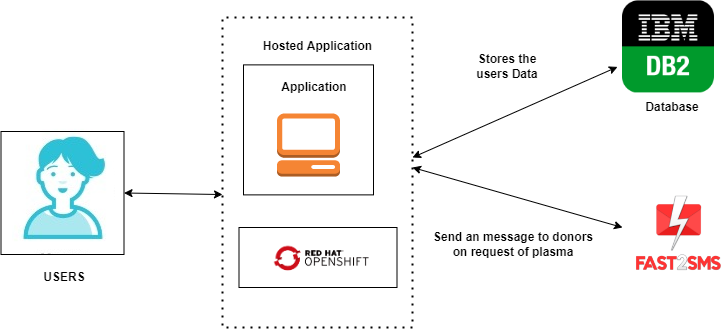
**Plasma Donor App**

**Project Description:** During the COVID 19 crisis, the requirement for plasma became high and the donor count being low. Saving the donor information and helping the need by notifying the current donors would be a helping hand. In regard to the problem faced, an application is to be built which would take the donor details, store it and inform them upon a request.

Users need to register an account and login to the application. Once the user logins, he will have a dashboard to view the total number of donors and count of people with specific blood groups. User will have the option to request the blood. Once the user requests, all the people with that blood group will be notified with an SMS. **Technical Architecture:**



**Learning Outcomes:**

By the end of this project:

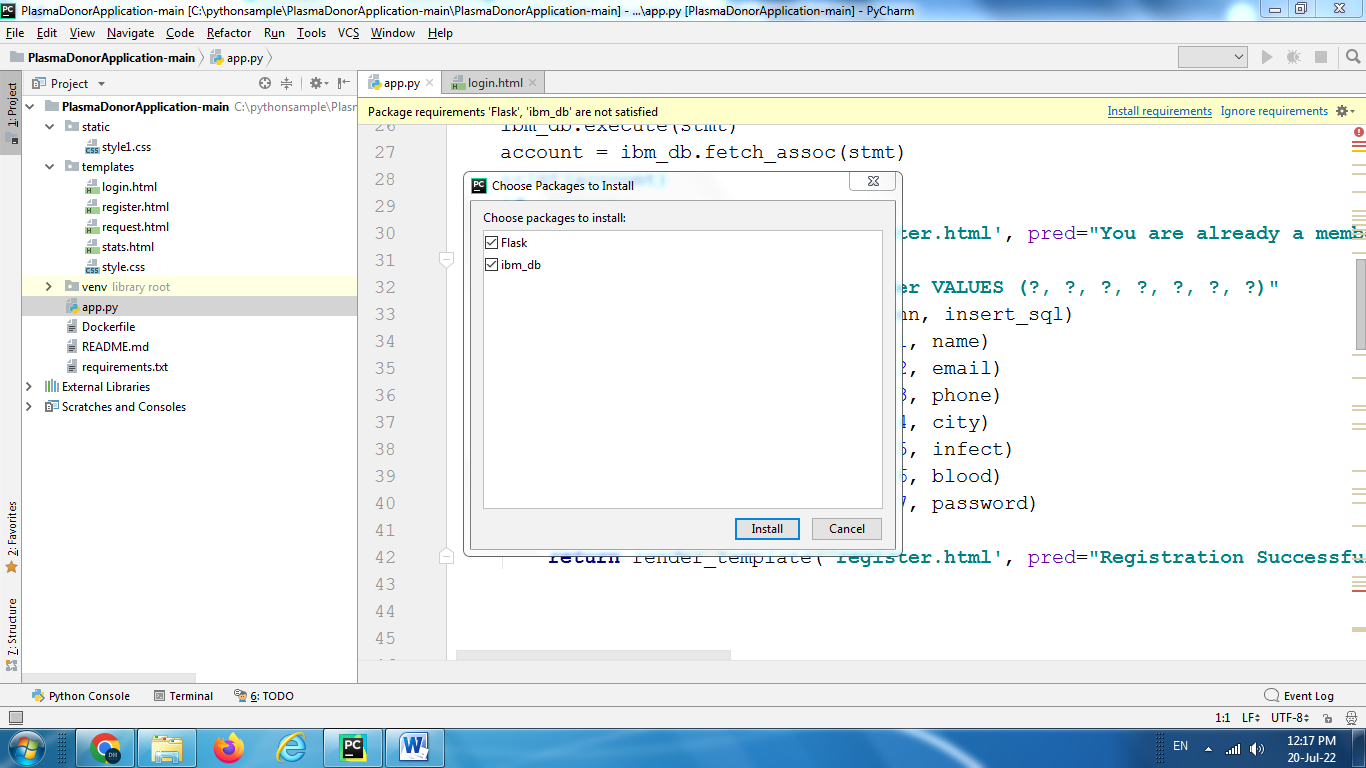
* You’ll be able to work on IBM DB2, creating flask application,making an application into an docker image and deploying app in Redhat Openshift dev space.
* Build a flask application which will take the user inputs, update the IBM DB2 database and notify the user upon request.

**Project Workflow:**

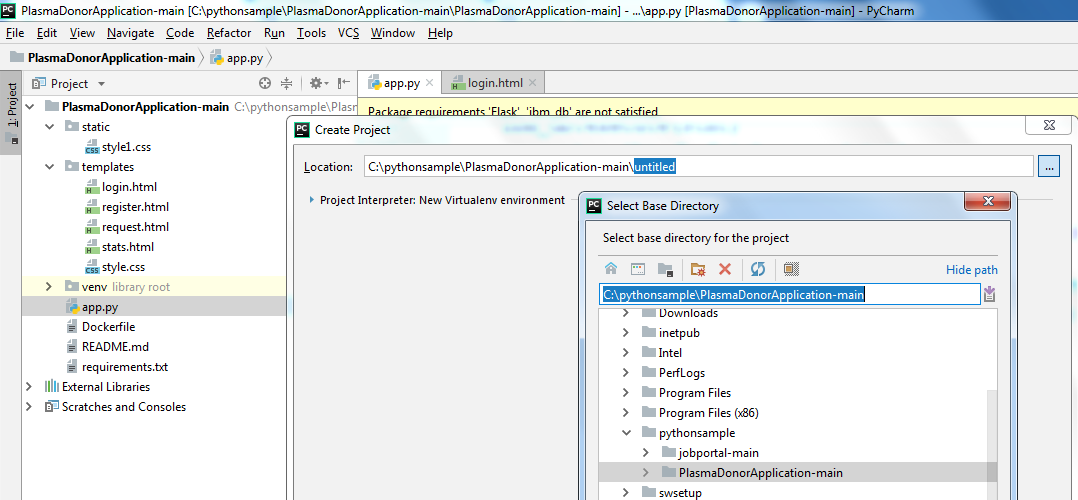
* User interacts with the application.
* Registers by giving the details as a donor.
* Database will have all the details and if a user posts a request then the concerned blood group donors will get notified about it.
* Create the docker image for the application and deploy on Redhat Openshift dev space.

### Setting up Application Environment

**I have downloaded and install python,pycharm**



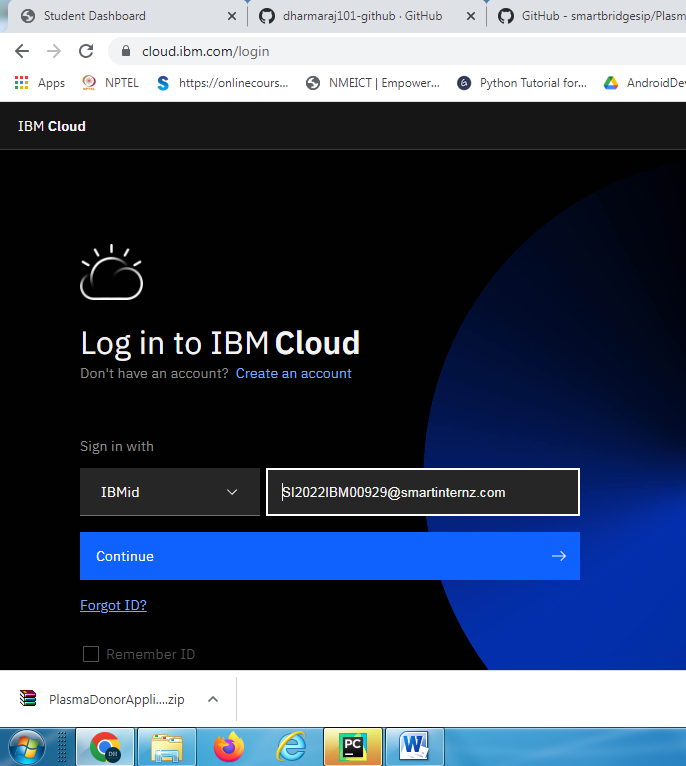
and then download the given source code from the git repo and extract it in the pythonsample folder and thereafter I opened project folder in the pycharm IDE as new project to edit.

