PERSONAL EXPENSE TRACKER APPLICATION

IBM NAALAIYA THIRAN

(TEAM ID:PNT2022TMID26896)

A PROJECT REPORT

Submitted by

RAMADEVI R(310519205032)

TRISHA M(310519205047)

HARIHARAN T(310519205013)

KEERTHANA J(310519205019)

JACKHARISH E(310519205016)

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOGY
DHANALAKSHMI SRINIVASAN COLLEGE OF ENGINEERING AND TECHNOLOGY
ECR ,MAMALLAPURAM CHENNAI-603 104

NOVEMBER- 2022

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"PERSONAL EXPENSE TRACKER APPLICATION"** is the bonafide work of Ramadevi.R**(310519205032), Trisha.M(310519205047),Hariharan.T(310519205013),Keerthana.J(310519205019),Jackharish.E(310519205016)**who carried out the **IBM NAALAIYA THIRAN** project work under our supervision.

Industry Mentor (Khushboo ,IBM)

Faculty Mentor (Mr.R.Idayathulla/AP)

Faculty Evaluator (Dr.Pon.Arivanantham/Hod)

Head of the Department/IT

Dr.Pon.Arivanantham

ACKNOWLEDGEMENT

First and foremost we thank the almighty for helping us in all situations for bringing out this project successfully.

We express our sincere heartfelt gratitude to MR. **SRINIVASAN**, **CHAIRMAN, Mr** P.MANI **Managing** Director**,** Dr.R.VIJAYARAJ M.E,Ph D Chief Executive Officer ,Dhanalakshmi Srinivasan College of Engineering and Technology , Chennai.

We would like to express our thanks to our principal **Dr.R.SARAVANAN, M.E, Ph D,** For his kind consent, inspiration and constant encouragement towards this project work.

We profusely thank **Dr.Pon.Arivanantham M.E,PhD** Head of the department, Information technology for the help and support, without which our project would have been sculpted successfully.

We express our heart full thanks to our **Industry Mentor, IBM** Faculty Mentor **Mrs.KHUSBOO** Faculty Evaluator **Dr.Pon.Arivanantham/Hod ,** For invaluable support, guidance, utmost patience, inspirational coordination and constant encouragements in completing this project successfully.

Also we would like to thank all the faculty members and non teaching staff of the Information Technology department for the kind advice and encouragement.

TABLE OF CONTENT

CHAPTER NO. **TITLE**

CHAPTER 1

INTRODUCTION

1.1 Project Overview

Mobile applications are top in user convenience and have overtaken the web applications in terms of popularity and usability. There are various mobile applications that provide solutions to manage personal and group expenses but not many of them provide a comprehensive view of both cases. In this paper, we develop a mobile application developed for the android platform that keeps record of user personal expenses, his/her contribution in group expenditures, top investment options, view of the current stock market, read authenticated financial news and grab the best ongoing offers in the market in popular categories. The proposed application would eliminate messy sticky notes, spreadsheets confusion and data handling inconsistency problems while offering the best overview of your expenses. With our application we can manage their expenses and decide on their budget more effectively.

1.2 Purpose

It is also known as expense manager and money manager, an expense tracker is a software or application that helps to keep an accurate record of your money inflow and outflow. Many people in India live on a fixed income, and they find that towards the end of the month they don't have sufficient money to meet their needs.

CHAPTER 2

LITERATURE SURVEY

2.1 Existing problem

The problem of the current generation population is that they can't remember where all of the money they earned have gone and ultimately have to live while sustaining the little money they have left for their essential needs. In this time there is no such perfect solution which helps a person to track their daily expenditure easily and efficiently and notify them about the money shortage they have. For doing so they have to maintain long ledger's or computer logs to maintain such data and the calculation is done manually by the user, which may generate errors leading to losses. Not having a complete tracking.

