

Sprint 3
PROJECT_DEVELOPMENT_PHASE

Date	12 November 2022
Team ID	PNT2022TMID49592
Project Name	Project - Personal expense tracker application

Code:

app.py(updated)

```
from flask import Flask, render_template, request, redirect, url_for, session
import ibm_db
import re
from flask import *
from flask import Flask
from random import *
from flask_mail import Mail, Message
app = Flask(__name__)
mail = Mail(app) # instantiate the mail class

# configuration of mail
app.config['MAIL_SERVER']='smtp.gmail.com'
app.config['MAIL_PORT'] = 465
app.config['MAIL_USERNAME'] = 'shrikumar13102001@gmail.com'
app.config['MAIL_PASSWORD'] = 'mlzuhzieegwenrhc'
app.config['MAIL_USE_TLS'] = False
app.config['MAIL_USE_SSL'] = True
mail = Mail(app)
otp = randint(000000,999999)
app.secret_key = 'a'
conn =
ibm_db.connect("DATABASE=bludb;HOSTNAME=b1bc1829-6f45-4cd4-bef4-10cf081900bf.c1o
gj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32304;SECURITY=SSL;SSLServerCerti
ficate=DigiCertGlobalRootCA.crt;UID=jpg92341;PWD=5HWT2ImD7POkuLnv","")
@app.route("/")
def home():
    return render_template('index.html')
@app.route('/login', methods=['GET', 'POST'])
def login():
    global userid
    msg = ""
    if request.method == 'POST' and 'email' in request.form and 'password1' in request.form:
        email = request.form['email']
```

```

password1 = request.form['password1']
stmt = ibm_db.prepare(conn,'SELECT * FROM regi WHERE email = ? AND password1 =
?')
ibm_db.bind_param(stmt,1,email)
ibm_db.bind_param(stmt,2,password1)
ibm_db.execute(stmt)
account = ibm_db.fetch_tuple(stmt)
if account:
    session['id'] = account[0]
    userid = account[0]
    session['email'] = account[1]
    msg = 'Logged in successfully !'
    return render_template('button.html', msg = msg)
else:
    msg = 'Incorrect username / password !'
    return render_template('login.html', msg = msg)
@app.route('/button')
def button():
    return render_template('button.html')

@app.route('/limit', methods=["POST", "GET"])
def limit():

    if request.method == "POST" and 'id' in session:
        keys = ['hos', 'ent', 'shop', 'groc', 'edu']
        values = []
        for k in keys:
            try:
                values.append(int(request.form[k]))
            except ValueError:
                return render_template('limit.html', status="Missing Expenses")

    print("Executing INSERT into LIMIT")
    uid = session['id']

    exist = ibm_db.prepare(conn, 'SELECT uid FROM limit WHERE uid = ?')
    ibm_db.bind_param(exist,1, session['id'])
    ibm_db.execute(exist)

    stmt = ""
    if ibm_db.fetch_tuple(exist) == False:
        print("Creating New")

```

```

        stmt = ibm_db.prepare(conn, 'INSERT INTO limit (hos, ent, shop, groc, edu, uid)
VALUES (?, ?, ?, ?, ?, ?)')
    else:
        print("Updating")
        stmt = ibm_db.prepare(conn, 'UPDATE limit SET \
        hos = ?, \
        ent = ?, \
        shop = ?, \
        groc = ?, \
        edu = ? \
        WHERE uid = ?')

    for i in range(1, 6):
        ibm_db.bind_param(stmt, i, values[i - 1])
    ibm_db.bind_param(stmt, 6, uid)
    ibm_db.execute(stmt)
    return render_template('limit.html', status="Success")
return render_template('limit.html')

@app.route('/stat')
def stat():
    if 'id' in session:
        stmt = ibm_db.prepare(conn, 'SELECT expense, expensedate FROM expense WHERE
uid = ?')
        ibm_db.bind_param(stmt, 1, session['id'])
        ibm_db.execute(stmt)
        tb = ibm_db.fetch_tuple(stmt)

        months = {}
        while tb != False:
            sliced = tb[1].strftime("%b")
            if sliced in months:
                months[sliced] += tb[0]
                continue
            months[sliced] = tb[0]
            tb = ibm_db.fetch_tuple(stmt)

        print(months)
        return render_template('stat.html', data=months)
    return 'Not Authed Please Login'

@app.route('/display')
def display():
    if 'id' in session :

```

```

        stmt = ibm_db.prepare(conn, 'SELECT amount, income, expense, expensedate, category
FROM expense WHERE uid = ?')
        ibm_db.bind_param(stmt,1,session['id'])
        ibm_db.execute(stmt)

```

```

        tb = ibm_db.fetch_assoc(stmt)
        data = []
        while tb != False:
            data.append(tb)
            tb = ibm_db.fetch_assoc(stmt)
        print(data)
        return render_template('display.html',data=data)

```

```

    return 'Not Authed'

```

```

@app.route('/wallet')

```

```

def wallet():

```

```

    return render_template('design.html')

```

```

@app.route('/income', methods=['GET', 'POST'])

```

```

def income():

```

```

    if 'id' in session and request.method == 'POST' and ('amount' in request.form) and ('income' in
request.form) and ('expense' in request.form) and ('expensedate' in request.form )and
('category' in request.form):

```

```

        msg = "

```

```

        print(request.form['category'])

```

```

        uid = session['id']

```

```

        amount = request.form['amount']

```

```

        income = request.form['income']

```

```

        expense = request.form['expense']

```

```

        expensedate = request.form['expensedate']

```

```

        category = request.form['category']

```

```

        stmt = ibm_db.prepare(conn, 'SELECT * FROM regi WHERE id = ?')