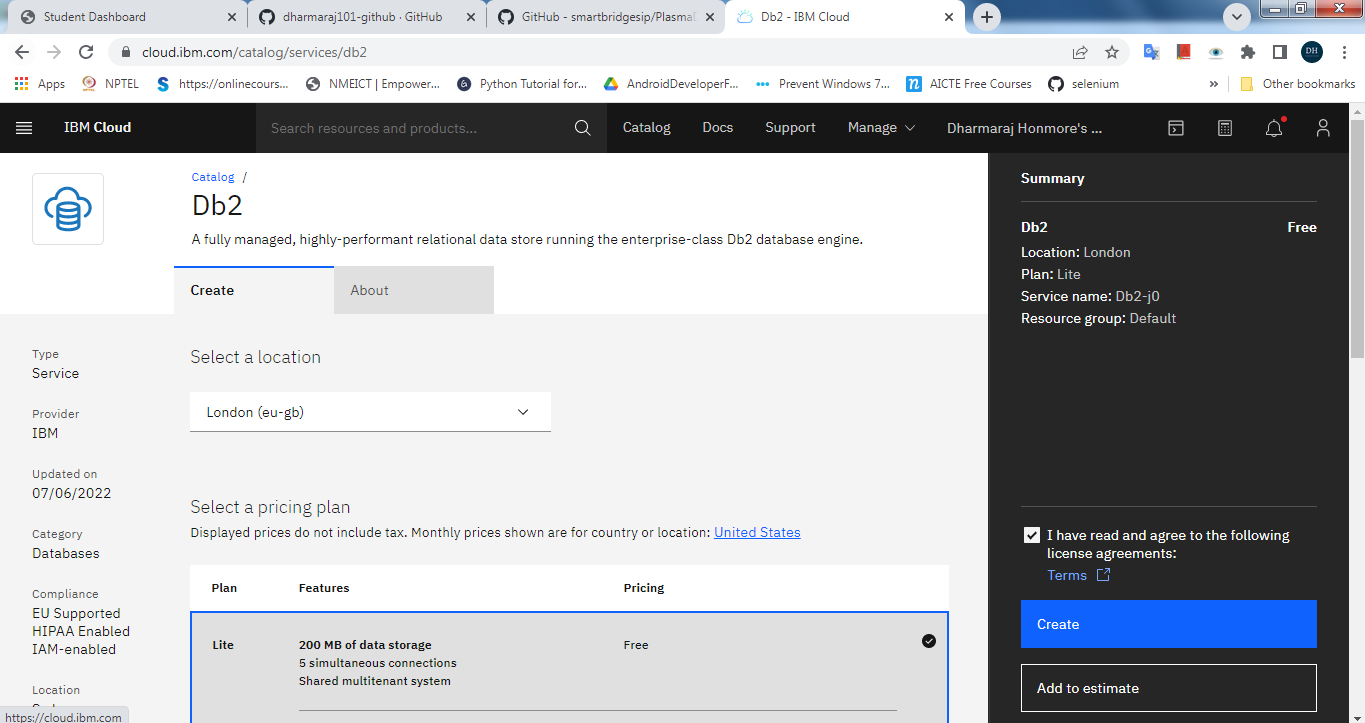


**Click Ok->create and installed Flask and ibm\_db**

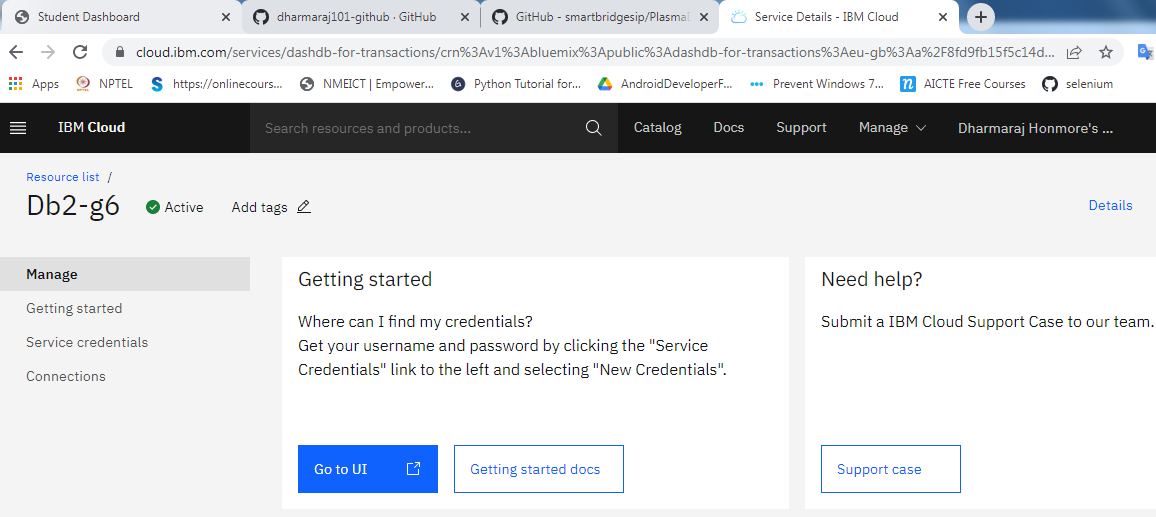
Creating Database on IBM Cloud

I logged in IBM Cloud using credentials

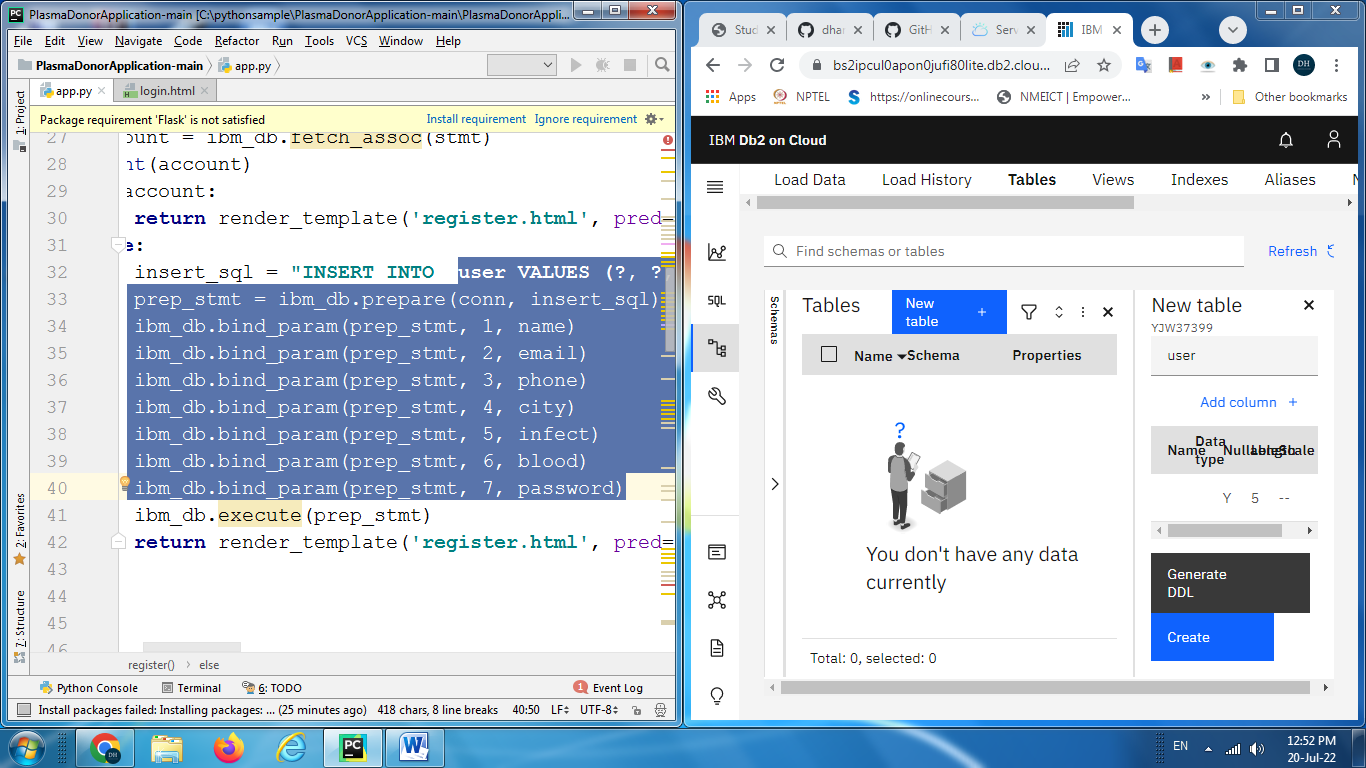


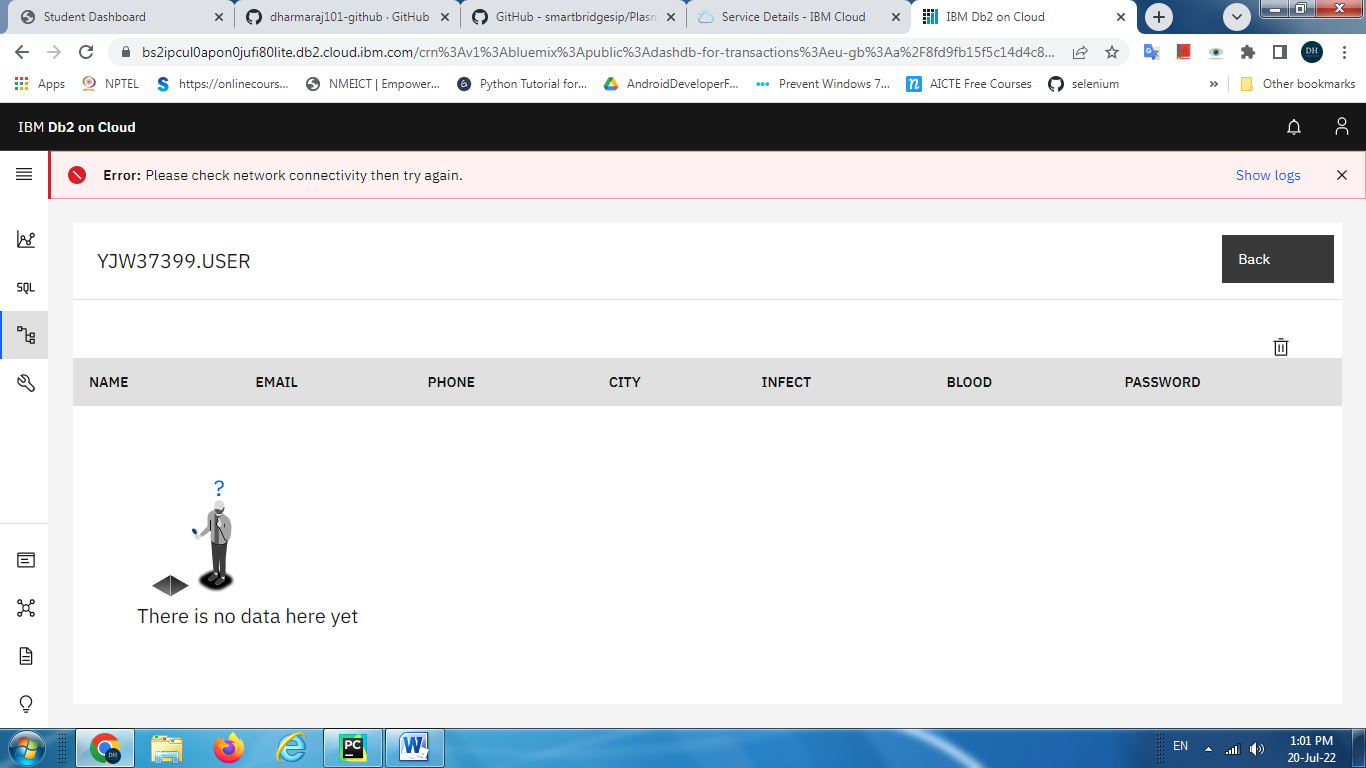


