PERSONAL EXPENSE TRACKER APPLICATION

FINAL REPORT

TEAM MEMBERS:

C.MANISHA (711519BCS028)

S.MOHAMMED AASHIQ (711529BCS034)

S.MONISH KANNA(711519BCS035)

M.S.RANJITH (711519BCS301)

INTRODUCTION

When it comes to tracking expenses, you can make your system as simple as collecting receipts and organizing them once a month. You might get a little more information from other expense tracking systems (listing them in a spreadsheet, using money management software or even choosing an online application), but all methods have one thing in common: you have to get in the habit of thinking about your expenses. It's very easy to misplace a receipt or forget about any cash you spent. You may even think that a cup of coffee or a trip to the vending machine isn't worth tracking — although those little expenses can add up amazingly fast. There are all sorts of opportunities to throw a kink into your plan to track expenses.

LITERATURE SURVEY

1. **Wallet**: The Wallet software is a personal cost tracker that offers a tonne of features, a stunning user interface, and excellent customer care**.**
    - **Features** : Wallet makes it easy to keep a track of where your money is going. You can link your bank accounts to the app, and it will use visuals and graphs to give you insights into your spending habits. You can also set saving goals, and the app will help you achieve it. One delightful feature is called 'Was It Worth It', which generates a score for your recent purchases. Both the Android and iOS versions are free to download, but some of the better features can only be accessed with a subscription (premium).
    - **Cons :** - Data security - It's not usable at every single store - It doesn't work with some credit card
2. **Walnut** :Walnut is a free personal expense tracker application that provides a comprehensive list of features to help you manage your money and expenses.
    - **Features** : The images make it easier for you to keep track of your daily spending. In order to deliver timely reminders for paying bills and credit card debts, it also accesses your SMS. Along with them, you also get an ATM finder and a Split wise-like feature for simply splitting costs with friends or family. Walnut Prime is a service that also provides straightforward personal loans that may be paid back through convenient EMIs.
    - **Cons:** - More prone to data leaks and breaches - Too many advertisements
3. **Mint** :It is a free, easy to use budgeting app that supports automatic and customizable categorization of downloaded transactions and other convenient expense tracking features.
    - **Features** : Users of Mint may create personalised budgets, keep tabs on their spending, set any necessary bill-paying reminders, check their credit score, and see how their assets are doing. As soon as a user logs

in, their financial information is automatically updated, and past transactions are automatically categorised into preexisting or newly established categories.

- **Cons:** - Does not support multiple currencies - No bill pay feature

4. **Money Manager,** expense tracker Money manager, expense tracker, budget, wallet: expense and income tracker, money, finances app will help you take your budget, money and finances under control and won't take muchtime.
    - **Features :** You won't need to check your bank account or rummage through your wallet to understand your financial situation. You may simply spend money while hoarding and saving up with the Budget: spending and income tracker, money, financial app. Explications, personalization, multicurrency, reminders, safety, and a user-friendly interface with illustrations.
    - **Cons:** - Error while switching device. - Ads that get stuck while buffering.
5. **AndroMoney :** AndroMoney is a personal finance tool for use on mobile phone. By using this tool, we can better manage our wealth.
    - **Features** : Multiple accounts and support account balance & account transfer - Cloud Storage (Dropbox , Google Docs) - SYNC with other devices - Any currency with downloadable rates - Number pad with calculation - Hierarchical categories with custom attributes - Simple/ Detail / Custom Budgets - Trend, Pie and Bar charts for Expense and Cash Flow - Password Protection - Overview your expense and income summary - Back up data to Excel/ Mac Number
    - **Cons:**
    - Error while syncing with database
    - Complicated UI

