Assignment Name: Shrreenithi Srinivasan

1.

a. System Knowledge: I have attached the **script file - network.sh.** To check if the Network is up, I connected to Google.com server to check the connectivity.I also **used chmod +x /init.d/network.sh** to make the file executable and to start the script at start of the system.

b. I have attached **service.sh script file** to extract ip address from text file and logging to each machine from farm of Ubuntu machines by doing ssh and then restarted fooBar service.

2.

* I created Restful API using Python flask to allow common user operations.

I first started off my application by separating into four modules such as model.py, resource.py, views.py and run.py.

**Run.py**- To start the server and contains the main script to run the application called resttest, it will add the corresponding service and call the resource class operations to generate data to be viewed in output.

**Resource.py**- contains 4 classes, namely:

*class UserReg* for user account creation (signup).

*class UserLogin* to check for returning user to login.

*class User* - Browsing users and view user with occupation and age details. Get function is used to see the user in database by giving the name of that user in browser, Post function is used to create a user with details of age and occupation, Put and Delete function updates the deletes the selected user.

*class SecretResource*, is used to authorize by giving adding acces token to authorization header and return the value.

**Models.py**- This module contains the database to access and store the information. I used Database SQLAlchemy which is an extension of SQLlite.

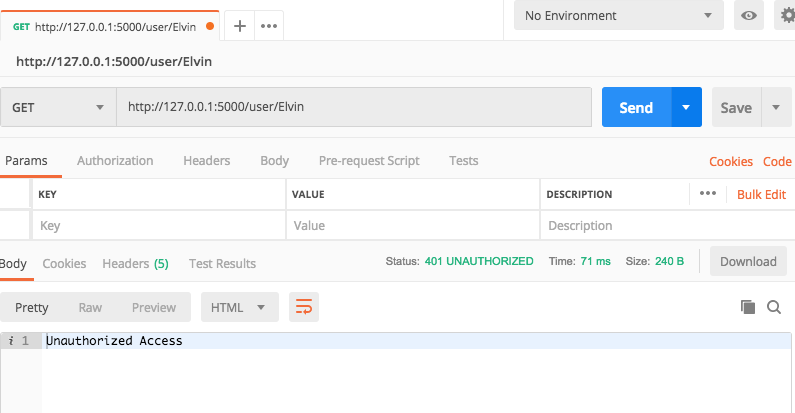
**View.py** – This module routes the output/ web application and to display the user information from database.

* DB used is SQLAlchemy extension of SQLlite in models.py file.
* Deployment Instruction: I used Postman application and browser to view the output. To view the output in terminal I used: curl -X GET http://localhost:5000/

Outputs:

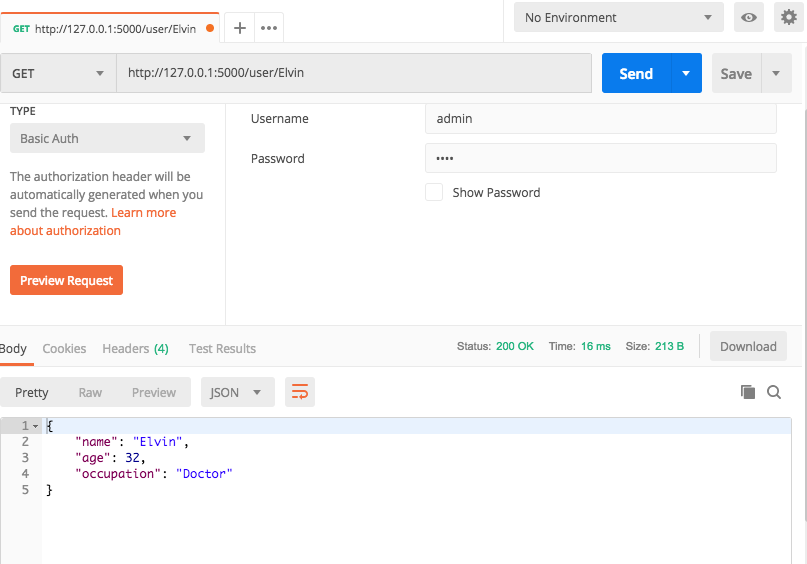
TO **authenticate** and display the user detail Elvin by using Get function.

