Project Development - Delivery Of Sprint-3

Team ID	PNT2022TMID52216
Project name	Plasma donor application
Team member	Shahin A Lakshmi M Santhiya R Aswathy M A Kokila R

style.css:

```
@import url(https://fonts.googleapis.com/css?family=Open+Sans);
.btn {
   display: inline-block;
   *display: inline;
   *zoom: 1; padding:
   4px 10px 4px;
   margin-bottom: 0;
   font-size: 13px;
   line-height: 18px;
   color: #333333;
   text-align: center;
   text-shadow: 0 1px 1px rgba(255, 255, 255, 0.75);
   vertical-align: middle;
   background-color: #d70c0c;
   background-image: -moz-linear-gradient(top, #ffffff, #e6e6e6);
   background-image: -ms-linear-gradient(top, #ffffff, #e6e6e6);
   background-image: -webkit-gradient(linear, 0 0, 0 100%, from(#ffffff),
to(#e6e6e6));
   background-image: -webkit-linear-gradient(top, #ffffff, #e6e6e6);
   background-image: -o-linear-gradient(top, #ffffff, #e6e6e6);
   background-image: linear-gradient(top, #ffffff, #e6e6e6);
   background-repeat: repeat-x;
   filter: progid:dximagetransform.microsoft.gradient(startColorstr=#ffffff,
endColorstr=#e6e6e6, GradientType=0);
   border-color: #e6e6e6 #e6e6e6 #e6e6e6;
   border-color: rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.25);
   border: 1px solid #e6e6e6;
   -webkit-border-radius: 4px;
   -moz-border-radius: 4px;
   border-radius: 4px;
   -webkit-box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px
rgba(0, 0, 0, 0.05);
   -moz-box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0,
0, 0, 0.05);
   box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0,
0, 0.05);
   cursor: pointer; *margin-left: .3em;
   }

.btn:hover, .btn:active, .btn.active, .btn.disabled, .btn[disabled] {
background-color: #e6e6e6; }

.btn-large {
   padding: 9px 14px;
   font-size: 15px;
   line-height: normal;
   -webkit-border-radius: 5px;
   -moz-border-radius: 5px;
   border-radius: 5px;
   }

.btn:hover {
   color: #333333;
   text-decoration: none;
   background-color: #e6e6e6;
   background-position: 0 -15px;
   -webkit-transition: background-position 0.1s linear;
   -moz-transition: background-position 0.1s linear;
   -ms-transition: background-position 0.1s linear;
   -o-transition: background-position 0.1s linear;
   transition: background-position 0.1s linear;
   }

.btn-primary, .btn-primary:hover {
   text-shadow: 0 -1px 0 rgba(0, 0, 0, 0.25);
   color: #ffffff;
   }

.btn-primary.active { color: rgba(255, 255, 255, 0.75); }

.btn-primary {
   background-color: #d70c0c;
   background-image: -moz-linear-gradient(top, #6eb6de, #4a77d4);
   background-image: -ms-linear-gradient(top, #6eb6de, #4a77d4);
   background-image: -webkit-gradient(linear, 0 0, 0 100%, from(#6eb6de),
to(#4a77d4));
   background-image: -webkit-linear-gradient(top, #6eb6de, #4a77d4);
   background-image: -o-linear-gradient(top, #6eb6de, #4a77d4);
   background-image: linear-gradient(top, #6eb6de, #4a77d4);
   background-repeat: repeat-x;
   filter: progid:dximagetransform.microsoft.gradient(startColorstr=#6eb6de,
endColorstr=#4a77d4, GradientType=0);
   border: 1px solid #3762bc;
   text-shadow: 1px 1px 1px rgba(0,0,0,0.4);
   box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0,
0, 0.5);
   }

.btn-primary:hover, .btn-primary:active, .btn-primary.active, .btn-
primary.disabled, .btn-primary[disabled] {
   filter: none;
   background-color: #d70c0c
   }

.btn-block { width: 100%; display:block; }

* { -webkit-box-sizing:border-box; -moz-box-sizing:border-box; -ms-box-
sizing:border-box; -o-box-sizing:border-box; box-sizing:border-box; }

html { width: 100%; height:100%; overflow:hidden; }
```

```
body {
   width: 100%;
   height:100%;
   font-family: 'Open Sans', sans-serif;
   color: #000000;
   font-size: 18px;
   text-align:center;
   letter-spacing:1.2px;


}
.header {
         top:0;
         margin:0px;
         left: 0px;
         right: 0px;
         position: fixed;
         background: #d44a4a;
         color: black;
         box-shadow: 0px 8px 4px grey;
         overflow: hidden;
         padding: 15px;
         font-size: 1.5vw;
         width: 100%;
