

PERSONAL EXPENSE TRACKER APPLICATION

[IBM-Project-23819-1659931639](#)

NALAIYA THIRAN PROJECT
BASED LEARNING ON
PROFESSIONAL
READLINESS FOR
INNOVATION,
EMPLOYNMENT AND
ENTERPRENEURSHIP

**A PROJECT REPORT
BY**

AAKASH R(111719205002)
ASHIK ESWAR (111719205007)
LOKESHWAR B(111719205021)
MEGARAJ S(111719205025)

**BACHELOR OF TECHNOLOGY
IN INFORMATION TECHNOLOGY**

RMK ENGINEERING COLLEGE,CHENNAI -601206

INDEX

1. INTRODUCTION

- a. Project Overview
- b. Purpose

2. LITERATURE SURVEY

- a. Existing Problem
- b. References
- c. Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION

- a. Empathy Map Canvas
- b. Ideation & Brainstorming
- c. Proposed Solution
- d. Problem Solution fit

4. REQUIREMENT ANALYSIS

- a. Functional requirement
- b. Non-Functional requirements

5. PROJECT DESIGN

- a. Data Flow Diagrams
- b. Solution & Technical Architecture
- c. User Stories

6. PROJECT PLANNING & SCHEDULING

- a. Sprint Planning & Estimation
- b. Sprint Delivery Schedule
- c. Reports from JIRA

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

- a. Feature 1
- b. Feature 2
- c. Database Schema (if Applicable)

8. TESTING

- a. Test Cases
- b. User Acceptance Testing

9. RESULTS

- a. Performance Metrics

10. ADVANTAGES & DISADVANTAGES

11. CONCLUSION

12. FUTURE SCOPE

13. APPENDIX

Source Code

GitHub & Project Demo Link

Skills Required:

IBM Cloud, HTML, Javascript, IBM Cloud Object Storage, Python- Flask, Kubernetes, Docker, IBM DB2, IBM Container Registry

1. INTRODUCTION**a. Project Overview**

This project is based on expense tracking. This project aims to create an easy, faster and smooth cloud application. For better expense tracking we developed our project that will help the users a lot. Most of the people cannot track their expenses and income leading to facing money crisis, so this application can help people to track their expense day to day and make life stress free. Money is the most valuable portion of our daily life and without money we will not last one day on earth. So using the daily expense tracker application is important to lead a happy family. It helps the user to avoid unexpected expenses and bad financial situations. It will save time and provide a responsible lifestyle.

b. Purpose

Personal finance management is an important part of people's lives. However, everyone does not have the knowledge or time to manage their finances in a proper manner. And, even if a person has time and knowledge, they do not bother with tracking their expenses as they find it tedious and time-consuming. Now, you don't have to worry about managing your expenses, as you can get access to an expense tracker that will help in the active management of your finances. 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. While this problem can arise due to low salary, invariably it is due to poor money management skills. People tend to overspend without realizing, and this can prove to be disastrous. Using a daily expense manager can help you keep track of how much you spend every day and on what. At the end of the month, you will have a clear picture where your money is going. This is one of the best ways to get your expenses under control and bring some semblance of order to your finances. Today, there are several expense manager applications in the market. Some are paid managers while others are free. Even banks like ICICI offer their customers expense tracker to help them out. Before you decide to go in for a money manager, it is important to decide the type you want.

2. LITERATURE SURVEY

a. Existing problem

In a study conducted by Forrester in 2016 surveying small and medium businesses (SMBs) across the world, 56% of companies reported expense management as being the biggest challenge for their finance departments.

In another survey conducted by Level Research in 2018 in North America, respondents reported the following pain points in expense management before adopting automation:

- i. Manual entry and routing of expense reports (62%)
- ii. Lack of visibility into spend data (42%)
- iii. Inability to enforce travel policies (29%)
- iv. Lost expense reports (24%)
- v. Lengthy expense approval system and reimbursement cycles (23%)

b. References

S.No	TITLE	PROPOSED WORK	TOOLS USED/ ALGORITHM	TECHNOLOGY	ADVANTAGES/ DISADVANTAGES
1.	EXPENSE MANAGER APPLICATION. (2020)	To Develop A Moblie Application That Keeps Record Of User Personal Expenses Contribution In Group Expenditure Top Investment Options View Of The Current Stock Market ,Read Authenticated Financial News	Android Studio	Cloud Application	Advantages: ➤ Keeps Track All Of Your Daily Transactions, Keeps Track Of Your Money Lent Or Borrowed. Disadvantages: ➤ Occupy Lot Of Space.
2.	A NOVEL EXPENSE TRACKER USING STATISTICAL ANALYSIS. (2021)	To Maintain And Manage Data Of Daily Expenditure In A More Precise Way.	SQL Lite	Cloud Application	Advantages: ➤ Its Suggest You With The Most Effective Investment Options. Disadvantages: ➤ The Work Done Being Is Not Accurate.