2.2 Reference

1. https://nevonprojects.com/daily-expense-tracker-system/
2. https://data-flair.training/blogs/expense-tracker-python/
3. https://phpgurukul.com/daily-expense-tracker-using-php-and-mysql/
4. https://ijarsct.co.in/Paper391.pdf
5. https://kandi.openweaver.com/?landingpage=python_all_projects&utm_source=google&utm_medium=cpc&utm_campaign=promo_kandi_ie&utm_content=kandi_ie_search&utm_term=python_devs&gclid=Cj0KCQiAgribBhDkARIsAASA5bukrZgbI9UZxzpoyf0P-ofB1mZNxzc-okUP-3TchpYMclHTYFYiqP8aAmmwEALw_wcB

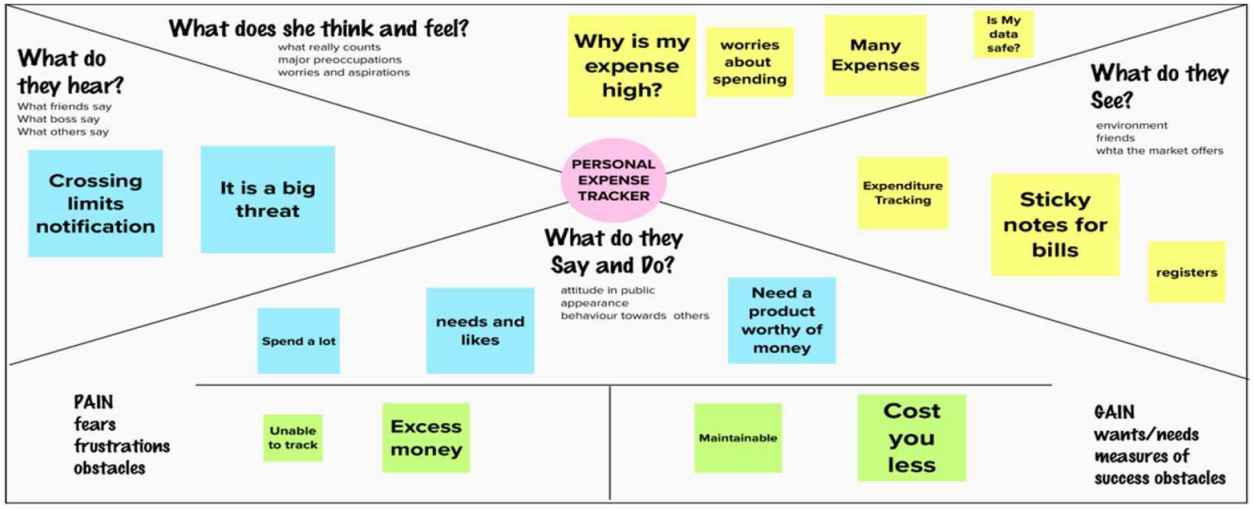
2.3 Problem Statement Definition

This Expense Tracker is a web application that facilitates the users to keep track and manage their personal as well as business expenses. This application helps the users to keep a digital diary. It will keep track of a user's income and expenses on a daily basis. The user will be able to add his/her expenditures instantly and can review them anywhere and anytime with the help of the internet. He/she can easily import transactions from his/her mobile wallets without risking his/her information and efficiently

protecting his/her privacy. It is common to delete files accidentally or misplace files. This expense tracker provides a complete digital solution to this problem. Excel sheets do very little to help in tracking .Furthermore, they don't have the advanced functionality of preparing graphical visuals automatically. Not only will it save the time of the people but also it will assure error free calculations. The user just has to enter the income and expenditures and everything else will be performed by the system. Keywords: Expense Tracker, budget, planning, savings, graphical visualization of expenditure.