```

```

        ibm_db.bind_param(stmt,1,uid)

```

```

        ibm_db.execute(stmt)

```

```

        ibm_db.fetch_tuple(stmt)

```

```

        prep_stmt = ibm_db.prepare(conn, "INSERT INTO

```

```

expense(uid,amount,income,expense,expensedate,category) VALUES (?, ?, ?, ?, ?, ?)")

```

```

        ibm_db.bind_param(prepare_stmt, 1, uid)

```

```

        ibm_db.bind_param(prepare_stmt, 2, amount)

```

```

        ibm_db.bind_param(prepare_stmt, 3, income)

```

```

        ibm_db.bind_param(prepare_stmt, 4, expense)

```

```

        ibm_db.bind_param(prepare_stmt, 5, expensedate)

```

```

        ibm_db.bind_param(prepare_stmt, 6, category)

```

```

        ibm_db.execute(prepare_stmt)

```

```

        msg = 'You have successfully added !'
        return render_template('add.html', a = msg)
    return render_template('add.html')

```

```

@app.route('/register', methods=['GET', 'POST'])

```

```

def register():

```

```

    msg = "

```

```

    if request.method == 'POST':

```

```

        firstname = request.form['firstname']

```

```

        lastname = request.form['lastname']

```

```

        email = request.form['email']

```

```

        password1 = request.form['password1']

```

```

        cpassword = request.form['cpassword']

```

```

        birthdate = request.form['birthdate']

```

```

        phonenumber = request.form['phonenumber']

```

```

        job = request.form['job']

```

```

        income = request.form['income']

```

```

        sql = "SELECT * FROM regi 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:

```

```

            msg = 'Account already exists !'

```

```

        elif not re.match(r'^[a-zA-Z0-9]+@[a-zA-Z0-9]+\.[a-zA-Z]+', email):

```

```

            msg = 'Invalid email address !'

```

```

        elif not re.match(r'[A-Za-z0-9]+', firstname):

```

```

            msg = 'firstname must contain only characters and numbers !'

```

```

        elif not email or not firstname or not password1:

```

```

            msg = 'Please fill out the form !'

```

```

        else:

```

```

            insert_sql = "INSERT INTO

```

```

regi(firstname,lastname,email,password1,cpassword,birthdate,phonenumber,job,income)

```

```

VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)"

```

```

            stmt = ibm_db.prepare(conn,insert_sql)

```

```

            ibm_db.bind_param(stmt, 1, firstname)

```

```

            ibm_db.bind_param(stmt, 2, lastname)

```

```

            ibm_db.bind_param(stmt, 3, email)

```

```

            ibm_db.bind_param(stmt, 4, password1)

```

```

            ibm_db.bind_param(stmt, 5, cpassword)

```

```

            ibm_db.bind_param(stmt, 6, birthdate)

```

```

            ibm_db.bind_param(stmt, 7, phonenumber)

```

```

            ibm_db.bind_param(stmt, 8, job)

```

```

        ibm_db.bind_param(stmt, 9, income)
        ibm_db.execute(stmt)
        msg = 'You have successfully registered !'
        return render_template('email.html')
    elif request.method == 'POST':
        msg = 'Please fill out the form !'

    return render_template('register.html', msg = msg)
@app.route('/verify', methods=['GET', 'POST'])
def verify():
    if request.method == 'POST':
        email1 = request.form['email1']
        sql = "SELECT * FROM validate WHERE email1 = ? "
        stmt = ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,email1)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            msg = 'Account already exists !'
        else:
            insert_sql = "INSERT INTO validate VALUES (?)"
            stmt = ibm_db.prepare(conn,insert_sql)
            ibm_db.bind_param(stmt, 1, email1)
            ibm_db.execute(stmt)
            msg = Message('Hello',sender ='shrikumar13102001@gmail.com',recipients = [email1])
            msg.body = str(otp)
            mail.send(msg)
            return render_template('verify.html')
    else:
        return render_template('verify.html')
@app.route('/validate',methods=['GET', 'POST'])
def validate():
    user_otp = request.form['otp']
    if otp == int(user_otp):
        return render_template('index.html')
if __name__ == '__main__':
    app.run(debug = True)

```

USN-8:

REGISTRATION(WEB USER)

register.html

```
<html>
```

```

<head>
  <link rel="stylesheet" href="{{ url_for('static', filename='css/reg.css') }}">
  <link href="//maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
rel="stylesheet" id="bootstrap-css">
  <script src="//maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
  <script src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
  <!-- Include the above in your HEAD tag -->
  <link href="//netdna.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css"
rel="stylesheet" id="bootstrap-css">
  <script src="//netdna.bootstrapcdn.com/bootstrap/3.2.0/js/bootstrap.min.js"></script>
  <script src="//code.jquery.com/jquery-2.1.3.min.js"></script>
  
<body>
  <form action="{{ url_for('register') }}" method="post">
  <!-- Include the above in your HEAD tag -->
  <div class="container" >
  <form class="form-horizontal" role="form">
  <h2>Registration</h2>
  <div class="form-group">
  <label for="firstname" class="col-sm-3 control-label" >First Name*</label>
  <div class="col-sm-9">
  <input type="text" id="firstname" name="firstname" placeholder="First Name"
minlength="5" required class="form-control" autofocus>
  </div>
  </div>
  <div class="form-group">
  <label for="lastname" class="col-sm-3 control-label">Last Name</label>
  <div class="col-sm-9">
  <input type="text" id="lastname" name="lastname" placeholder="Last Name"
class="form-control" autofocus>
  </div>
  </div>
  <div class="form-group">
  <label for="email" class="col-sm-3 control-label">Email* </label>
  <div class="col-sm-9">
  <input type="email" id="email" name="email" placeholder="Email" class="form-control"
name="email" required>
  </div>
  </div>

```

```
<div class="form-group">
<label for="password" class="col-sm-3 control-label">Password* </label>
<div class="col-sm-9">
<input type="password" id="password" name="password1" minlength="4" required
placeholder="Password" class="form-control" >
</div>
</div>
<div class="form-group">
<label for="cpassword" class="col-sm-3 control-label">Confirm Password* </label>
<div class="col-sm-9">
<input type="password" id="cpassword" name="cpassword" placeholder="Password"
class="form-control" required>
</div>
</div>
<div class="form-group">
<label for="birthdate" class="col-sm-3 control-label">Date of Birth* </label>
<div class="col-sm-9">
<input type="date" id="birthdate" name="birthdate" class="form-control" required>
</div>
</div>
<div class="form-group">
<label for="phonenumber" class="col-sm-3 control-label">Phone number* </label>
<div class="col-sm-9">
<input type="text" id="phonenumber" name="phonenumber"
pattern="[0-9]{10}" required placeholder="Phone number" class="form-control">
<span class="help-block">Your phone number won't be disclosed anywhere </span>
</div>
</div>
<div class="form-group">
<label for="job" class="col-sm-3 control-label">Job* </label>
<div class="col-sm-9">
<input type="text" id="job" name="job" placeholder="Please write your job title"
class="form-control" required>
</div>
</div>
<div class="form-group">
<label for="income" class="col-sm-3 control-label">Income(in Rs)* </label>
<div class="col-sm-9">
<input type="text" id="income" name="income" placeholder="Please write your
income per month" class="form-control" min="0" required>
```



```

</div>
</div>
<div class="form-group">
<div class="col-sm-9 col-sm-offset-3">
<span class="help-block">*Required fields</span>
</div>
</div>
<button type="submit" class="btn btn-primary btn-block" action="{{ url_for('register') }}"
method="post">Register</button>
</div>
</form> <!-- /form -->
</div> <!-- ./container -->
</form>
</body>

</html>

```

OUTPUT:

Registration

First Name*

Last Name*

Email*

Password*

Confirm Password*

Date of Birth*

Phone number*

Your phone number won't be disclosed anywhere

Job*

Please write your job title

Income(in Rs)*

Please write your income per month

*Required fields

Register

Registration

First Name*

Last Name

Email*

Password*

Confirm Password*

Date of Birth*

Phone number*

Job*

Income(in Rs)*

*Required fields

IBM Db2 on Cloud

Load Data Load History **Tables** Views Indexes Aliases MQTs Sequences Application objects

JPG92341.REGISTER Back

Export to CSV

FIRSTNAME	LASTNAME	EMAIL	PASSWORD	CPASSWORD	BIRTHDATE	PHONENUMBER	JOB	INCOME
Sabana	S	f@gmail.com	sabana	sabana	2022-10-27	9876543210	student	12000
shrima	s	shrimathi@gmail.com	shri	shri	2022-10-13	1234567878	hin	10000
shrimathi	s	mathishri19@gmail.com	mathi	mathi	2022-10-13	1234567890	student	12000
shrtvvbkh	j	u@gmail.com	shri	shri	2022-10-12	9876543210	gfrdeswrt	12098

Resolving host...