         text-align: center;
      }
.login {
   position: absolute;
   top: 70%;
   left: 50%;
   margin: -25px 0 0 -150px;
   width:400px;
   height:400px;
}

.header div { color: #fff; text-shadow: 0 0 10px rgba(0,0,0,0.3); letter-
spacing:1px; text-align:center; float:left; padding-left:150px;}

ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  padding-right:150px;
  overflow: hidden;
 }


li {
  float: right;
}

li a {
  display: block;
  color: white;
  text-align: center;
  padding: 0px 15px;
  text-decoration: none;
```

```
}


input {
   width: 100%;
   margin-bottom: 10px;
   background: rgba(255,255,255,255);
   border: none;
   outline: none;
   padding: 10px;
   font-size: 13px;
   color: black;
   text-shadow: black;
   border: 1px solid rgba(0,0,0,0.3);
   border-radius: 4px;
   box-shadow: inset 0 -5px 45px rgba(100,100,100,0.2), 0 1px 1px
rgba(255,255,255,0.2);
   -webkit-transition: box-shadow .5s ease;
   -moz-transition: box-shadow .5s ease;
   -o-transition: box-shadow .5s ease;
   -ms-transition: box-shadow .5s ease;
   transition: box-shadow .5s ease;
}
input:focus { box-shadow: inset 0 -5px 45px rgba(100,100,100,0.4), 0 1px 1px
rgba(255,255,255,0.2); }


textarea {
   width: 100%;
   margin-bottom: 10px;
   background: rgba(255,255,255,255);
   border: none;
   outline: none;
   padding: 10px;
   font-size: 13px;
   color: black;
   text-shadow: black;
   border: 1px solid rgba(0,0,0,0.3);
   border-radius: 4px;
   box-shadow: inset 0 -5px 45px rgba(100,100,100,0.2), 0 1px 1px
rgba(255,255,255,0.2);
   -webkit-transition: box-shadow .5s ease;
   -moz-transition: box-shadow .5s ease;
   -o-transition: box-shadow .5s ease;
   -ms-transition: box-shadow .5s ease;
   transition: box-shadow .5s ease;
}
textarea:focus { box-shadow: inset 0 -5px 45px rgba(100,100,100,0.4), 0 1px
1px rgba(255,255,255,0.2); }

select {
   width: 100%;
   margin-bottom: 10px;
   background: rgba(255,255,255,255);
   border: none;
   outline: none;
   padding: 10px;
   font-size: 13px;
   color: #000000;
   text-shadow: 1px 1px 1px rgba(0,0,0,0.3);
   border: 1px solid rgba(0,0,0,0.3);
   border-radius: 4px;
   box-shadow: inset 0 -5px 45px rgba(100,100,100,0.2), 0 1px 1px
rgba(255,255,255,0.2);
   -webkit-transition: box-shadow .5s ease;
   -moz-transition: box-shadow .5s ease;
   -o-transition: box-shadow .5s ease;
   -ms-transition: box-shadow .5s ease;
   transition: box-shadow .5s ease;
}
```

app.py:

```
from distutils.log import debug
# from sendgridmail import sendmail
from flask import Flask, render_template, request, redirect, url_for, session
from flask_mail import Mail, Message
import re
import os
import ibm_db
from dotenv import load_dotenv

load_dotenv()


app = Flask(__name__)

app.secret_key = 'a'
print("Try to connect to Db2")
conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=2f3279a5-73d1-4859-88f0-
a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=
;UID=;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;PWD=",
'','')
print("Connected Successfully")

app.config['MAIL_SERVER']='smtp.gmail.com'
app.config['MAIL_PORT'] = 465
app.config['MAIL_USERNAME'] = 'example@gmail.com'
app.config['MAIL_PASSWORD'] = ‘*******'
app.config['MAIL_USE_TLS'] = False
app.config['MAIL_USE_SSL'] = True
mail = Mail(app)

@app.route('/')
@app.route('/login')
def login():

    return render_template('login.html')


@app.route('/loginpage',methods=['GET', 'POST'])
def loginpage():
    global userid
    msg = ''

    if request.method == 'POST' :
        username = request.form['username']
        password = request.form['password']
        sql = "SELECT * FROM donors WHERE username =? AND password=?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt,1,username)
        ibm_db.bind_param(stmt,2,password)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print (account)
        if account:
            session['loggedin'] = True
            session['id'] = account['USERNAME']
            userid=  account['USERNAME']
            session['username'] = account['USERNAME']
            msg = 'Logged in successfully !'