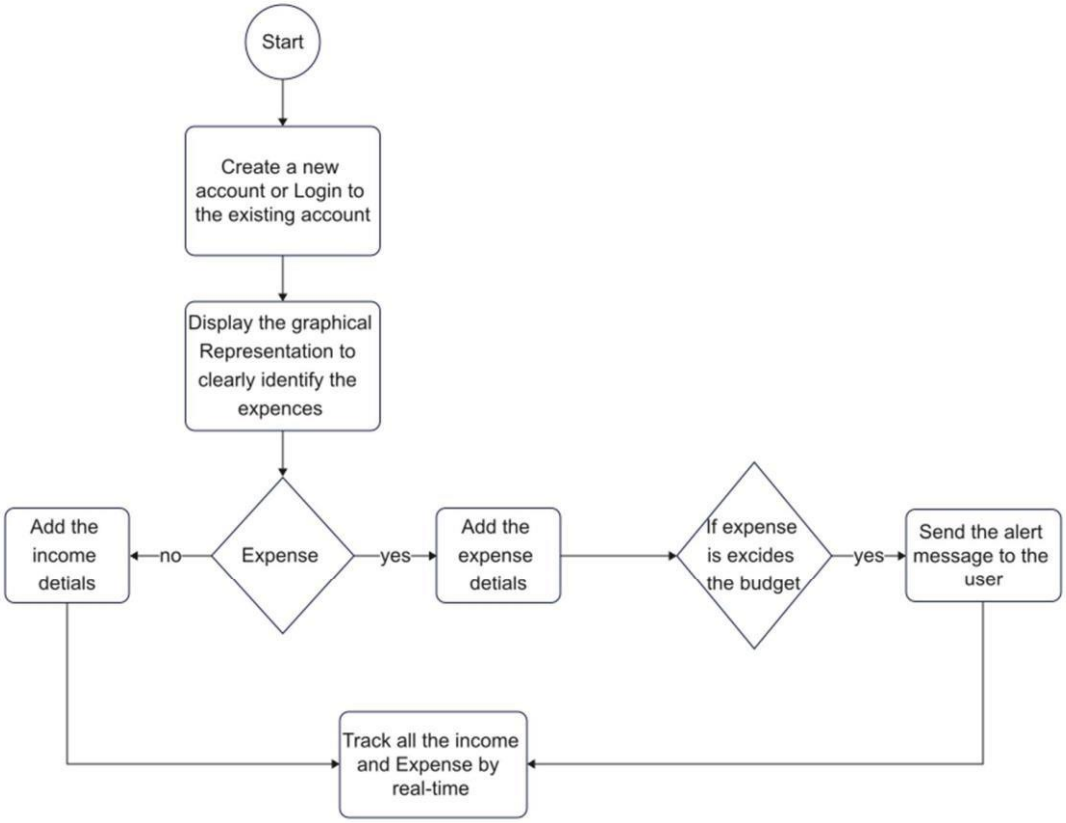
PROPOSED SYSTEM

S.No.	Parameter	Description
1	Problem Statement(Problem to besolved)	To track expense of user accusing to their budget. People can earn money in many ways but they can saveit only in one way i.e. by controlling the unwanted expenditure . Pen paper method

		and excel method is not efficient in maintaining expense and it is difficult to manage.
2	Idea / Solution Description	The goal of this project is to understandth e users requirement and to design themodule in order to create an efficient expense tracker. Making them aware oftheir expenses by exhibiting their expenditure through colorful insights. Throwing a mail as an alert warning iftheir expenditure exceeds their budgetproposed. Allowing them to compare their expenses on the daily, weekly, monthly and yea rly basis.
3	Novelty or Uniqueness	Creating special option for remindingusers about their loan repay or other commitments on savings for specific reasons. Providing beautiful insights like bar graph or pie chart for showingcomparisons and giving the best user interfaces.

4	Social Impact/ CustomerSatisfaction	By using Cloud computing provided by IBM for hosting the website it could able to make a small change which could lead to many tremendousfor an individual. A good change in anindividual leads to change in the society.
5	Business Model(RevenueBasis)	We could able to gain profit through this project by creating premium accounts for specialized features. By fixing the amount for premium with moderate rate the user could able to use the application with ease and the developer could able to gain some profit. The profit of the application is based on the best design and userexperience of the product.

SOLUTION ARCHITECTURE



MODULES

- Python
- Flask
- Dockers
- IBM Cloud

SOURCECODE

HOME

```
{% extends 'base.html' %}


{% block body %}
<style>

   H1 {       position:
relative;       right: -
790PX;       top: -
400PX;       color:
RED;


   }
```

```
p{ position: relative;
right: -800px; top: -
350px; font-
family:monospace;
}

span{
   position:
relative; right: -
800px; top: -
360px; }



  .ccc {        position:
relative;
top:80px;        left:-
100px;

}
</sty
le>
   <div id=aa class="container">
   <div class="ccc">
      <img src="../static/images/images (13).jpeg">
      <h1>LET START JOURNEY</h1>
      <P>MyBudget web application helps<br> you to maintain budget<br>
and analyse the expense</P>
      <img height="13%" width="13%" style="position: relative; top: -690PX; left: -
60PX;" src="../static/images/istockphoto-943300706-612x612.jpg">
   </div>
```

```
    <span  class="btn btn-outline-dark">Let's Begin</span>
</div>


{% endblock %}
```