USN-9

LOGIN PAGE(WEB USER)

login.html:

```
<!DOCTYPE html>
```

```
<html lang="en" >
<head>
<style>
body {
  background-image: linear-gradient(135deg, #eecda3 10%, #FAB2FF 100%);
  background-size: cover;
  background-repeat: no-repeat;
  background-attachment: fixed;
  font-family: "Open Sans", sans-serif;
  color: #333333;
}

.box-form {
  margin: 0 auto;
  width: 60%;
  height: 500px;
  background: #FFFFFF;
  border-radius: 10px;
  overflow: hidden;
  display: flex;
  flex: 1 1 100%;
  align-items: stretch;
  justify-content: space-between;
  box-shadow: 0 0 20px 6px #090b6f85;
}
@media (max-width: 750px) {
  .box-form {
    flex-flow: wrap;
    text-align: center;
    align-content: center;
    align-items: center;
  }
}
.box-form div {
  height: auto;
  top: 20%;
}
.box-form .left {
  color: #FFFFFF;
  background-size: cover;
```

```
background-repeat: no-repeat;
background: url("https://pngimg.com/uploads/piggy_bank/piggy_bank_PNG82.png");
overflow: hidden;
}
.box-form .left .overlay {
padding: 30px;
width: 500px;
height: 400px;
background: #c7c8d8ad;
overflow: hidden;
box-sizing: border-box;
}
.box-form .left .overlay h1 {
font-size: 5vmax;
line-height: 1;
font-weight: 900;
margin-top: 40px;
margin-bottom: 20px;
color: #4c4c4c;
}
.box-form .left .overlay p {
color: #4c4c4c;
}
.box-form .left .overlay span p {
margin-top: 30px;
font-weight: 900;
color: #904e95;
}
}
.box-form .left .overlay span a {
background: #3b5998;
color: #FFFFFF;
margin-top: 10px;
padding: 14px 50px;
border-radius: 100px;
display: inline-block;
box-shadow: 0 3px 6px 1px #042d4657;
