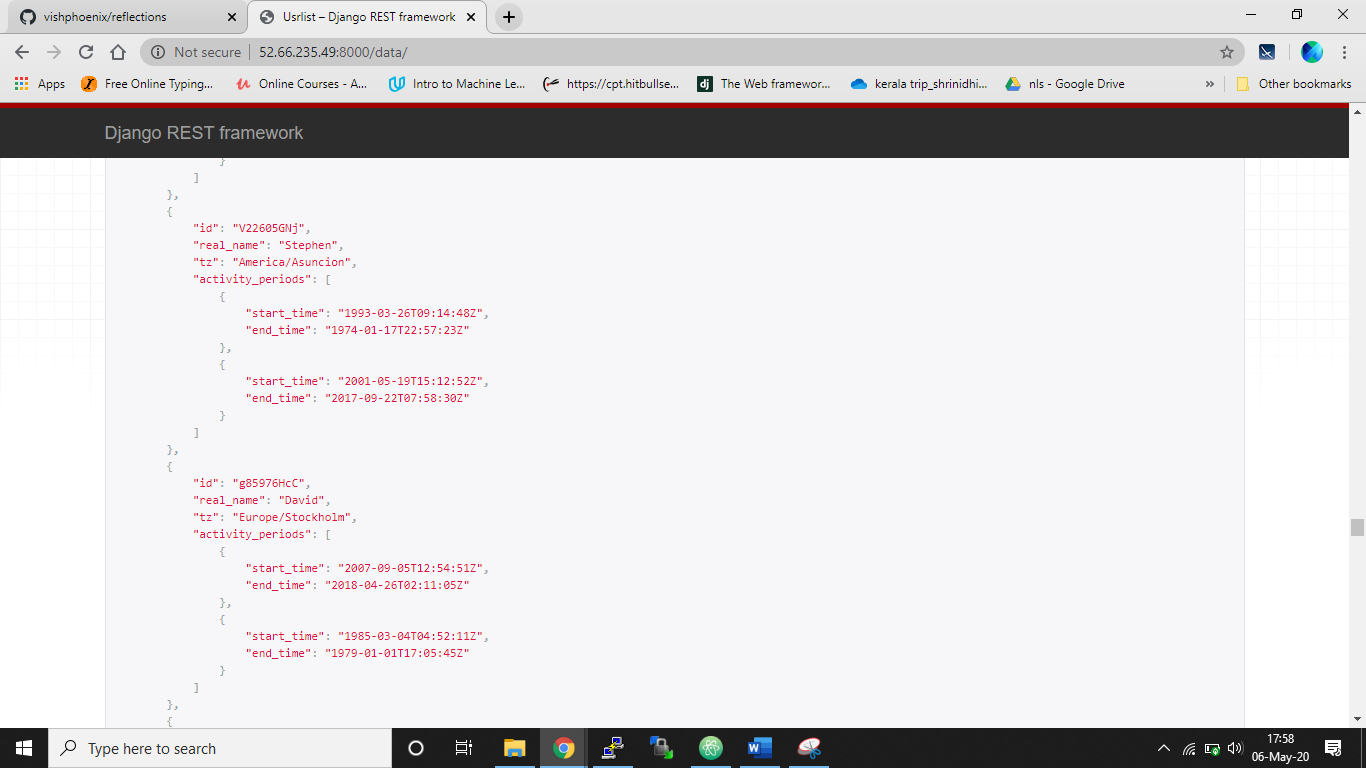


Home Page

* Home Page is first screen to come across.
* It consists of a Navigation bar containing links to Registering for a new user or logging in to an existing user and a link to Django Admin page.
* And then there’s a button that takes to the json data page