Since user name and password not provided, displaying Unauthorized access:



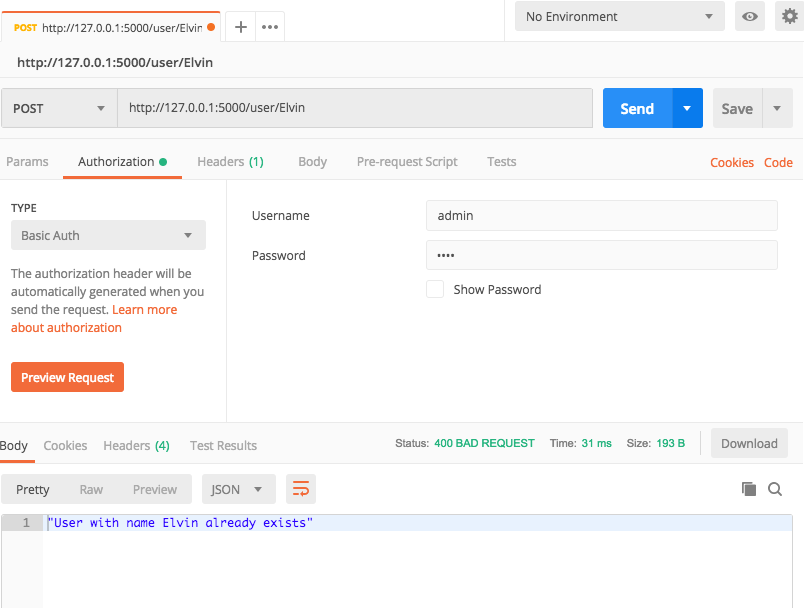
TO authenticate and display the user detail Elvin by using Get function.

Since user name and password is provided, displaying User details:



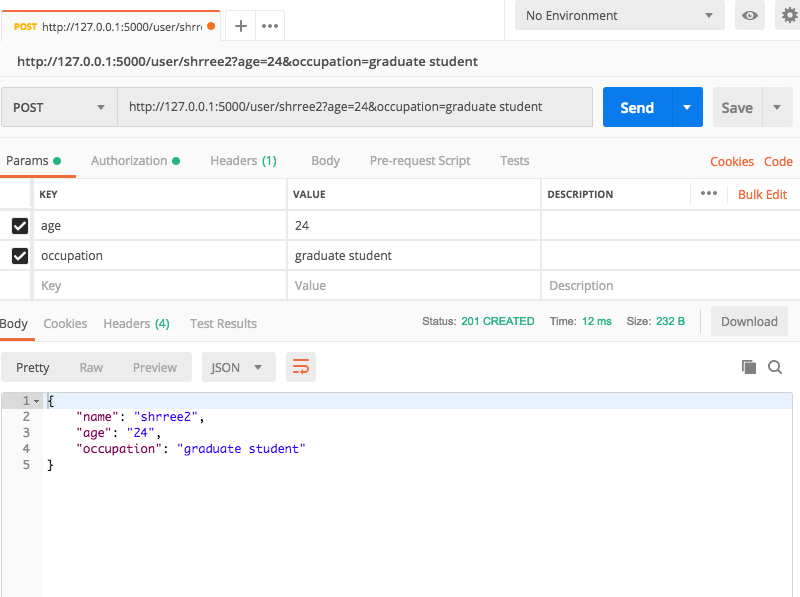
TO authenticate and create the user detail Elvin by using Post function.

Since user name and password is provided, displayed message for existing user.



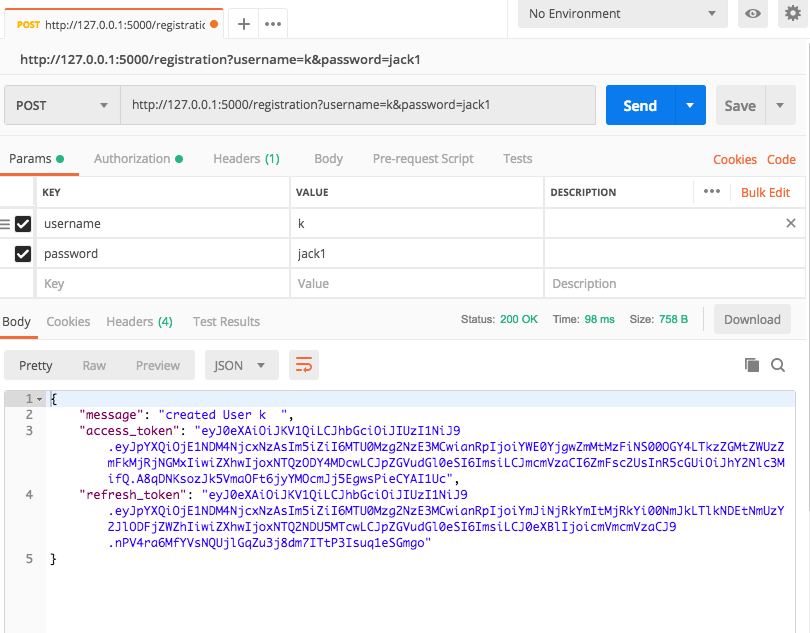
TO authenticate and create the user detail shrree2 by using Post function.

Since user name and password is provided, creating new user information with age and occupation.