}
}
.box-form .left .overlay span a:last-child {
background: #1dcaff;
```

```
margin-left: 30px;
}
.box-form .right {
padding: 40px;
overflow: hidden;
background-image: linear-gradient(135deg, #904e95 40%, #fa9075 100%);
}
@media (max-width: 980px) {
    .box-form .right {
        width: 300px;
        height: auto;
    }
}
.box-form .right h5 {
font-size: 3vmax;
line-height: 0;
}
.box-form .right p {
font-size: 14px;
color: #B0B3B9;
}
.box-form .right .inputs {
overflow: hidden;
}
.box-form .right input {
width: 100%;
padding: 8px;
margin-top: 20px;
font-size: 14px;
border: none;
outline: none;
border-bottom: 2px solid #B0B3B9;
}
```

```
.box-form .right button {
color: #fff;
font-size: 16px;
padding: 12px 35px;
border-radius: 50px;
```

```
display: inline-block;
border: 0;
outline: 0;
box-shadow: 0px 4px 20px 0px #ff7d5c;
background-image: linear-gradient(135deg, #ffb4a1 10%, #fa9075 100%);
}
```

```
label {
display: block;
position: relative;
margin-left: 30px;
}
```

```
label::before {
content: ' \f00c';
position: absolute;
font-family: FontAwesome;
background: transparent;
border: 3px solid #70F570;
border-radius: 4px;
color: transparent;
left: -30px;
transition: all 0.2s linear;
}
```

```
label:hover::before {
font-family: FontAwesome;
content: ' \f00c';
color: #fff;
cursor: pointer;
background: #70F570;
}
```

```
label:hover::before .text-checkbox {
background: #70F570;
}
```

```
label span.text-checkbox {
display: inline-block;
height: auto;
```

```
position: relative;
cursor: pointer;
transition: all 0.2s linear;
}