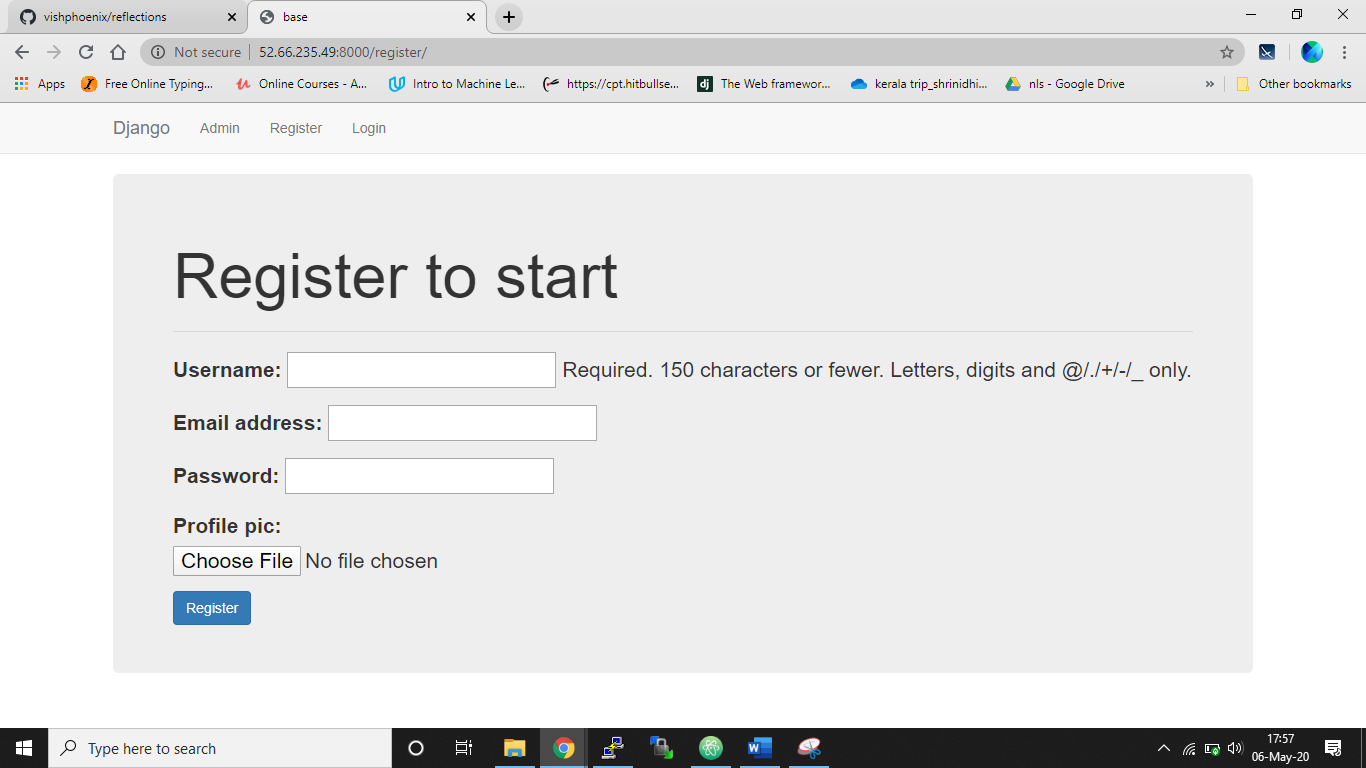
JSON Data Screen



Json Data Screen Continued…

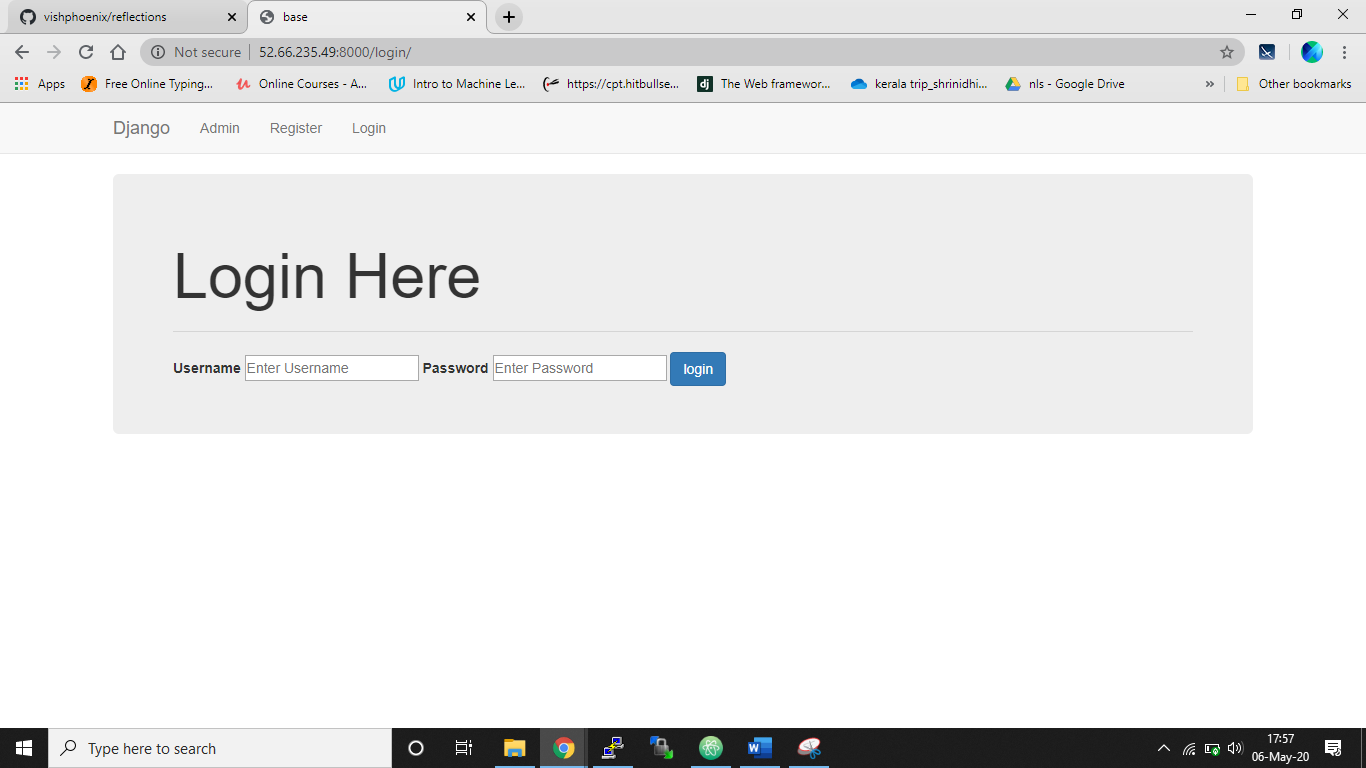
* The JSON data screen is designed as it was specified in the task set.
* It uses Faker library and Rest API (Serializer) to output the data in JSON format.
* It can be seen as the data is dummy it is distributed randomly.