Click on “Create” and then Click on “**Db2-g6” database service** and the lick **Go to UI button**

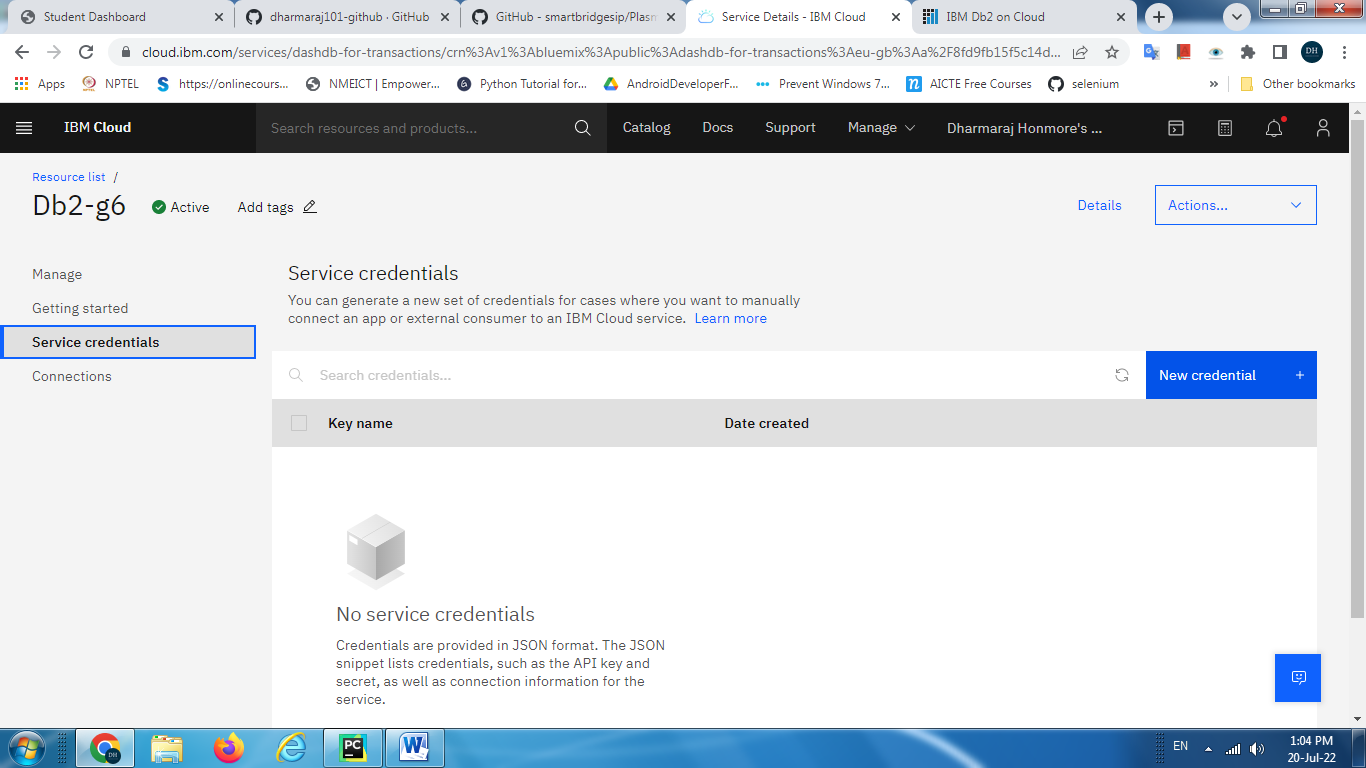


And the create the “**user**” table for current schema and add the columns in the table by observing the code shown in following screenShot.