            index(account['EMAIL'],'Plasma donor App login','You are
successfully logged in!')
            return redirect(url_for('dash'))
        else:
            msg = 'Incorrect username / password !'
    return render_template('login.html', msg = msg)

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

@app.route('/register',methods=['GET', 'POST'])
def register():
    msg = ''
    if request.method == 'POST' :
        username = request.form['username']
        email = request.form['email']
        password = request.form['password']
        phone = request.form['phone']
        city = request.form['city']
        infect = request.form['infect']
        blood = request.form['blood']
        sql = "SELECT * FROM donors WHERE username =?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt,1,username)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print("ac",account)
        if account:
            msg = 'Account already exists !'
        elif not re.match(r'[^@]+@[^@]+\.[^@]+', email):
            msg = 'Invalid email address !'
        elif not re.match(r'[A-Za-z0-9]+', username):
            msg = 'name must contain only characters and numbers !'
        else:
            insert_sql = "INSERT INTO  donors VALUES (?, ?, ?, ?, ?, ?, ?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, username)
            ibm_db.bind_param(prep_stmt, 2, password)
            ibm_db.bind_param(prep_stmt, 3, email)
            ibm_db.bind_param(prep_stmt, 4, phone)
            ibm_db.bind_param(prep_stmt, 5, city)
            ibm_db.bind_param(prep_stmt, 6, infect)
            ibm_db.bind_param(prep_stmt, 7, blood)

            ibm_db.execute(prep_stmt)
            msg = 'You have successfully registered, !'
            index(email,'Plasma donor App Registration','You are successfully
Registered {}!'.format(username))

    elif request.method == 'POST':
        msg = 'Please fill out the form !'
    return render_template('register.html', msg = msg)

@app.route('/dashboard')
def dash():
    if session['loggedin'] == True:
        sql = "SELECT COUNT(*), (SELECT COUNT(*) FROM DONORS WHERE blood= 'O
Positive'), (SELECT COUNT(*) FROM DONORS WHERE blood='A Positive'), (SELECT
COUNT(*) FROM DONORS WHERE blood='B Positive'), (SELECT COUNT(*) FROM DONORS
WHERE blood='AB Positive'), (SELECT COUNT(*) FROM DONORS WHERE blood='O
Negative'), (SELECT COUNT(*) FROM DONORS WHERE blood='A Negative'), (SELECT
COUNT(*) FROM DONORS WHERE blood='B Negative'), (SELECT COUNT(*) FROM DONORS
WHERE blood='AB Negative') FROM donors"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print(account)
        return render_template('dashboard.html',b=account)
    else:
        msg = 'Please login!'
        return render_template('login.html', msg = msg)

@app.route('/requester')
def requester():
    if session['loggedin'] == True:
        return render_template('request.html')
    else:
        msg = 'Please login!'
        return render_template('login.html', msg = msg)

@app.route('/requested',methods=['POST'])
def requested():
    bloodgrp = request.form['bloodgrp']
    address = request.form['address']
    name=  request.form['name']
    email=  request.form['email']
    phone= request.form['phone']
    insert_sql = "INSERT INTO  requested VALUES (?, ?, ?, ?, ?)"
    prep_stmt = ibm_db.prepare(conn, insert_sql)
    ibm_db.bind_param(prep_stmt, 1, bloodgrp)
    ibm_db.bind_param(prep_stmt, 2, address)
    ibm_db.bind_param(prep_stmt, 3, name)
    ibm_db.bind_param(prep_stmt, 4, email)
    ibm_db.bind_param(prep_stmt, 5, phone)
    ibm_db.execute(prep_stmt)
    index(email,'Plasma donor App plasma request','Your request for plasma is
recieved.')
    return render_template('request.html', pred="Your request is sent to the
concerned people.")


def index(usermail,subject,content):

   msg = Message(subject, sender = 'example@gmail.com', recipients =
[usermail])
   msg.body = format(content)
   mail.send(msg)
   return "Sent"

@app.route('/logout')
def logout():
   session.pop('loggedin', None)
   session.pop('id', None)
   session.pop('username', None)
   return render_template('login.html')

if __name__ == '__main__':
   app.run(host='0.0.0.0',debug='TRUE')
```