SIGNIN

```
<!DOCTYPE html>
<html>
<head>
    <title>Animated Login Form</title>
    <link rel="stylesheet" type="text/css" href="..\static\css\login.css">
 <link href="https://fonts.googleapis.com/css?family=Poppins:600&display=swap" rel="stylesheet">
    <script src="https://kit.fontawesome.com/a81368914c.js"></script>
    <meta name="viewport" content="width=device-width, initial-scale=1">
</head>
<body >
    <img class="wave" src="..\static\images\wave.png">
```

```
<div class="container">

        <div class="img">
                <div id="png"><a href="D:/IBM/python-flask--personal-expensetracker-main/personal_expense_ttracker/templates/home.html" title="HOME"><img style="width:75px; height:75px ; " src="..\static\images\home-page.png"></a></div>
                <img src="..\static\images\bg.svg">
        </div>



        <div class="login-content">

                <form action='/login' method="POST">
                        <div class="msg">{{ msg }}</div>
                        <img src="..\static\images\avatar.svg">
                        <h2 class="title">Welcome</h2>
                <div class="input-div one">
                  <div class="i">
                                <i class="fas fa-user"></i>
                  </div>
                  <div class="div">
                                <h5>Username</h5>
                                 <input type="text" name="username" class="input" required>
                  </div>
                </div>
                <div class="input-div pass">
                  <div class="i">
                        <i class="fas fa-lock"></i>
                  </div>
                  <div class="div">
```

```
                        <h5>Password</h5>
                        <input type="password"  name="password" class="input"
required>
                    </div>
                </div>
                <a href="#">Forgot Password?</a>
                <input type="submit" class="btn" value="Login">
                        <span>OR</span>

                        <div><b>Login with</b></div>
                        <div>
                            <ul>
                                <li><a href="#"><i class="fab fa-facebook"
aria-hidden="true"></i></a></li>
                                <li><a href="#"><i class="fab fa-twitter"
ariahidden="true"></i></a></li>
                                <li><a href="#"><i  class="fab fa-google"
ariahidden="true"></i></a></li>
                                <li><a href="#"><i class="fab fa-linkedin"
ariahidden="true"></i></a></li>
                                <li><a href="#"><i class="fab fa-instagram"
aria-hidden="true"></i></a></li>
                            </ul>

                        </div>
                        <div class="app" ><b>Don't have an account?</b><a
id="app1" href="D:/IBM/python-flask--personal-expense-tracker-
main/personal_expense_ttracker/templates/signup.html">REGISTER.here</a></div>
            </form>

        </div>

    </div>
```

```
        <script type="text/javascript" src="..\static\js\login.js"></script>
</body>
</html>
```

SIGNUP

```
<html>
<head>
<meta charset="utf-8">
<title>Sign-up</title>
<link href="..\static\css\signup.css" rel="stylesheet">
<script src="https://kit.fontawesome.com/a81368914c.js"></script>
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm
" crossorigin="anonymous">
</head>
<body>
 <!--container---------------------->
<div class="container" >
 <!--sign-up-box-container--->
<div class="sign-up">

   <div id="png"><a href="/" title="HOME"><img style="width:55px; height:55px ; "
src="..\static\images\home-page.png"></a></div>
 <!--heading-->
 <form action="/register" method="post">
    <div class="msg">{{ msg }}</div>
 <h1 class="heading">Hello,Friend</h1>
 <!--name-box-->
 <div class="text">
 <img height="20px" src="..\static\images\user.png" />
 <input placeholder="Name" type="text" name="username"/>
```

```
</div>
<!--Email-box-->
<div class="text">
<img height="12px" src="..\static\images/email.png" />
<input placeholder=" Example@gmail.com" type="email" name="email"" />
</div>
<!--Password-box-->
<div class="text">
<img height="20px" src="..\static\images\password.png" />
<input placeholder=" Password" type="password" name="password"/>
</div>
<div class="or"><b>OR</b></div>
<div class="s1"><p><b>Sign-up with</b></p></div>

<div>
  <ul>
    <li><a href="#"><i class="fab fa-facebook" aria-hidden="true"></i></a></li>
    <li><a href="#"><i class="fab fa-twitter" aria-hidden="true"></i></a></li>
    <li><a href="#"><i <i class="fab fa-google"  aria-hidden="true"></i></a></li>
    <li><a href="#"><i class="fab fa-linkedin" aria-hidden="true"></i></a></li>
    <li><a href="#"><i class="fab fa-instagram" aria-hidden="true"></i></a></li>
</ul>

</div>
<!--trems-->

<div class="trems">
  <input class="check" type="checkbox" required/>
<p class="conditions">I read and agree to <a href="#">Trems &amp;amp;
Conditions</a></p>
```

```
 </div>
<!--button-->
<div class="toop">
<button type="submit" class="btn btn-primary" >CREATE ACCOUNT</button> </div>
</form>
<!--sign-in-->
<div class="t"><p class="conditions" id="p3">Already have an account <a
href="D:/IBM/python-flask--personal-expense-tracker-
main/personal_expense_ttracker/templates/login.html">Sign in</a></p> </div></div>
</div>
<!--text-container-->
<div class="text-container">

 <h1 style="color: #2d2c2c;font-family:cursive;">Glad to see you</h1>

 <div class="diag"><img class="fig1" width="100%" height="105%"
src="..\static\images\Inkeddia_LI.jpg"</div>
<div class="para"> <b>Welcome</b>,Please Fill in the blanks for sign up</div>

 </div>
 </div>
</body>
</html>
```