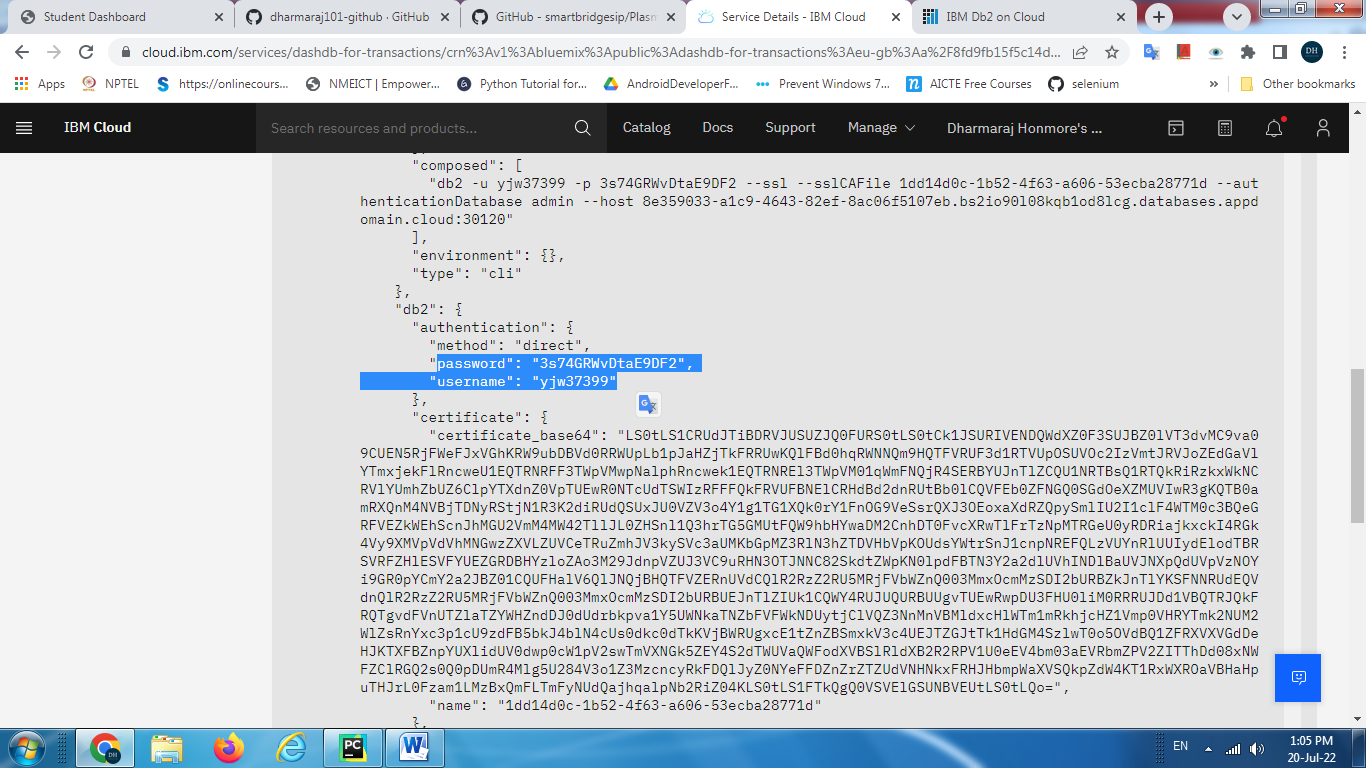




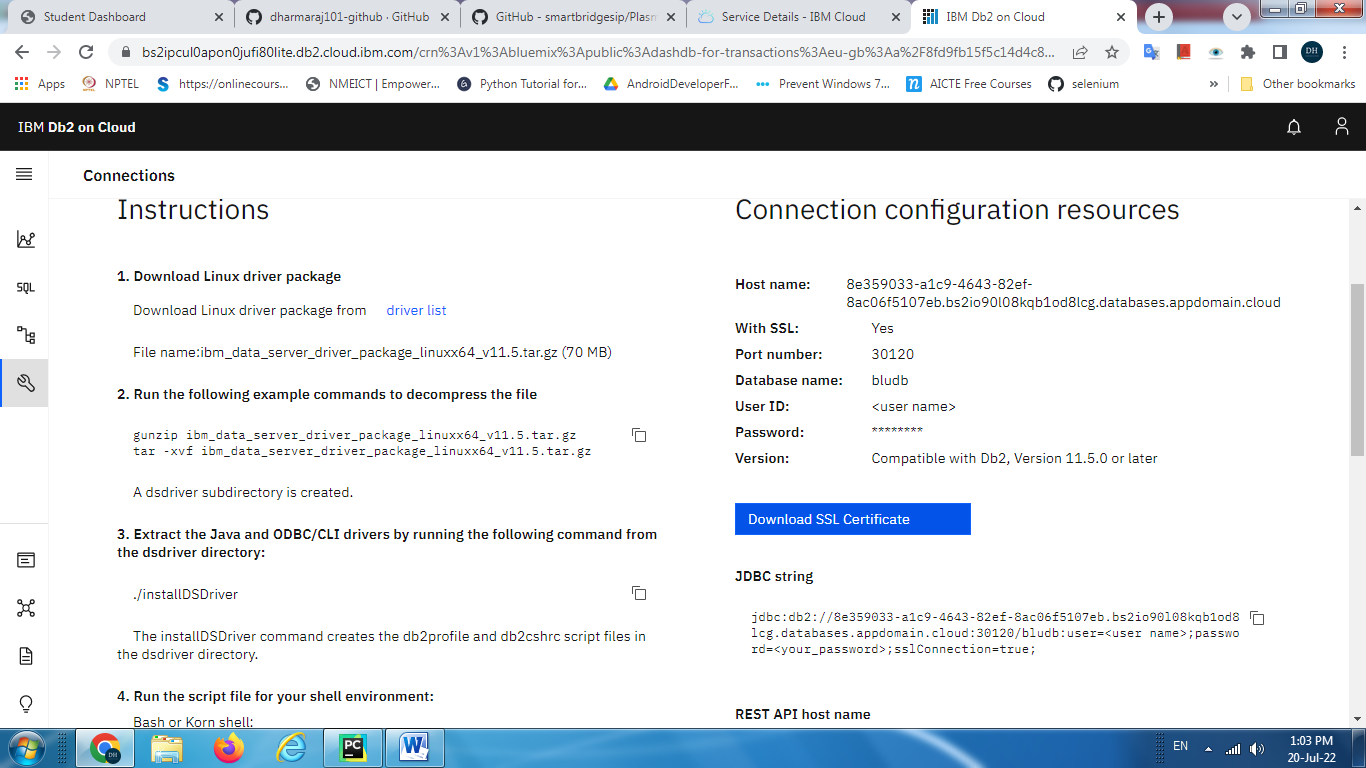
Add the new service credentials click on **New credential** andclick **Add**