```

```
label input[type="checkbox"] {
  display: none;
}
```

```
</style>
```

```
<meta charset="UTF-8">
```

```
<title>Login Page</title>
```

```
<link href="https://fonts.googleapis.com/css?family=Open+Sans" rel="stylesheet">
```

```
<link
```

```
href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css"
```

```
rel="stylesheet"
```

```
integrity="sha384-wvfXpqpZZVQGK6TAh5PVIGfQNHSoD2xbE+QkPxCAFINEevoEH3  
Sl0sibVcOQVnN" crossorigin="anonymous"><link rel="stylesheet" href="/style.css">
```

```
</head>
```

```
<body>
```

```
<form action="{{ url_for('login') }}" method="post">
```

```
<!-- partial:index.partial.html -->
```

```
<div class="box-form">
```

```
  <div class="left">
```

```
    <div class="overlay">
```

```
    <h1>Welcome!!!</h1>
```

```
    <p>Your budget is a passport to a healthy financial <br>life.You must learn  
to control your money <br>or it will control you...</p>
```

```
  </div>
```

```
</div>
```

```
  <div class="right">
```

```
  <h5>Login</h5>
```

```
  <p>Don't have an account? <a href="{{ url_for('register') }}">Creat Your  
Account</a> it takes less than a minute</p>
```

```
  <div class="inputs">
```

```

        <input type="text" name="email" placeholder="email">
        <br>
        <input type="password" name="password1"
placeholder="password">
        </div>

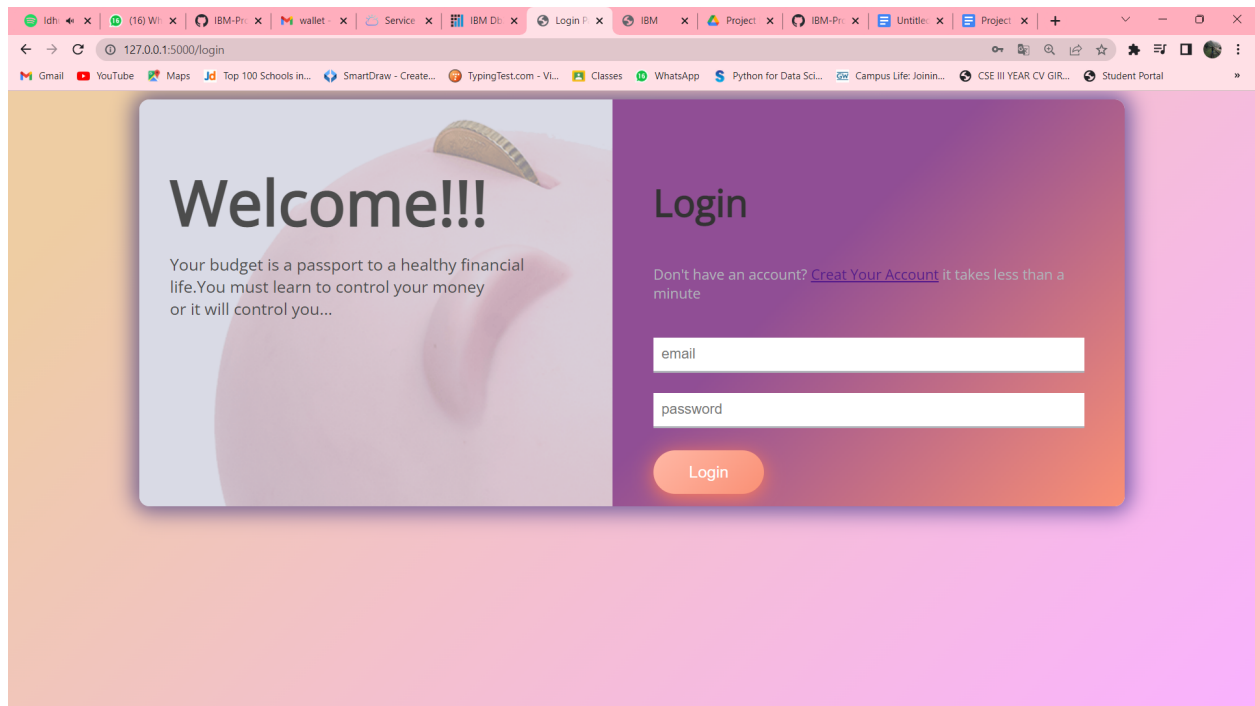
        <br>
        <button>Login</button>

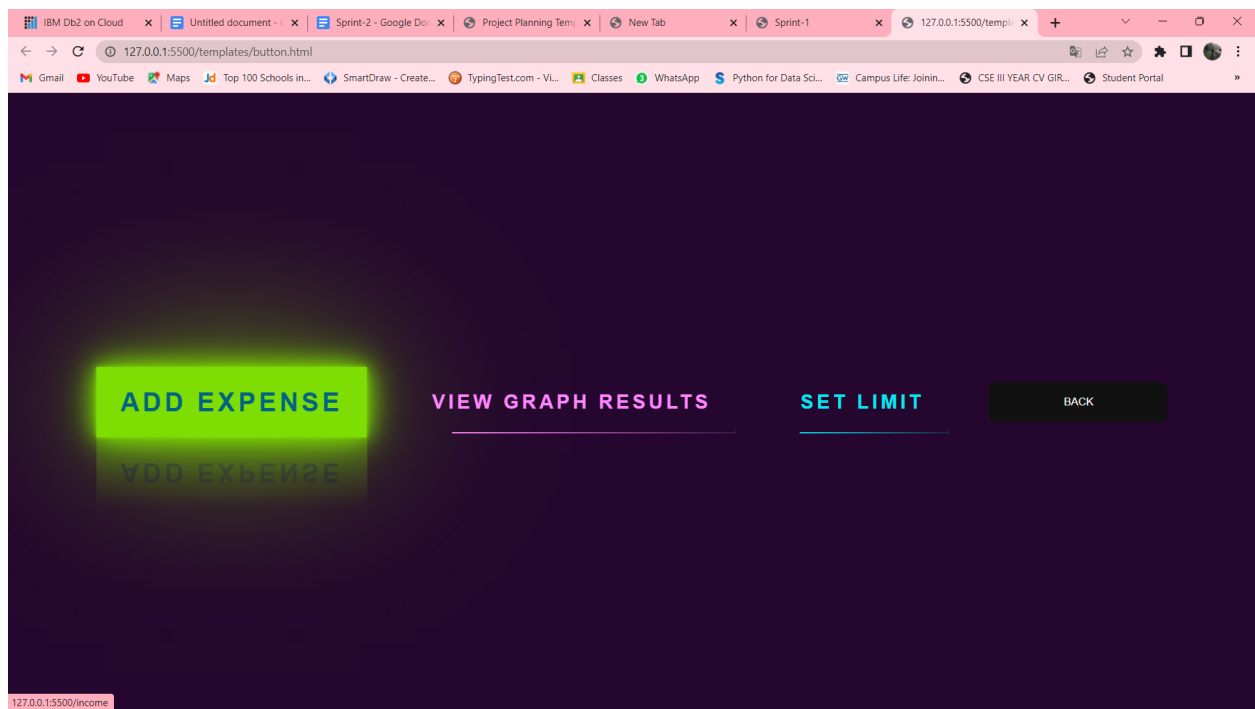
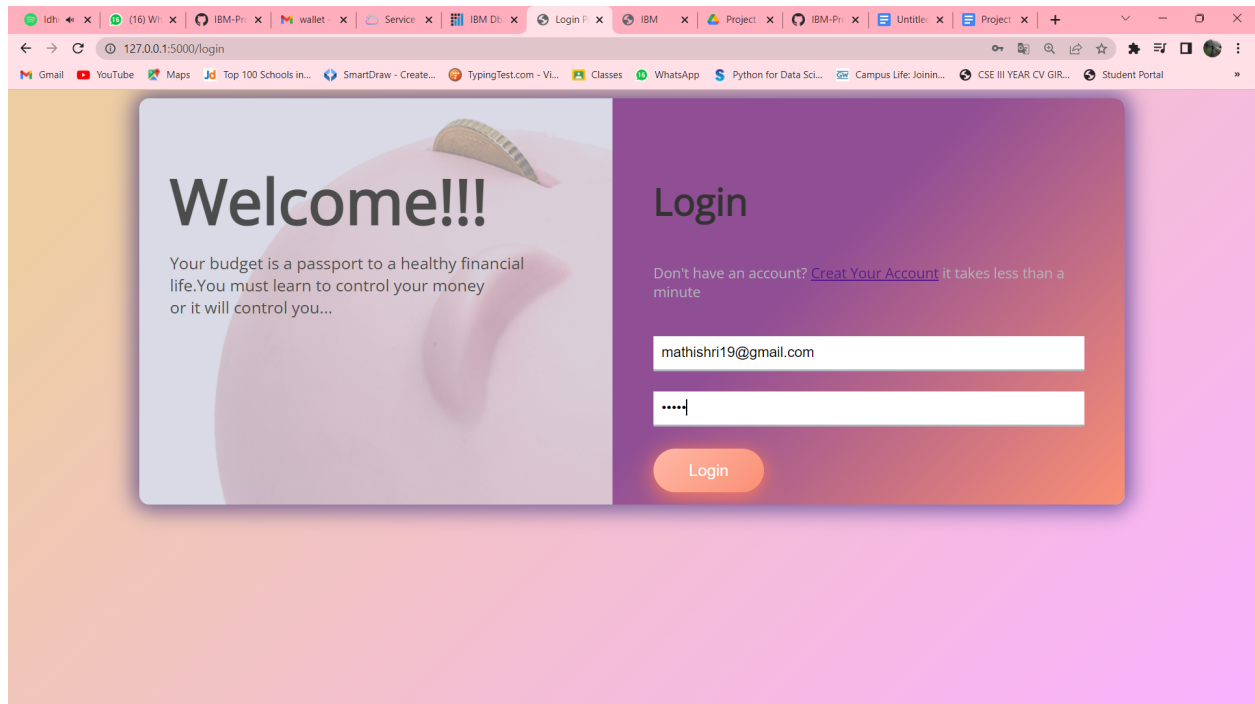
    </div>

</div>
<!-- partial -->
</form>
</body>
</html>

