Name: Manasi Choche

Institute: A. P. Shah Institute of Technology, Thane

Application name: Plasma Donor App

GitHub link: https://github.com/manasi1512/PLASMA_APP.git

Demo Video Link: https://drive.google.com/file/d/12Qw9FnC1GwxivwR25FvhfDNc4-uRQL2_/view?usp=sharing

Description:

Plasma donor app is created for users who needs issue plasma requests.

For which user needs to register in the app.

User is supposed to login through credentials.
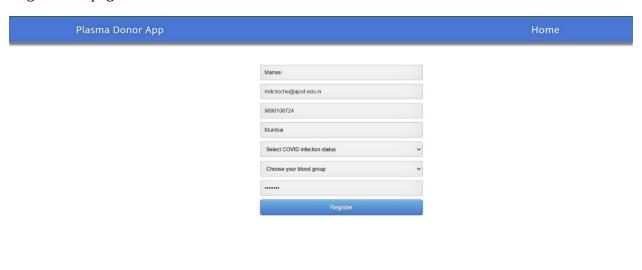
After logging in user can request for plasma.

To complete this task following steps have been performed.

1.Setting up application environment.

      a. Install python

      b. Install pip

      c. Install flask

      d. Install ibm_db

2.Implementing web application

      a. Imported python code to python idle

      b. Updated host name, port number, uid, password in the python code, made sure db name is similar

      c. Downloaded SSL digital certificate

      d. Created IBM cloud database

      e. Created table named user

      f. Created 7 columns with VARCHAR data type in table

      g. Executed code with python app.py

      h. Tested our code on localhost:8080/login

3.Uploaded source code on GitHub

      a. logged in to own GitHub account

      b. Created public repository and imported my code into that

      c. Taken link of our GitHub repository

4.Deploy the app on OpenShift red-hat dedicated

      a. Under the developer tab go to Add àImport from Gità Pasted GitHub linkàCreate

      b. Go to topology in which we can check the status of building the app

      c. One the build completed we can open the URL

      d. Our app is running successfully.

Screenshots:

Registration page:

Login page:

Plasma Donor App   Home   Register

Enter UserName

Enter Password

Login

User interface after login:

Plasma Donar App   Home   Request

5
Donors

2
O Positive

3
A Positive

4
B Positive

2
AB Positive

Code:

File   Edit   Format   Run   Options   Window   Help

```python
from flask import Flask, render_template, request, redirect, url_for, session
import ibm_db
import json
app = Flask(__name__)

conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90108kqblod8lcg.databases.appdomain.cloud;PORT=31498;SECURITY=SSL;SS

@app.route('/registration')
def home():
    return render_template('register.html')

@app.route('/register',methods=['POST'])
def register():
    x = [x for x in request.form.values()]
    print(x)
    name=x[0]
    email=x[1]
    phone=x[2]
    city=x[3]
    infect=x[4]
    blood=x[5]
    password=x[6]
    sql = "SELECT * FROM user WHERE email =?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt,1,email)
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    print(account)
    if account:
        return render_template('register.html', pred="You are already a member, please login using your details")
    else:
        insert_sql = "INSERT INTO  user VALUES (?, ?, ?, ?, ?, ?, ?)"
        prep_stmt = ibm_db.prepare(conn, insert_sql)
        ibm_db.bind_param(prep_stmt, 1, name)
        ibm_db.bind_param(prep_stmt, 2, email)
        ibm_db.bind_param(prep_stmt, 3, phone)
        ibm_db.bind_param(prep_stmt, 4, city)
        ibm_db.bind_param(prep_stmt, 5, infect)
        ibm_db.bind_param(prep_stmt, 6, blood)
        ibm_db.bind_param(prep_stmt, 7, password)
        ibm_db.execute(prep_stmt)
```

Ln: 1 Col: 0

31°C
Haze

ENG
IN

17:40
31-07-2022

Deployment status