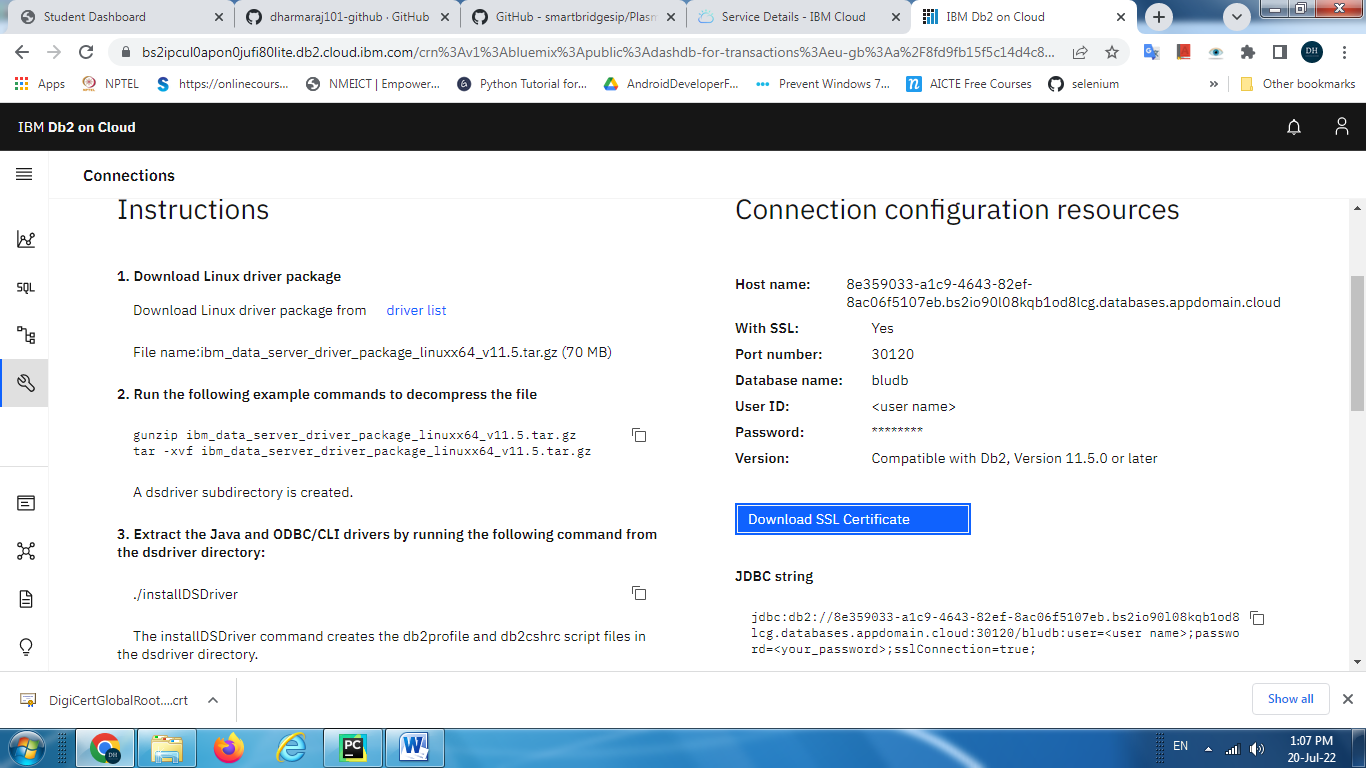


Click on the down arrow of service credential to copy username and password and inser it in the app.py

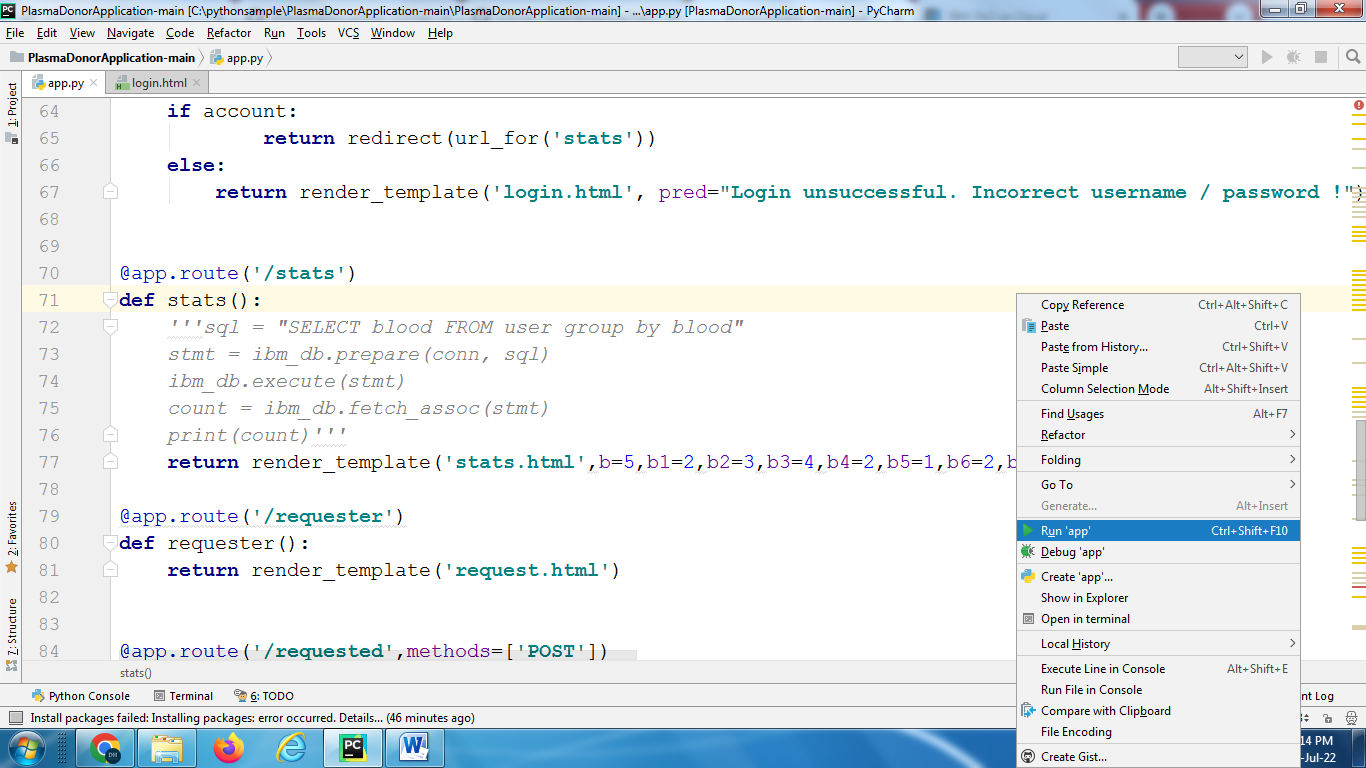


And also copy the connection string info (hostname,port number) from **connection configuration resources** for windows then download and copy the **digital SSL certificate** in the project folder