APP.PY

```
# -*- coding: utf-8 -*-
"""
Spyder Editor

This is a temporary script file.
"""

from flask import Flask, render_template, request, redirect,
session  from flask_mysqldb import MySQL import
MySQLdb.cursors import re


app = Flask(__name__)


app.secret_key = 'a'

app.config['MYSQL_HOST'] = 'remotemysql.com'
app.config['MYSQL_USER'] = 'D2DxDUPBii'
app.config['MYSQL_PASSWORD'] = 'r8XBO4GsMz'
app.config['MYSQL_DB'] = 'D2DxDUPBii'

mysql = MySQL(app)


#HOME--PAGE
```

```
@app.route("/home")
def home():
   return render_template("homepage.html")

@app.route("/")
def add():
   return render_template("home.html") #SIGN--UP--OR--REGISTER


@app.route("/signup")
def signup():
   return render_template("signup.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']


        cursor = mysql.connection.cursor()
        cursor.execute('SELECT * FROM register WHERE username = % s', (username,
))         account = cursor.fetchone()         print(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:
        cursor.execute('INSERT INTO register VALUES (NULL, % s, % s, % s)',
(username, email,password))
        mysql.connection.commit()
        msg = 'You have successfully registered !'
return render_template('signup.html', msg = msg)



 #LOGIN--PAGE


@app.route("/signin")
def signin():
   return render_template("login.html")


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



   if request.method == 'POST' :
      username = request.form['username']
password = request.form['password']
cursor = mysql.connection.cursor()
      cursor.execute('SELECT * FROM register WHERE username = % s AND password
= % s', (username, password ),)
account = cursor.fetchone()
print (account)


      if account:
```

```
        session['loggedin'] = True
session['id'] = account[0]          userid=
account[0]          session['username'] =
account[1]

        return redirect('/home')
else:
        msg = 'Incorrect username / password !'
return render_template('login.html', msg = msg)
```

#ADDING----DATA

```
@app.route("/add")
def adding():
   return render_template('add.html')


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

```
    date = request.form['date']     expensename
= request.form['expensename']     amount =
request.form['amount']

    paymode = request.form['paymode']
category = request.form['category']


    cursor = mysql.connection.cursor()
    cursor.execute('INSERT INTO expenses VALUES (NULL,  % s, % s, % s, % s, % s,
% s)', (session['id'] ,date, expensename, amount, paymode, category))
mysql.connection.commit()
    print(date + " " + expensename + " " + amount + " " + paymode + " " + category)


    return redirect("/display")




#DISPLAY---graph


@app.route("/display")
def display():
    print(session["username"],session['id'])


    cursor = mysql.connection.cursor()
    cursor.execute('SELECT * FROM expenses WHERE userid = % s AND date ORDER
BY `expenses`.`date` DESC',(str(session['id'])))
expense = cursor.fetchall()



    return render_template('display.html' ,expense = expense)
```

```
#delete---the--data

@app.route('/delete/<string:id>', methods = ['POST', 'GET' ])
def delete(id):
    cursor = mysql.connection.cursor()
    cursor.execute('DELETE FROM expenses WHERE  id =
{0}'.format(id))      mysql.connection.commit()      print('deleted
successfully')         return redirect("/display")


#UPDATE---DATA

@app.route('/edit/<id>', methods = ['POST', 'GET' ])
def edit(id):
   cursor = mysql.connection.cursor()
   cursor.execute('SELECT * FROM expenses WHERE  id = %s', (id,))
row = cursor.fetchall()

   print(row[0])
   return render_template('edit.html', expenses = row[0])




@app.route('/update/<id>', methods =
['POST']) def update(id):   if request.method ==
'POST' :
```

```
    date = request.form['date']
    expensename = request.form['expensename']
    amount = request.form['amount']
    paymode = request.form['paymode']
    category = request.form['category']

    cursor = mysql.connection.cursor()

    cursor.execute("UPDATE `expenses` SET `date` = % s , `expensename` = % s , `amount` = % s, `paymode` = % s, `category` = % s WHERE `expenses`.`id` = % s ",(date, expensename, amount, str(paymode), str(category),id))
mysql.connection.commit()          print('successfully updated')
return redirect("/display")




 #limit
@app.route("/limit" )
def limit():
     return redirect('/limitn')

@app.route("/limitnum" , methods = ['POST'
]) def limitnum():      if request.method ==
"POST":
```

```
        number= request.form['number']
cursor = mysql.connection.cursor()
cursor.execute('INSERT INTO limits
VALUES (NULL, % s, % s) ',(session['id'],
number))
        mysql.connection.commit()
return redirect('/limitn')



@app.route("/limitn")
def limitn():
   cursor = mysql.connection.cursor()
   cursor.execute('SELECT limitss FROM `limits` ORDER BY `limits`.`id`
DESC LIMIT 1')     x= cursor.fetchone()     s = x[0]



   return render_template("limit.html" , y= s)

#REPORT


@app.route("/today")
def today():
     cursor = mysql.connection.cursor()
     cursor.execute('SELECT TIME(date)   , amount FROM expenses  WHERE userid =
%s AND DATE(date) = DATE(NOW())
',(str(session['id'])))       texpense = cursor.fetchall()
print(texpense)


     cursor = mysql.connection.cursor()
```

```
    cursor.execute('SELECT * FROM expenses WHERE userid = % s AND DATE(date)
= DATE(NOW()) AND date ORDER BY `expenses`.`date` DESC',(str(session['id'])))
expense = cursor.fetchall()

    total=0 t_food=0
    t_entertainment=
    0 t_business=0
    t_rent=0
    t_EMI=0
t_other=0


    for x in expense:
total += x[4]          if
x[6] == "food":
t_food += x[4]