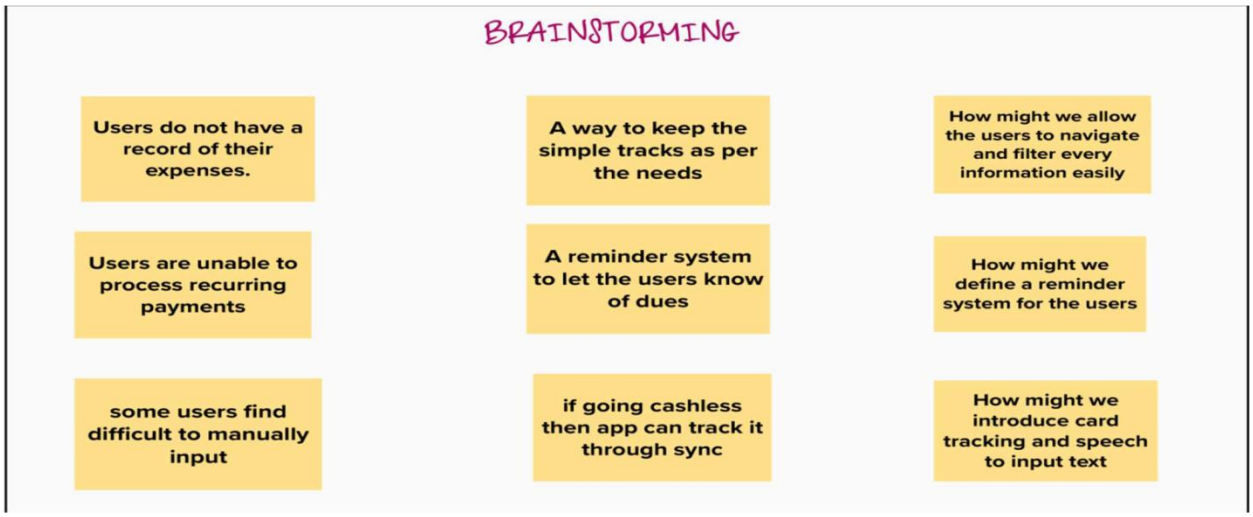
CHAPTER 3

IDEATION & PROPOSED SOLUTION



3.1 Empathy Map canvas

3.2 Ideation & Brainstorming



3.3 Proposed Solution

All people in the earning sector need a way to manage their financial resources and track their expenditure, so that they can improve and monitor their spending habits. This makes them understand the importance of financial management and making better decisions in the future .They have an option to set a limit for the amount to be used for that particular month if the limit is exceeded the user will be notified with an email alert. The solution to this problem is, the people who get regular payments are able to track their payments and avoid unwanted expenses. If the limit is exceeded the user will be notified with an email alert.

3.4 Proposed Solution Fit

The solution to this problem is, the people who gets regular payments can able to track their payments and avoid unwanted expenses. If the limit is exceeded the user will be notified with an email alert.

- Novelty / Uniqueness Notification can be received through email.
- Social Impact / Customer Satisfaction Using this application one can track their personal expenses and frame a monthly/annual budget. If your expense exceeds the specified limit, the application will show you an alert message .This will make an impact on Mobile Banking for Customers' Satisfaction.

- Business Model (Revenue Model) Business people can use the subscription/premium feature of this application to gain revenue.
- Scalability of the Solution The scalability of the application depends on security, the working of the application even during when the network gets down etc…

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 Functional requirement

Following are the functional requirements of the proposed solution.

- FR-1 User Registration ,Registration through Form Registration through Gmail Registration through LinkedIN
- FR-2 User Confirmation ,Confirmation via Email Confirmation via OTP
- FR-3 Tracking Expense Helpful insights about money management
- FR-4 Alert Message Give alert mail if the amount exceeds the budget limit
- FR-5 Category This application shall allow users to add categories of their expenses

4.2 Non Functional requirement

Following are the non-functional requirements of the proposed solution.

- NFR-1 Usability You will able to allocate money to different priorities and also help you to cut down on unnecessary spending
- NFR-2 Security More security of the customer data and bank account details.
- NFR-3 Reliability Used to manage his/her expense so that the user is the path of financial stability. It is categorized by week, month, and year and also helps to see more expenses made. Helps to define their own categories.