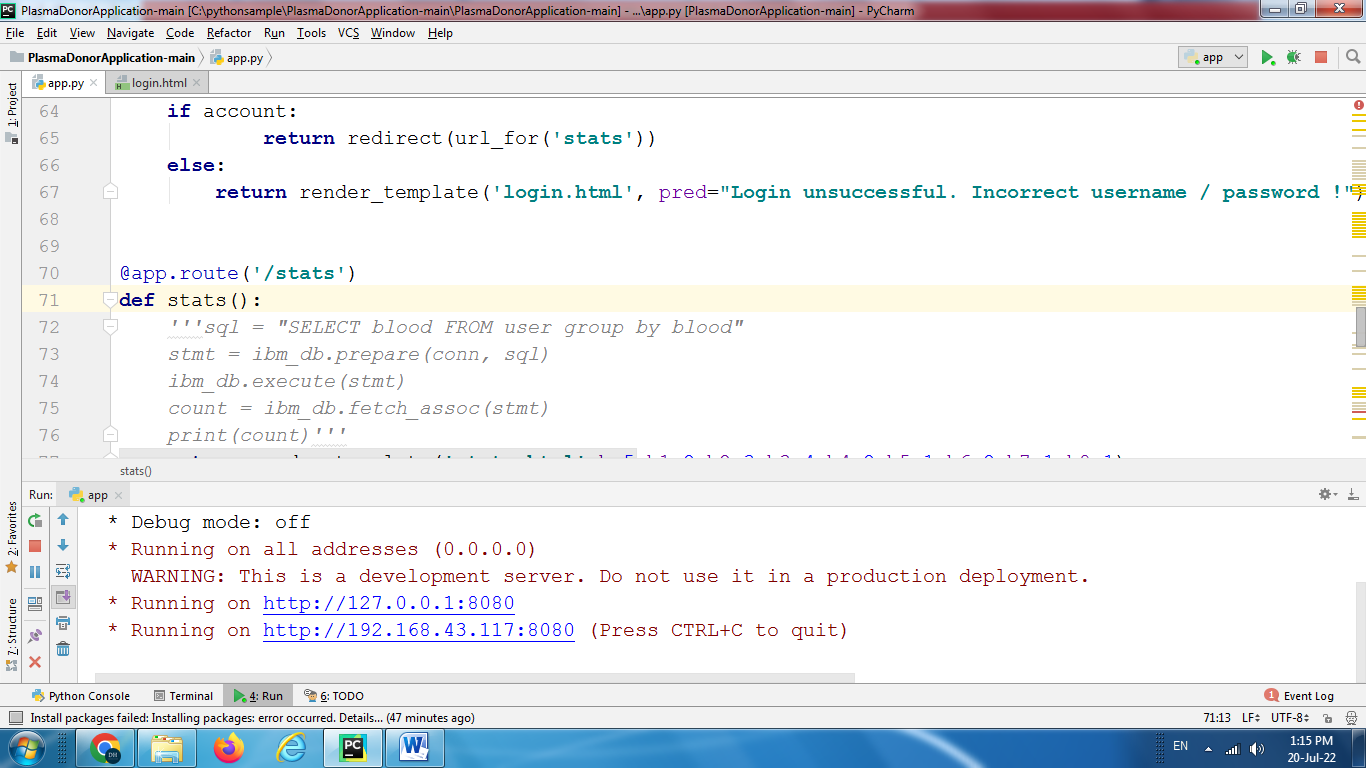


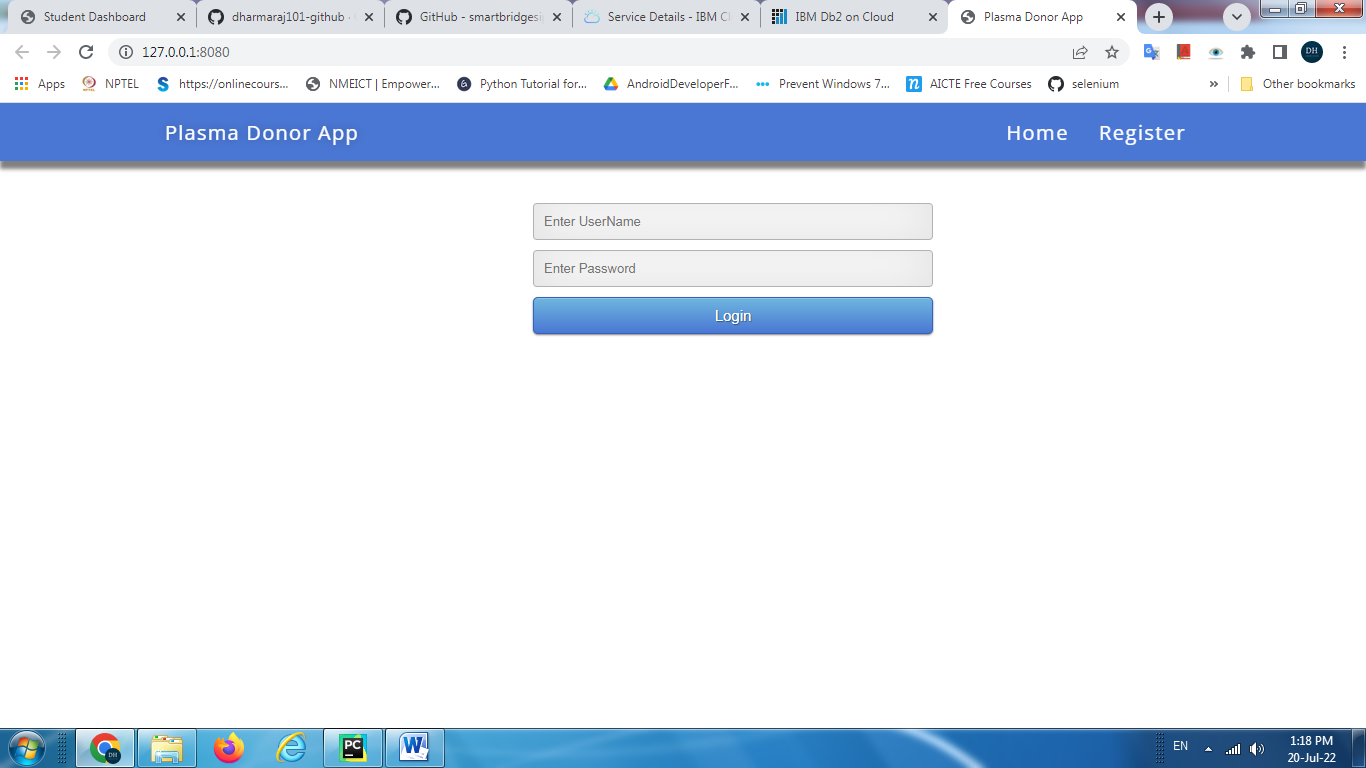


Run the **app.py**

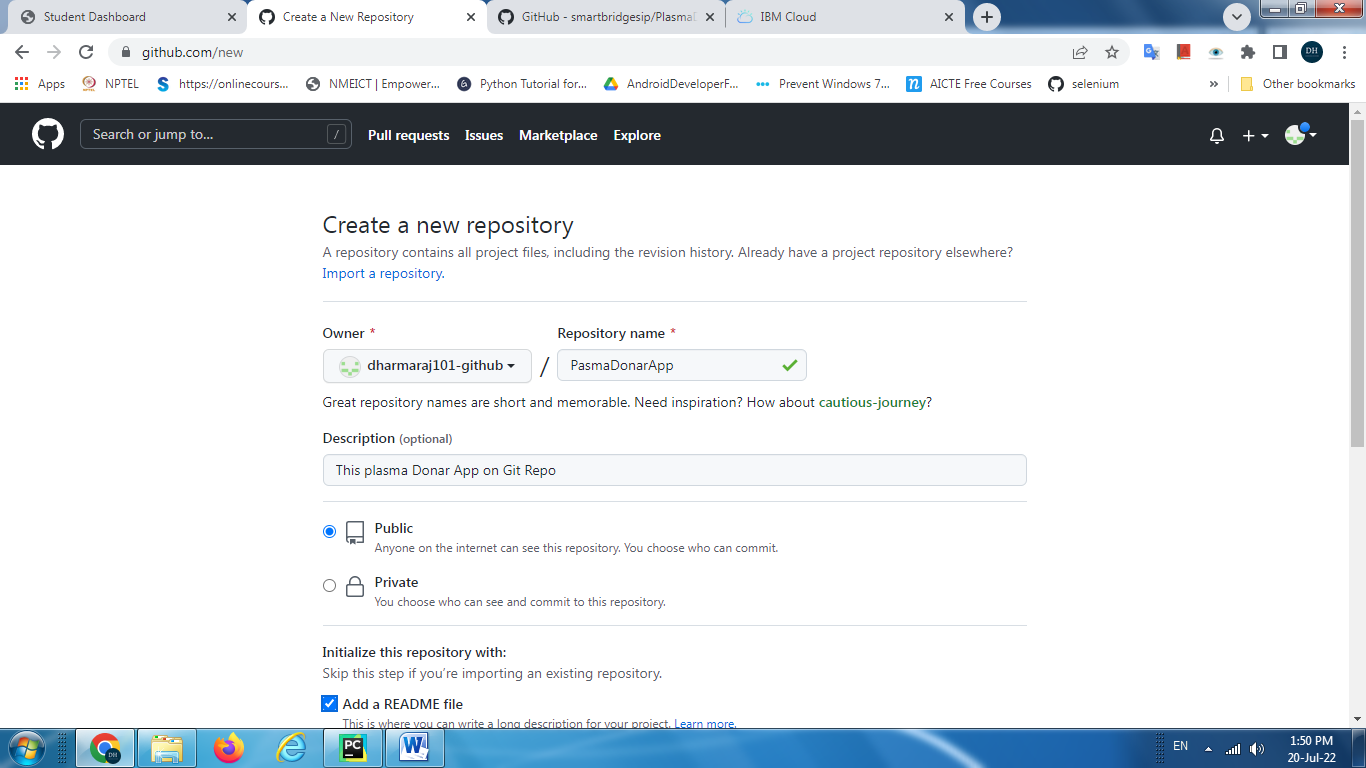


Click on <http://127.0.0.1:8080> to see the application in the browser

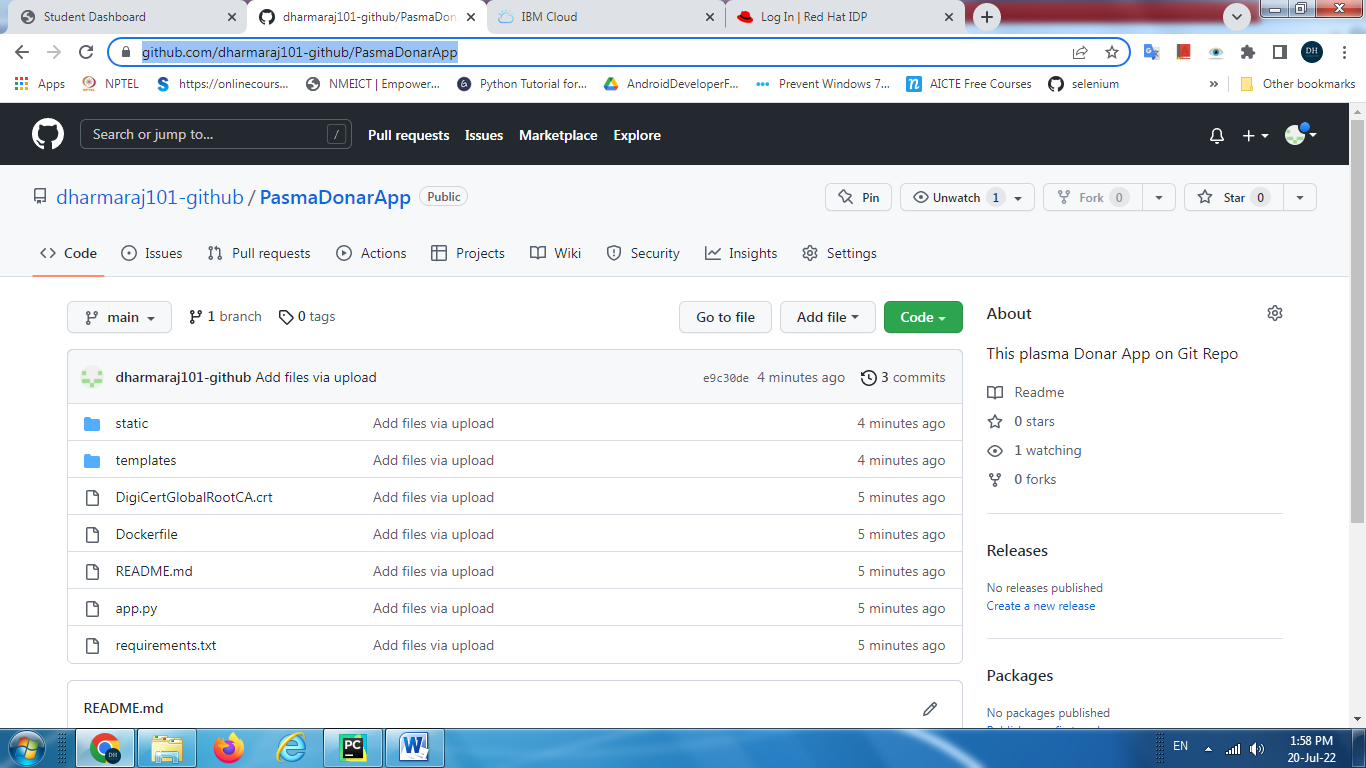