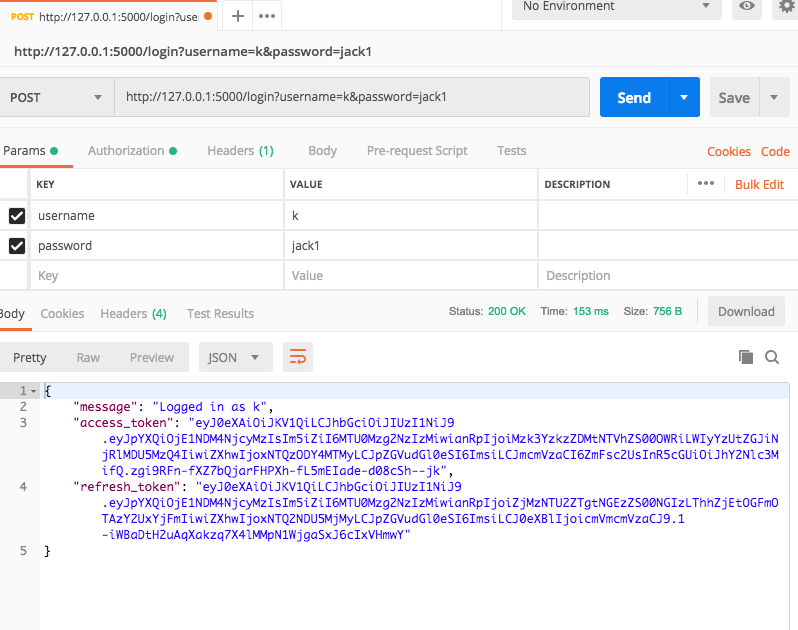


User Registration class – **user account creation**

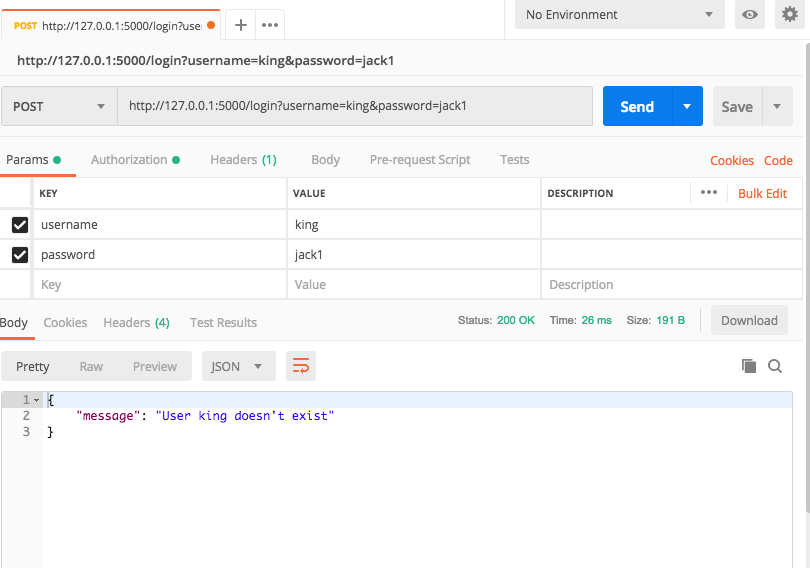
To create user with username and password to sign up and generates token.



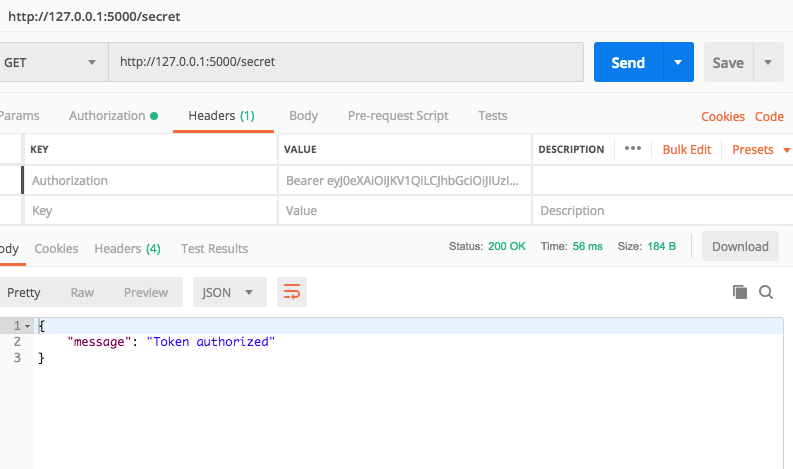
Login class- To check if user is already signed up or to throw message saying wrong username and wrong credentials. Here it is displayed as user K already logged in



Login class- To check if user is already signed up or to throw message like wrong username and wrong credentials. Here it is displayed as user king is not registered.



Secret class- **Authorization**, to check if using the token access the particular module by the given user:



Web App: - Signup Application.