- NFR-4 Performance The types of expense are categories along with an option .Throughput of the system is increased due to light weight database support.
- NFR-5 Availability Able to track business expenses and monitor important for maintaining healthy cash flow. NFR-6 Scalability The ability to appropriately handle increasing demands.

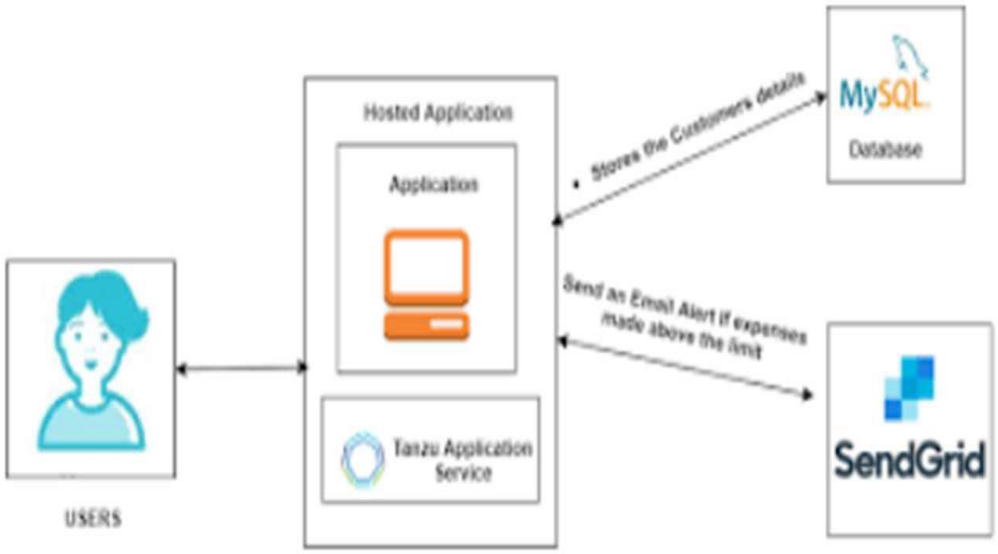
CHAPTER 5

PROJECT DESIGN

5.1 Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

5.2 Solution & Technical Architecture



5.3 User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story/ Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password,and confirming my password.	I can access my account / dashboard	High	Sprint 1

		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint 1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint 2
		USN-4	As a user, I can register for the application through Gmail	I can register by entering the details	Medium	Sprint 1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can access my dashboard	High	Sprint 1
	Dashboard	USN-6	As a user ,I can log into the dashboard and manage income	I can access my account / dashboard	High	Sprint 1
Customer (Web user)		USN-7	As a user, I can register for the application by Bank account.	I can access my account / dashboard	High	Sprint 1

Customer Care Executive		USN-8	As a user, I can get a report is based on the details	I can manage my money by viewing this report	Medium	Sprint 2
		USN-9	As a user, I can get an email if the money level is above the limit	I can receive alert email	High	Sprint 1
Administrative	Responsibility	USN-10	As a system administrator, track the user expenses anytime	I can track expense	High	Sprint 1

CHAPTER 6

PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement(Epic)	User Story Number	User Story/Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As user ,I can register for the application by entering my email, password ,and confirming my password.	2	High	Ramadevi Trisha

Sprint-1		USN-2	As a user ,I will receive confirmation email once I have registered for the application	1	Medium	Ramadevi Trisha
Sprint-2	Login	USN-3	As a user ,I can register for the application through Facebook	2	Medium	Keerthana Jackharish
Sprint-1	Dashboard	USN-4	As a user, I can register for the application through Gmail	2	High	Hariharan

6.2 Sprint Delivery Schedule

Sprint	**Total Story Points**	**Duration**	**Sprint Start Date**	**Sprint End Date (Planned)**	**Story Points Completed(as on Planned End Date)**	**Sprint Release Date(Actual)**
Sprint-1	20	6 Days	24Oct2022	29Oct2022	20	29Oct2022
Sprint-2	20	6 Days	31Oct2022	05Nov2022	18	06Nov2022
Sprint-3	20	6 Days	07Nov2022	12Nov2022	15	14Nov2022
Sprint-4	20	6 Days	14Nov2022	19Nov2022	19	21Nov2022