**Uploading source code on Github Repository**

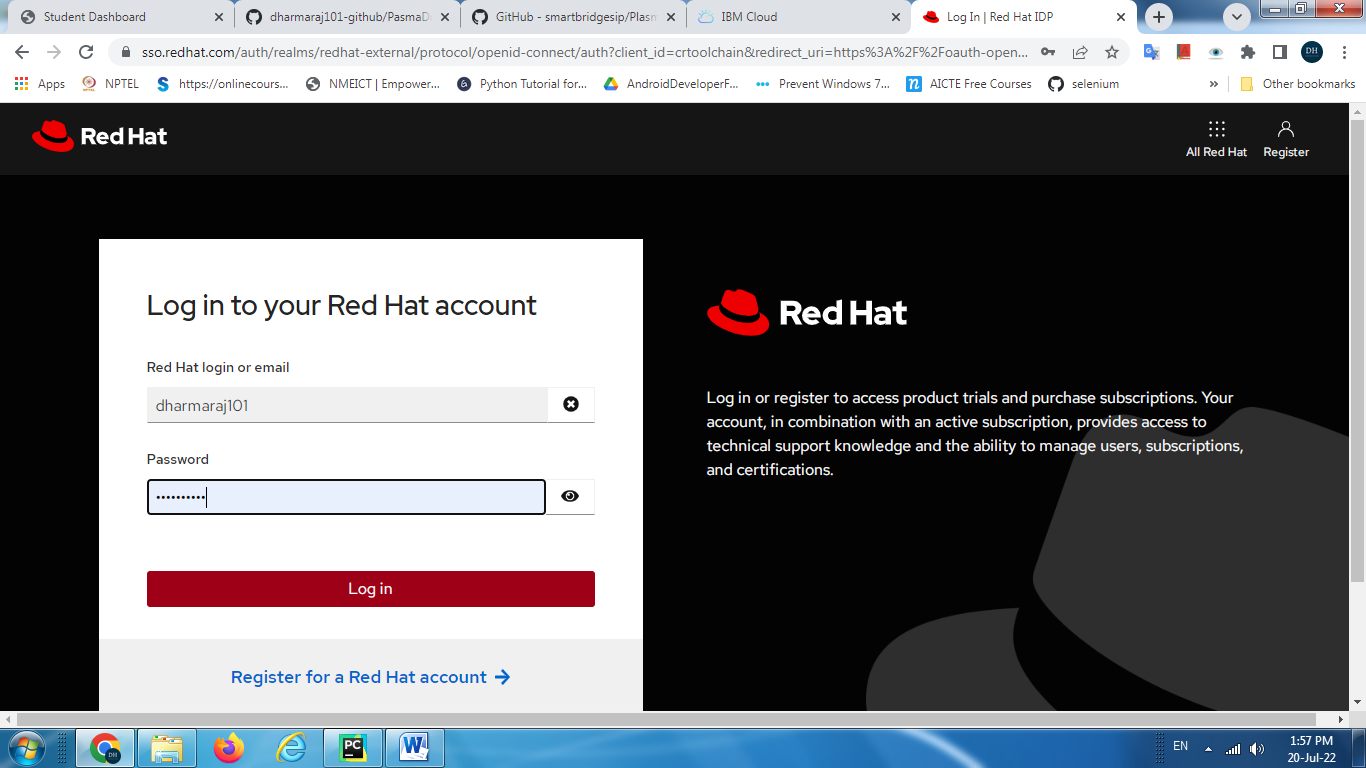


Click on **Add File🡪upload files** then **drag and drop** the files



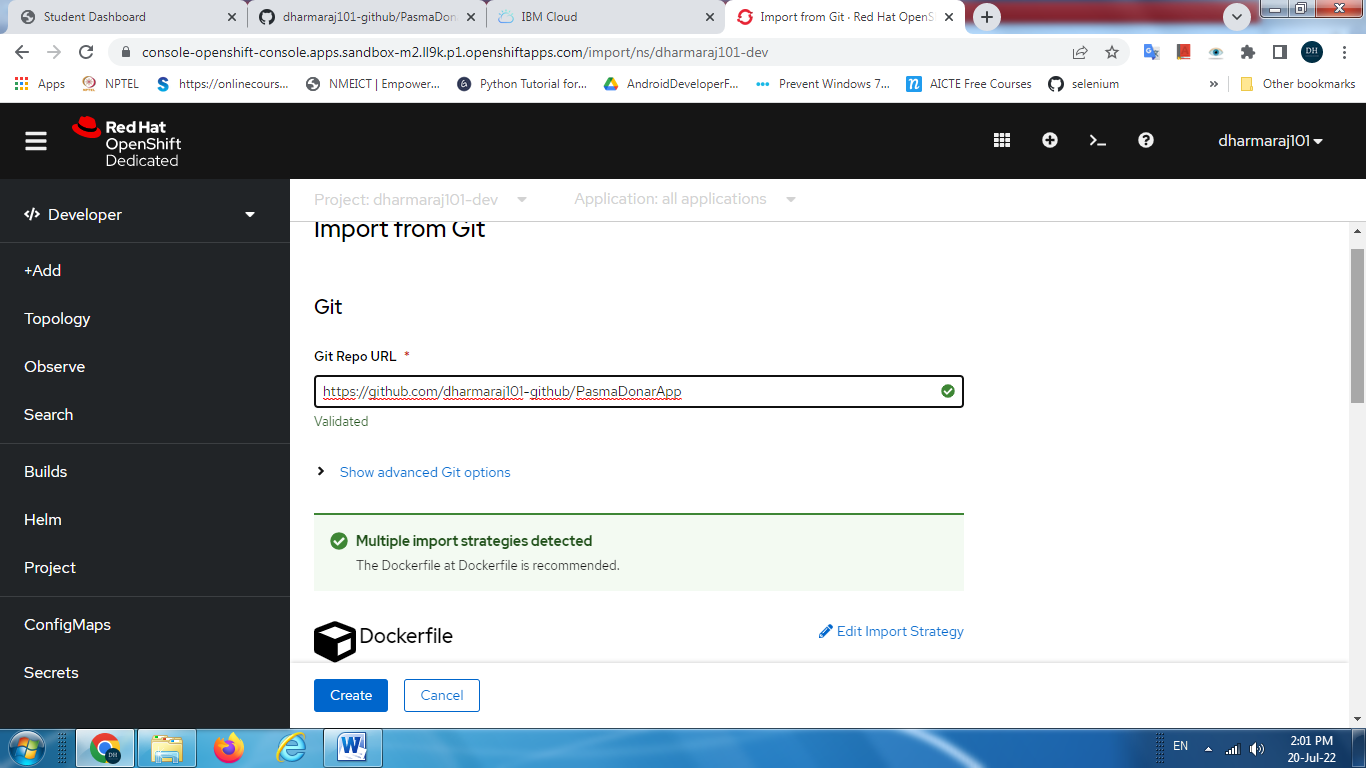
### Deploy the App on Open Shift Red Hat Dedicated