        elif x[6] == "entertainment":
t_entertainment  += x[4]

        elif x[6] == "business":
t_business  += x[4]          elif
x[6] == "rent":             t_rent
+= x[4]

        elif x[6] == "EMI":
t_EMI  += x[4]

        elif x[6] == "other":
t_other  += x[4]
```

```
    print(total)

    print(t_food)
    print(t_entertainment)
    print(t_business)
    print(t_rent)
    print(t_EMI)
    print(t_other)



    return render_template("today.html", texpense = texpense, expense = expense,  total = total ,
                        t_food = t_food,t_entertainment =  t_entertainment,
t_business = t_business,  t_rent =  t_rent,                        t_EMI =
t_EMI,  t_other =  t_other )



@app.route("/month")
def month():
    cursor = mysql.connection.cursor()
    cursor.execute('SELECT DATE(date), SUM(amount) FROM expenses WHERE userid= %s AND MONTH(DATE(date))= MONTH(now()) GROUP BY DATE(date) ORDER BY DATE(date) ',(str(session['id'])))
texpense = cursor.fetchall()
print(texpense)


    cursor = mysql.connection.cursor()
    cursor.execute('SELECT * FROM expenses WHERE userid = % s AND MONTH(DATE(date))= MONTH(now()) AND date ORDER BY `expenses`.`date` DESC',(str(session['id'])))
expense = cursor.fetchall()
```

```
    total=0
    t_food=0
t_entertainment=0
t_business=0
t_rent=0 t_EMI=0
t_other=0


    for x in expense:
total += x[4]          if
x[6] == "food":
t_food += x[4]

        elif x[6] == "entertainment":
t_entertainment  += x[4]

        elif x[6] == "business":
t_business  += x[4]          elif
x[6] == "rent":             t_rent
+= x[4]

        elif x[6] == "EMI":
t_EMI  += x[4]

        elif x[6] == "other":
t_other  += x[4]

    print(total)
```

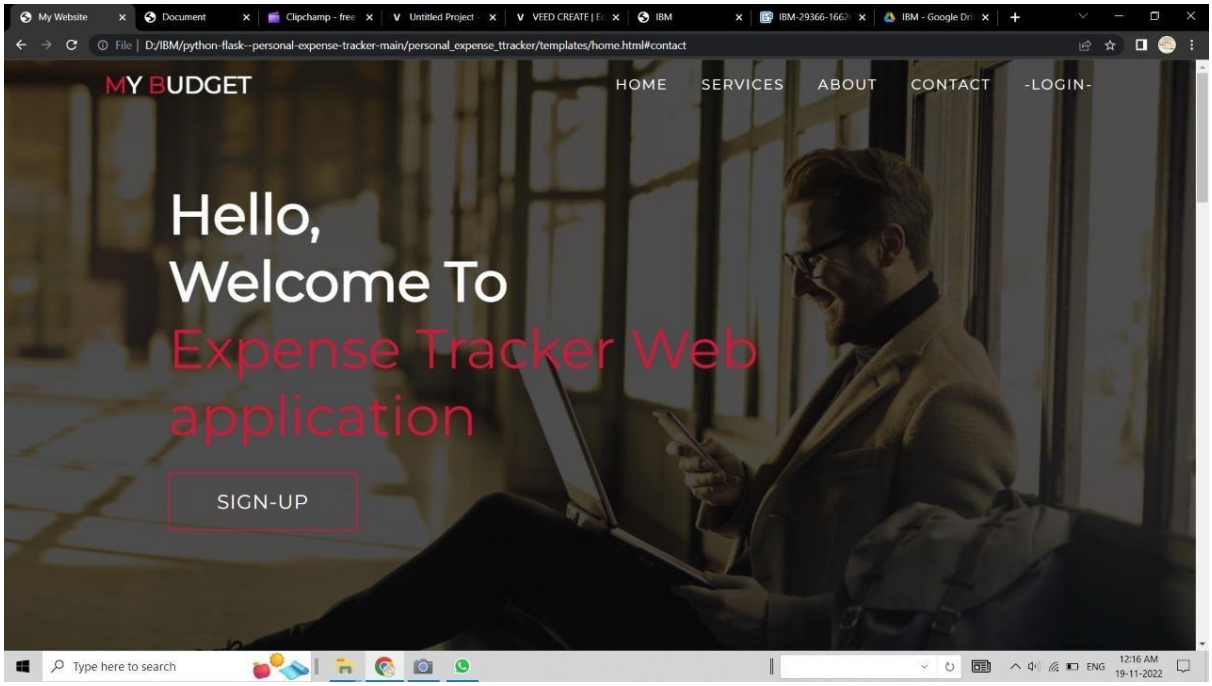
```
    print(t_food) print(t_entertainment) print(t_business) print(t_rent) print(t_EMI)
print(t_other)




    return render_template("today.html", texpense = texpense, expense = expense,  total =
total ,
                        t_food = t_food,t_entertainment =  t_entertainment,
t_business = t_business,  t_rent =  t_rent,                        t_EMI =
t_EMI,  t_other =  t_other )