**Few Optional Screens**



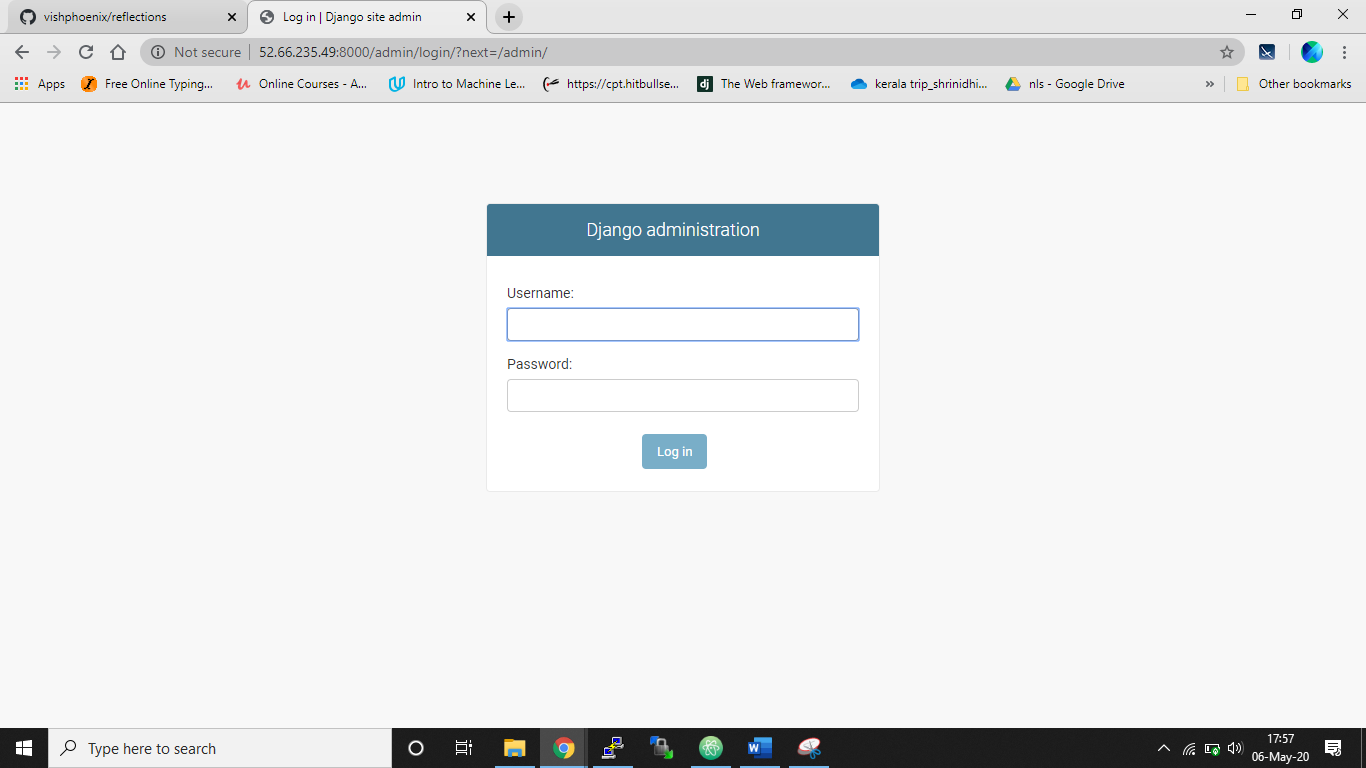
Register Screen

* If any user wants to use the features of registering and login he can opt for it, the user is saved in the database once he/she is registered.



Login Screen

* Once the user is registered he can just login and logout and perform the operations which he desires.



Django Admin Screen

* Django admin screen, to verify the data from the backend part.