CHAPTER 7

Coding And Solutioning

7.1. Features

Feature 1: Add Expense
Feature 2: Update expense
Feature 3: Delete Expense
Feature 4: Set Limit
Feature 5: Send Alert Emails to users

7.2. Other Features:

Track your expenses anywhere, anytime. Seamlessly manage your money and budget without any financial paperwork. Just click and submit your invoices and expenditures. Access, submit, and approve invoices irrespective of time and location. Avoid data loss by scanning your tickets and bills and saving in the app. Approval of bills and expenditures in real-time and get notified instantly. Quick settlement of claims and reduced human errors with an automated and streamlined billing process.

Codes:

```
from flask import Flask, render_template, request, redirect, session
import ibm_db
import re
app = Flask(__name__)
app.secret_key = 'a'
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=b1bc1829-6f45-
4cd4-bef4-
10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=323
04;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID
=vzr37860;PWD=XOTSaOmkJNfmcxQt;","","")
#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 ( % 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 ( % 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 ORDER BY date DESC'.format(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  userid = %s', (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  userid = %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`.`userid` = % 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 (% s, % s)
',(session['id'], number))
        mysql.connection.commit()
        return redirect('/limitn')
   @app.route("/limitn")
def limitn():
   cursor = mysql.connection.cursor()
   cursor.execute('SELECT limits 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()
     print ("HI")
     #cursor.execute('SELECT TIME(date)   , amount FROM expenses
WHERE userid = {0} AND DATE(date) = DATE(NOW())
'.format(str(session['id'])))
     cursor.execute('SELECT TIME(date)   , amount FROM expenses
WHERE userid = %s AND DATE(date) = DATE(NOW()) ',(id,))
```

```
    texpense = cursor.fetchall()
    print(texpense)

    cursor = mysql.connection.cursor()
    print("HIII")
    #cursor.execute('SELECT * FROM expenses WHERE userid = {0} AND
DATE(date) = DATE(NOW()) AND date ORDER BY `expenses`.`date`
DESC'.format(str(session['id'])))
    cursor.execute('SELECT * FROM expenses WHERE userid = %s AND
DATE(date) = DATE(NOW()) AND date ORDER BY `expenses`.`date`
DESC',(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)
```

The other code features are submitted in [github]

CHAPTER 8

TESTING

8.1. TESTING:

- Login Page (Funcional)
- Login Page (UI)
- Add Expense Page (Functional)

8.2. User Acceptance Testing:

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	8	15
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	9	2	4	11	20
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2