### Log in to open shift dedicated console

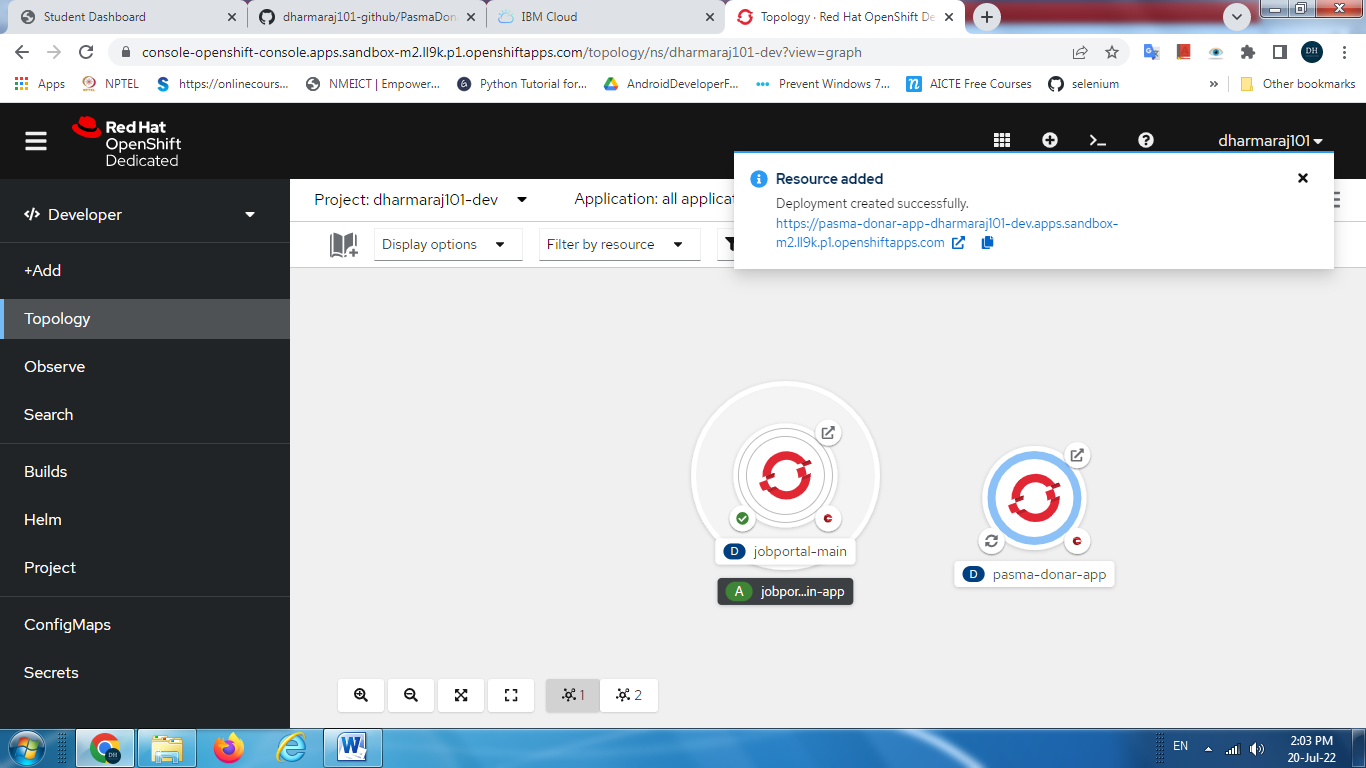


Select”**Developer”** Choose **Add🡪import from Git**

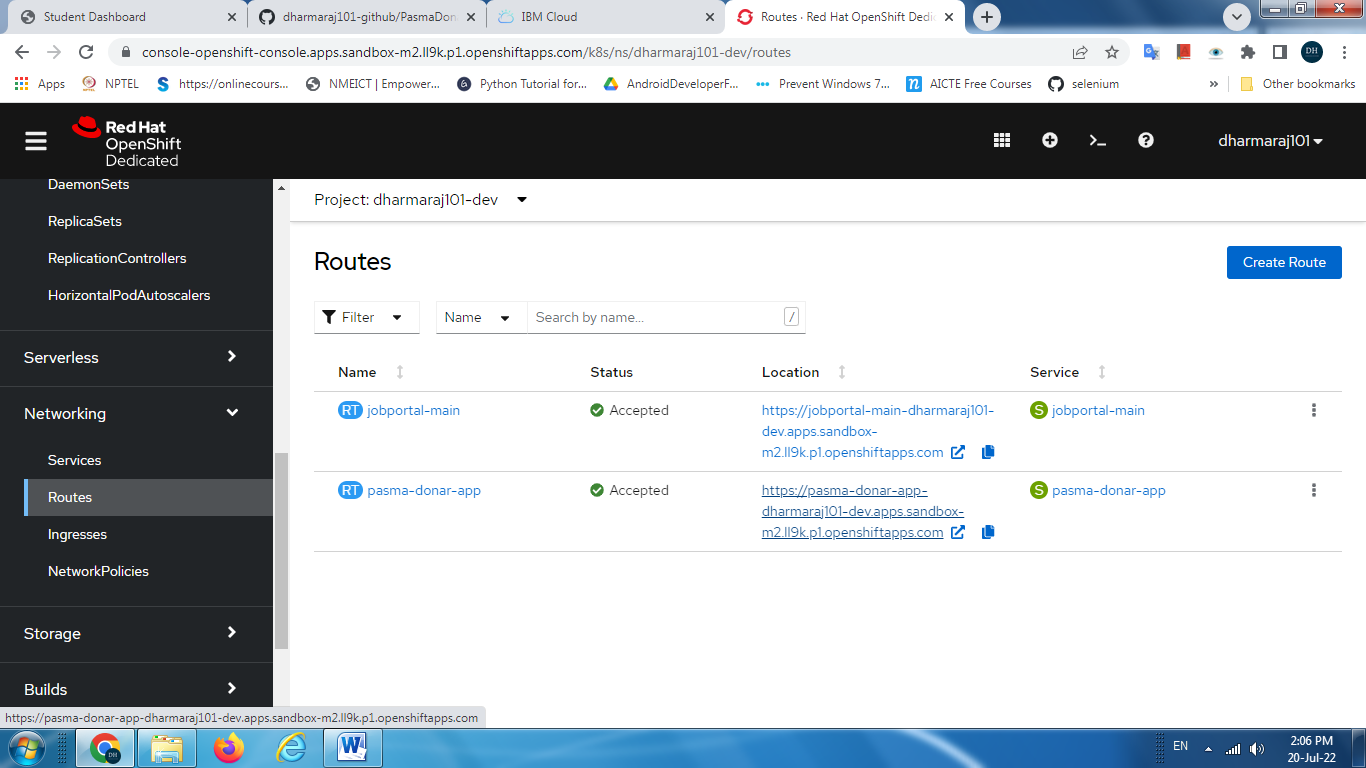
**Copy and paste git repo URL**



Click on **Create** and observe the created **pods services and routes** etc.



Click on the location URL in the route tab to see the application on the browser





**App URL**:

[**https://plasma-donar-app-dharmaraj101-dev.apps.sandbox-m2.ll9k.p1.openshiftapps.com/**](https://plasma-donar-app-dharmaraj101-dev.apps.sandbox-m2.ll9k.p1.openshiftapps.com/)

**Git Repo Address:**

[**https://github.com/dharmaraj101-github/PlasmaDonarApp**](https://github.com/dharmaraj101-github/PlasmaDonarApp)