@app.route("/year")
def year():
    cursor = mysql.connection.cursor()
    cursor.execute('SELECT MONTH(date), SUM(amount) FROM expenses WHERE
userid= %s AND YEAR(DATE(date))= YEAR(now()) GROUP BY MONTH(date)
ORDER BY MONTH(date)
',(str(session['id'])))      texpense =
cursor.fetchall()      print(texpense)


    cursor = mysql.connection.cursor()
    cursor.execute('SELECT * FROM expenses WHERE userid = % s AND
YEAR(DATE(date))= YEAR(now()) AND date ORDER BY `expenses`.`date`
DESC',(str(session['id'])))
expense = cursor.fetchall()


    total=0
t_food=0
t_entertainment=0
t_business=0
t_rent=0 t_EMI=0
t_other=0
```

```
    for x in expense:
total += x[4]          if
x[6] == "food":
t_food += x[4]

        elif x[6] == "entertainment":
t_entertainment  += x[4]

        elif x[6] == "business":
t_business  += x[4]          elif
x[6] == "rent":             t_rent
+= x[4]

        elif x[6] == "EMI":
t_EMI  += x[4]

        elif x[6] == "other":
t_other  += x[4]

    print(total)

    print(t_food)
print(t_entertainment)
print(t_business) print(t_rent)
print(t_EMI) print(t_other)
```

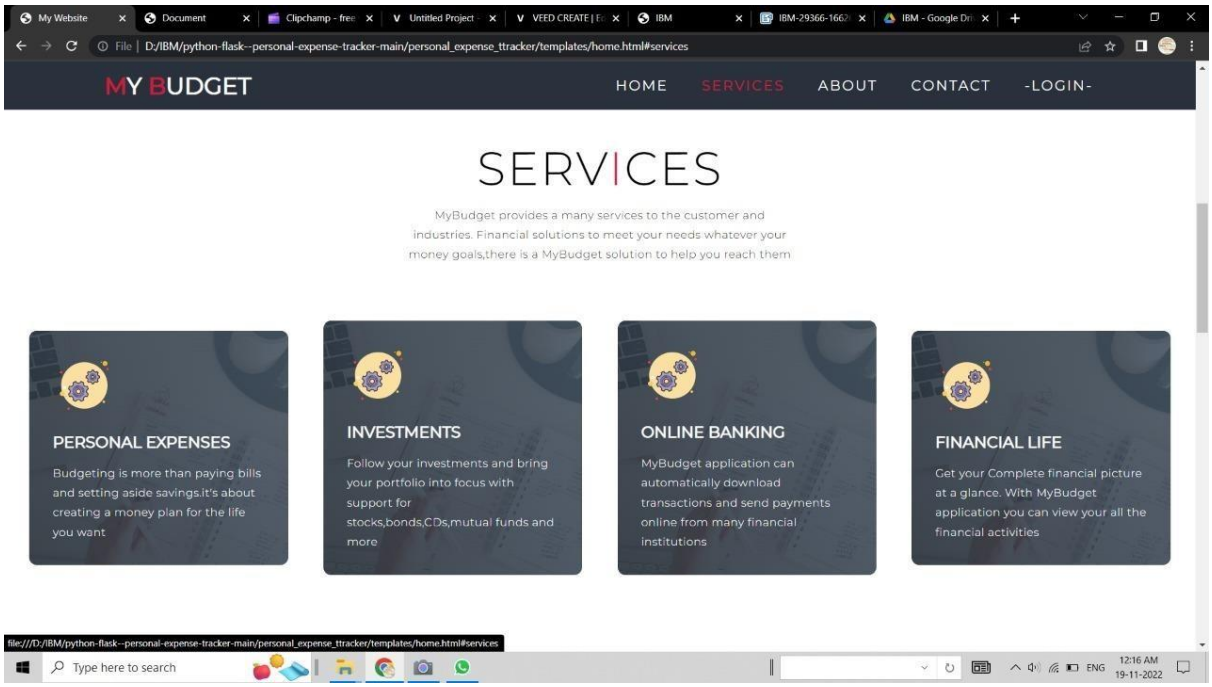
```
    return render_template("today.html", texpense = texpense, expense = expense,  total =
total ,
                        t_food = t_food,t_entertainment =  t_entertainment,
t_business = t_business,  t_rent =  t_rent,                        t_EMI =
t_EMI,  t_other =  t_other )