Won't Fix	0	5	0	1	8
Totals	22	14	11	22	51

3. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

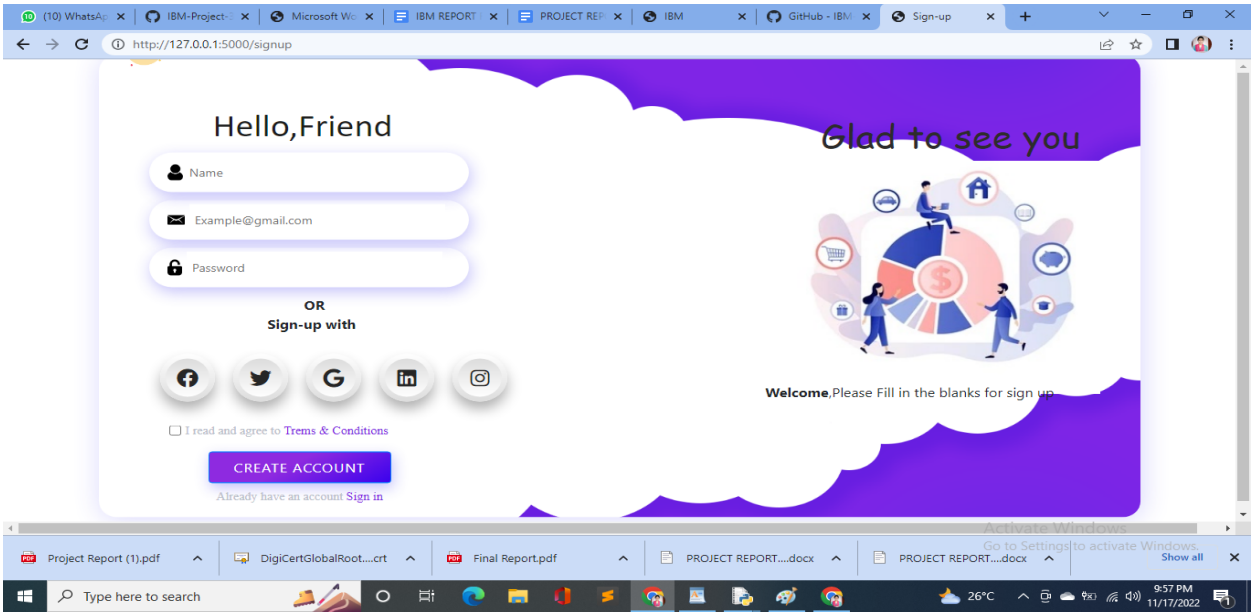
Section	Total Cases	Not Tested	Fail	Pass
Interface	7	0	0	7
Login	43	0	0	43
Logout	2	0	0	2