```

OUTPUT:





USN-10
GRAPHICAL VIEW:

stat.html

```
<body>
  <header>
    <button href="/button">Back</button>
  </header>
</body>
```

```
<canvas id="myChart"></canvas>
<script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/3.9.1/chart.min.js"></script>
<script>
const ctx = document.getElementById('myChart').getContext('2d');
```

```
const months = JSON.parse('{{ data | tojson | safe }}');
```

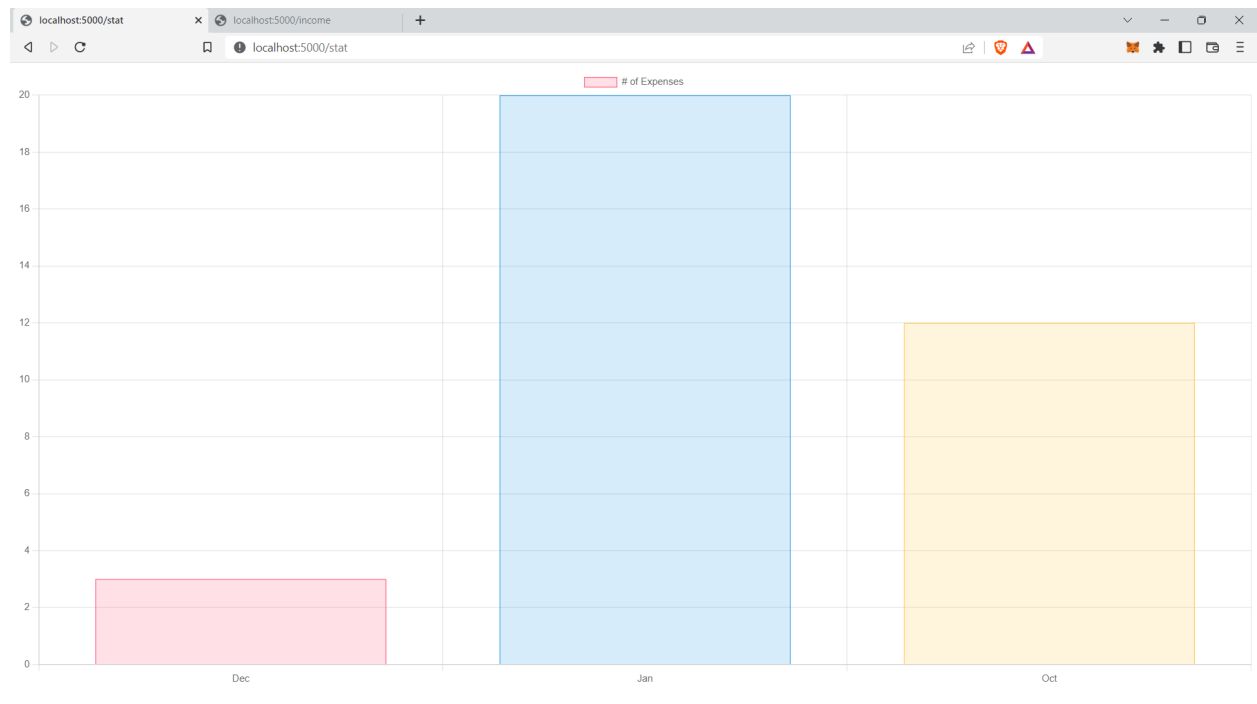
```
const monKeys = [];
const vals = [];
for(let val in months){
  monKeys.push(val);
  vals.push(months[val]);
}
// vals.reverse();
// monKeys.reverse();
console.log(months);
const myChart = new Chart(ctx, {
  type: 'bar',
  data: {
    labels: monKeys,
    datasets: [{
      label: '# of Expenses',
      data: vals,
      backgroundColor: [
        'rgba(255, 99, 132, 0.2)',
        'rgba(54, 162, 235, 0.2)',
        'rgba(255, 206, 86, 0.2)',
        'rgba(75, 192, 192, 0.2)',
        'rgba(153, 102, 255, 0.2)',
        'rgba(255, 159, 64, 0.2)'
      ],
      borderColor: [
        'rgba(255, 99, 132, 1)',
```

```

        'rgba(54, 162, 235, 1)',
        'rgba(255, 206, 86, 1)',
        'rgba(75, 192, 192, 1)',
        'rgba(153, 102, 255, 1)',
        'rgba(255, 159, 64, 1)'
    ],
    borderWidth: 1
  }]
},
options: {
  scales: {
    y: {
      beginAtZero: true
    }
  }
}
});
</script>

```

OUTPUT:



Users can able to view their monthly expenses in a graphical form.

HISTORY TABLE:

display.html

```
<head>
  <style>
    table {
      font-family: arial, sans-serif;
      border-collapse: collapse;
      width: 100%;}
    td, th {
      border: 1px solid #dddddd;
      text-align: left;
      padding: 8px;
    }
    tr:nth-child(even) {
      background-color: #ffacdc;
    }
  .fadeIn {
    opacity: 0;
    transform: translate3d(0, 20px, 0);
    transition: 0.25s all ease-in-out;
  }

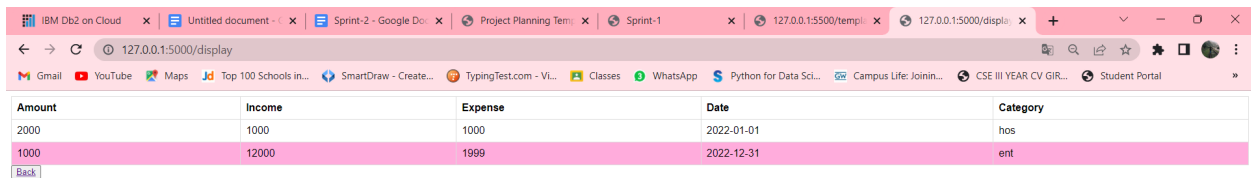
  .fadeIn:hover {
    opacity: 1;
    transform: translate3d(0, 0, 0);
  }
  </style>
</head>
<!-- <div class="fadeIn"> -->
  <table>
    <thead>
      <th>Amount</th>
      <th>Income</th>
      <th>Expense</th>
      <th>Date</th>
      <th>Category</th>
    </thead>
```

```

<tbody>
  {% for val in data %}
    <tr>
      <td>{{val['AMOUNT']}}</td>
      <td>{{val['INCOME']}}</td>
      <td>{{val['EXPENSE']}}</td>
      <td>{{val['EXPENSEDATE']}}</td>
      <td>{{val['CATEGORY']}}</td>
    </tr>
  {% endfor %}
</tbody>
</table>
<button align="center"><a href="/income">Back</a></button>
</div>