#log-out


@app.route('/logout')


def logout():
  session.pop('loggedin', None)
session.pop('id', None)
session.pop('username', None)
return render_template('home.html')




if __name__ == "__main__":
   app.run(debug=True)
```

SCREENRECORDS

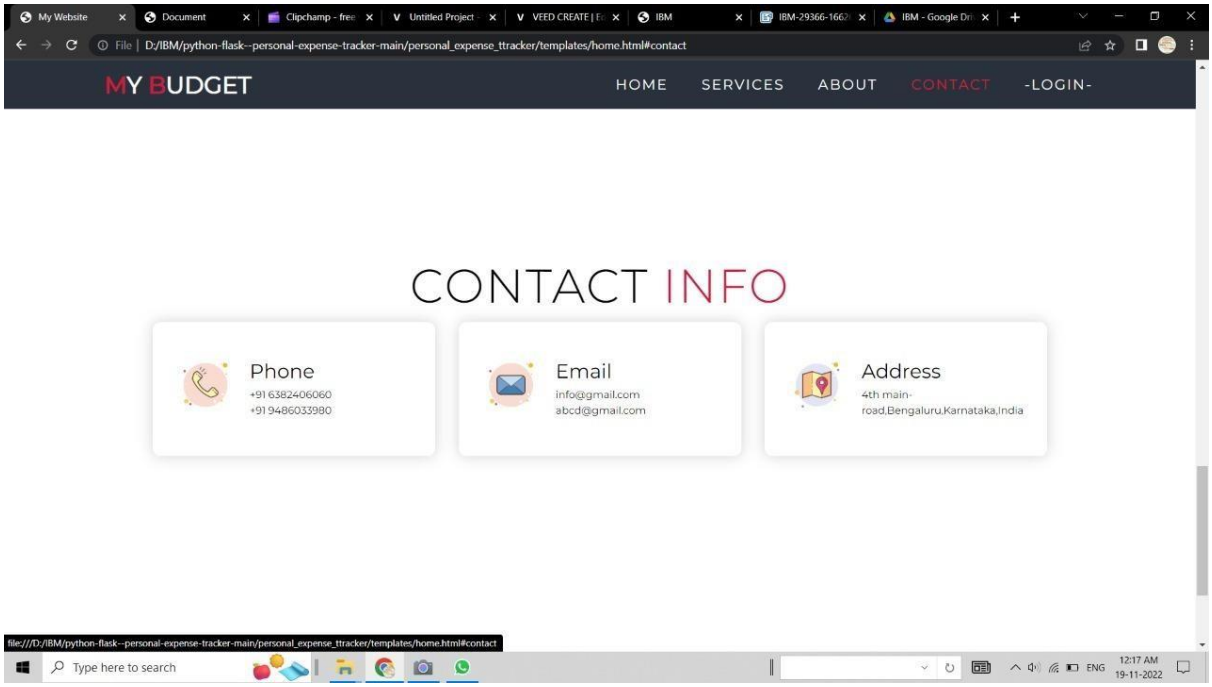
HOME


MY BUDGET
HOME
SERVICES
ABOUT
CONTACT
-LOGIN-
Hello,
Welcome To
Expense Tracker Web application
SIGN-UP

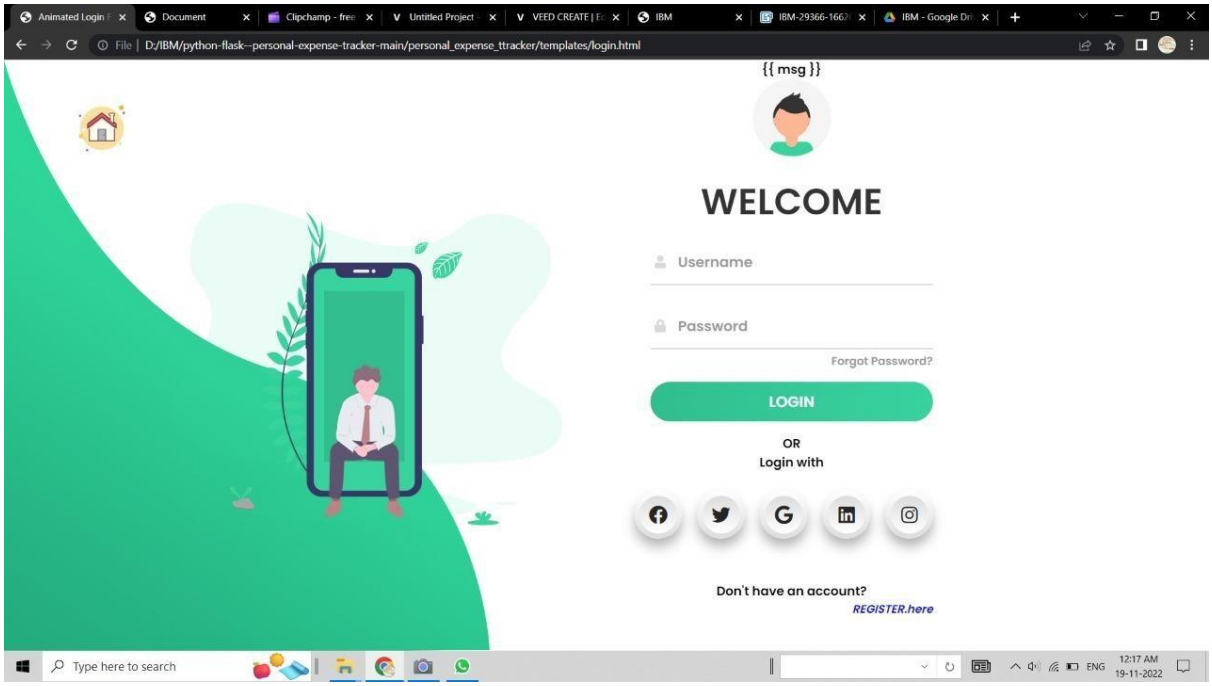
SERVICES


MY BUDGET
HOME
SERVICES
ABOUT
CONTACT
-LOGIN-
SERVICES
MyBudget provides a many services to the customer and industries. Financial solutions to meet your needs whatever your money goals,there is a MyBudget solution to help you reach them
PERSONAL EXPENSES
Budgeting is more than paying bills and setting aside savings.it's about creating a money plan for the life you want
INVESTMENTS
Follow your investments and bring your portfolio into focus with support for stocks,bonds,C.Ds,mutual funds and more
ONLINE BANKING
MyBudget application can automatically download transactions and send payments online from many financial institutions
FINANCIAL LIFE
Get your Complete financial picture at a glance. With MyBudget application you can view your all the financial activities

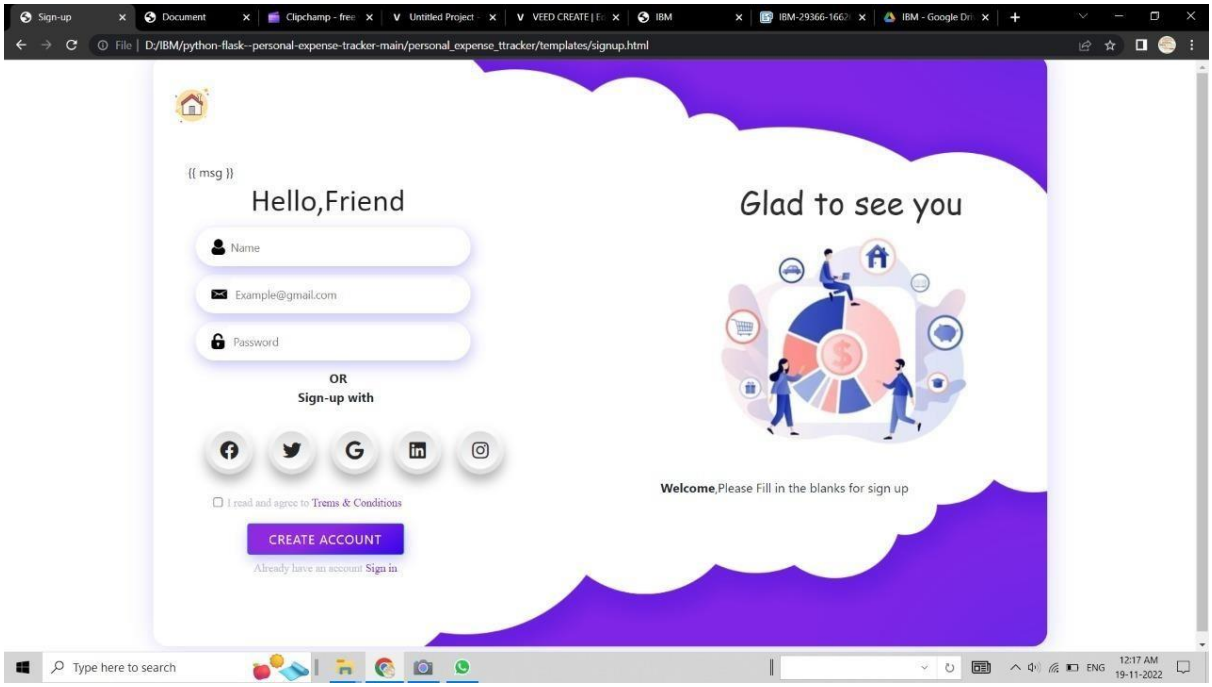
ABOUTUS



SIGNIN



SIGNUP



Future Enhancement

We commit to include the accompanying highlights into our cost chief application,demonstrating patterns, afterward we are going to include patterns. This is often the graphical portrayal of the final client costs per certain period, that might be month to month, quarterly, half-yearly and yearly too. Evaluating costs the opposite part interesting piece of our cost supervisor application is to measure costs that client may spend in present or one month from now. This can assist the client with adjusting his costs appropriately so he/she can abstain from bobbing up short on cash, gathering client information The assortment of knowledge of costs and pay of client from databases is easy undertaking since each record of client action is put away straightforwardly in database server maintained by us. Yet, to create our application safe and to stay up secrecy of client information we attempt vital strides to counter this issue. We are going to request client authorizations to utilize client information for examination reason and can guarantee classification of every datums identified with information, the knowledge tests are gathered from client without uncovering any of client personality Applying information mining strategies, the knowledge gathered from client is used to develop certain client examples, bunching and relationship of the prices.

CONCLUSIONS

After making this application we assure that this application will help its users to manage the cost of their daily expenditure. It will guide them and make them aware about their daily expenses. It will prove to be helpful for the people who are frustrated with their daily budget management, irritated because of the amount of expenses and wish to manage money and to preserve the record of their daily cost which may be useful to change their way of spending money. In short, this application will help its users to overcome the wastage of money