CHAPTER 8

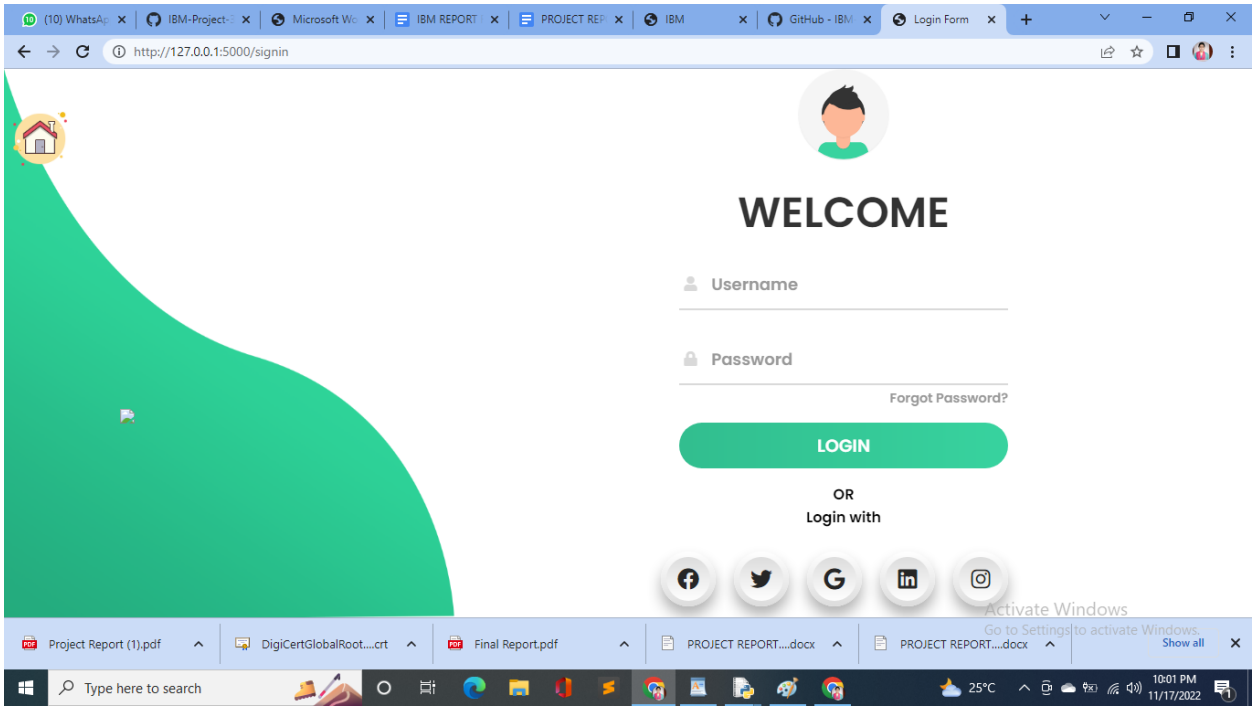
RESULTS

9.1 HomePage

9.2 Sign Up Page



9.3 Login Page



CHAPTER 10

ADVANTAGES AND DISADVANTAGES

10.1. ADVANTAGES:

One of the major pros of tracking spending is always being aware of the state of one's personal finances. Tracking what you spend can help you stick to your budget, not just in a general way, but in each category such as housing, food, transportation and gifts. While a con is that manually
tracking all cash that is spent can be irritating as well as time consuming, a pro is that doing this automatically can be quick and simple.
Another pro is that many automatic spending tracking software programs are available for free. Having the program on a hand-held device can be a main pro since it can be checked before spending occurs in order to be sure of the available budget. Another pro is that for those who just
I wish to keep tracking spending by hand with a paper and pen or by entering data onto a computer spreadsheet, these options are also available. Some people like to keep a file folder or box to store receipts and record the cash spent each day. A pro of this simple daily tracking system is that it can make one more aware of where the money is going before the end of a pay period or month.

10.2. DISADVANTAGES:

A con with any system used to track spending is that one may start doing it then taper off until it's forgotten about all together. Yet, this is a risk for any new goal such as trying to lose weight or quit smoking. If a person first makes a budget plan, then places money in savings before spending any new pay period or month, the tracking goal can help. In this way, tracking spending and making sure all receipts are accounted for only needs to be done once or twice a month. Even with constant tracking of one's spending habits, there is no guarantee that financial goals will be met. Although this can be considered to be a con of tracking spending, it could be changed into a pro if one makes up his or her mind to keep trying to properly manage all finances. Another con that may occur when spending is being tracked is an error, but this may also be able to be changed into a pro if the person does regular tracking. Frequent tracking of cash spending can allow one to catch and correct errors so that the budget plan is still able to be

adhered to despite the mistake.

CHAPTER 11

CONCLUSION

A comprehensive money management strategy requires clarity and conviction for decision making. You will need a defined goal and a clear vision for grasping the business and personal finances. That's when an expense tracking app comes into the picture. An expense tracking app is an exclusive suite of services for people who seek to handle their earnings and plan their expenses and savings efficiently. It helps you track all transactions like bills, refunds, payrolls, receipts, taxes, etc., on a daily, weekly, and monthly basis.

CHAPTER 12

FUTURE SCOPE

- Achieve your business goals with a tailored mobile app that perfectly fits your business.
- Scale-up at the pace your business is growing.
- Deliver an outstanding customer experience through additional control over the app.
- Control the security of your business and customer data.
- Open direct marketing channels with no extra costs with methods such as push notifications.
- Boost the productivity of all the processes within the organization.
- Increase efficiency and customer satisfaction with an app aligned to their needs.
- Seamlessly integrate with existing infrastructure.
- Ability to provide valuable insights.
- Optimize sales processes to generate more revenue through enhanced data collection.
- Robo Advisors: Get expert investment advice and solutions with the Robo-advisors

- feature. This feature will analyze, monitor, optimize, and improve diversification in investments by turning data into actionable insights in real-time. Chats: Equip your expense tracking app with a bot that can understand and answer all user queries and address their needs such as account balance, credit score, etc.
- Prediction: With the help of AI, your mobile app can predict your next purchase, according to your spending behavior. Moreover, it can recommend products and provide unique insights on saving money. It brings out the factors causing fluctuations in your expenses.
- Employee Travel Budgeting: Most businesses save money with a travel budgeting app as It helps prepare a budget for an employee's entire business trip. The feature will predict the expenses and allocate resources according to the prediction.

APPENDIX:

The source code , demo video and github are in the link (github)