```

OUTPUT:



Amount	Income	Expense	Date	Category
2000	1000	1000	2022-01-01	hos
1000	12000	1999	2022-12-31	ent

Back

Users can able to view their history details in this web application

SETTING LIMIT PAGE

limit.html

```
<!DOCTYPE html>
```

```
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.c
ss">
<style>
* {box-sizing: border-box;}

body {
  margin: 0;
  font-family: Arial, Helvetica, sans-serif;
}
#navbar {
  overflow: hidden;
  background-image: linear-gradient(120deg, #d4fc79 0%, #96e6a1 100%);
  padding: 90px 10px;
  transition: 0.4s;
  position: fixed;
  width: 100%;
  top: 0;
  z-index: 99;
}

#navbar a {
  float: left;
  color: black;
  text-align: center;
  padding: 12px;
  text-decoration: none;
  font-size: 18px;
  line-height: 25px;
  border-radius: 4px;
}

#navbar #logo {
  font-size: 35px;
  font-weight: bold;
  transition: 0.4s;
}
```

```
#navbar a:hover {  
  background-color: #ddd;  
  color: black;  
}
```

```
#navbar a.active {  
  background-color: dodgerblue;  
  color: white;  
}
```

```
#navbar-right {  
  float: right;  
}
```

```
@media screen and (max-width: 580px) {  
  #navbar {  
    padding: 20px 10px !important;  
  }  
  #navbar a {  
    float: none;  
    display: block;  
    text-align: left;  
  }  
  #navbar-right {  
    float: none;  
  }  
}  
.content {  
  padding: 16px;  
}
```

```
.sticky {  
  position: fixed;  
  top: 0;  
  width: 100%;  
}
```

```
.sticky + .content {  
  padding-top: 60px;
```

```
}  
.input-container {  
  display: -ms-flexbox; /* IE10 */  
  display: flex;  
  width: 100%;  
  margin-bottom: 30px;  
}  
  
.icon {  
  padding: 10px;  
  background-image: linear-gradient(120deg, #d4fc79 0%, #96e6a1 100%);  
  color: white;  
  min-width: 70px;  
  text-align: center;  
}  
  
.input-field {  
  width: 100%;  
  padding: 10px;  
  outline: none;  
}  
  
.input-field:focus {  
  background-image: linear-gradient(120deg, #d4fc79 0%, #96e6a1 100%);  
}  
  
/* Set a style for the submit button */  
.btn {  
  background-image: linear-gradient(120deg, #d4fc79 0%, #96e6a1 100%);  
  padding: 15px 20px;  
  border: none;  
  cursor: pointer;  
  width: 100%;  
  opacity: 0.9;  
}  
.btn:hover {  
  opacity: 1;  
}  
  
</style>
```



```

</head>
<body>

<div id="navbar">
  <a href="#default" id="logo"><span
style='font-size:100px;'>#128246;</span>PERSONAL EXPENSE
TRACKER<br><br><i class="fa fa-line-chart" style="font-size:24px"></i>SET LIMIT</a>
  <div id="navbar-right">
    <a class="active" href="#index"><i class="fa fa-home"></i>Home</a>
    <a href="#contact"><i class="fa fa-envelope"></i>Notify</a>
  </div>
</div>

<div style="margin-top:210px;padding:15px 15px 2500px;font-size:30px">

</div>
<body>
<div class="content">

  {{status}}
  <form action="{{ url_for('limit') }}" method="post"
style="max-width:500px;margin:auto">
    <h2><i class="fa fa-line-chart icon"> SET LIMIT</i></h2>

    <label for="hos">Hospital</label> <br>
    <input type="number" id="hos" name="hos" placeholder="1000">
    <br>
    <label for="ent">Entertainment</label> <br>
    <input type="number" id="ent" name="ent" placeholder="1000">

    <br>
    <label for="shop">Shopping</label> <br>
    <input type="number" id="shop" name="shop" placeholder="1000">
    <br>

    <label for="edu">Education</label> <br>
    <input type="number" id="edu" name="edu" placeholder="1000">
    <br>

    <label for="groc">Grocery</label> <br>

```

```

<input type="number" id="groc" name="groc" placeholder="1000">

    <button type="submit" class="btn"><i class="fa fa-thumbs-up"
style="font-size:24px"> set limit</i></button>
</form></div>

</body>
</html>

```

OUTPUT:

Success

SET LIMIT

Hospital
1000

Entertainment
1000

Shopping
1000

Education
1000

Grocery
1000

set limit

IBM Db2 on Cloud

Load Data Load History **Tables** Views Indexes Aliases MQTs Sequences Application objects

JPG92341.LIMIT

Back

Export to CSV

ENT	SHOP	GROC	EDU	HOS	UID
1000	1000	1000	1000	1000	1

Users can able to set the limit for some categories to avoid financial problems.