S.No	TITLE	PROPOSED WORK	TOOLS USED/ ALGORITHM	TECHNOLOGY	ADVANTAGES/ DISADVANTAGES
3.	EXPENSE TRACKER. (2021)	Facilitates The User To Keep Track And Manage Their Personal As Well As Business Expenses.	Android OS	Cloud Application	Advantages: ➤ Become Aware Of Poor Spending Habits And Take Care Of Your Finances Saving And Investment. Disadvantages: ➤ Searching And Referencing Is Difficult And Time-consuming.
4.	EXPENSE TRACKER. (May 2021)	The Application Keeps The Track Of The Income And Expenses Both Of User On A Day To Day Bases	Java	Cloud Application	Advantages: ➤ The Project Effectively Keeps Away From The Manual Figuring. Disadvantages: ➤ Report Generation Is A Tedious Process.

I am	Describe customer with 3-4 key characteristics - who are they?	Describe the customer and their attributes here
I'm trying to	List their outcome or "job" the care about - what are they trying to achieve?	List the thing they are trying to achieve here
but	Describe what problems or barriers stand in the way - what bothers them most?	Describe the problems or barriers that get in the way here
because	Enter the "root cause" of why the problem or barrier exists - what needs to be solved?	Describe the reason the problems or barriers exist
which makes me feel	Describe the emotions from the customer's point of view - how does it impact them emotionally?	Describe the emotions the result from experiencing the problems or barriers

Personal Expense Tracker Application:

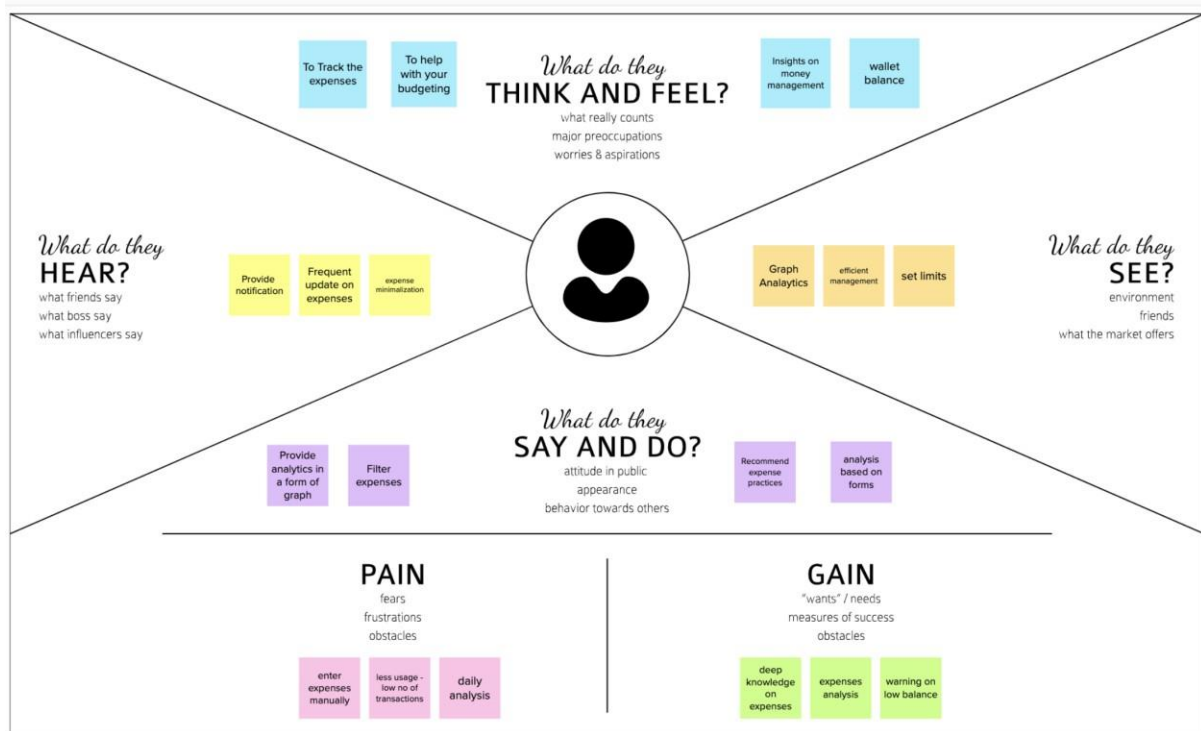


3.Problem Statement Definition Customer Problem Statement :

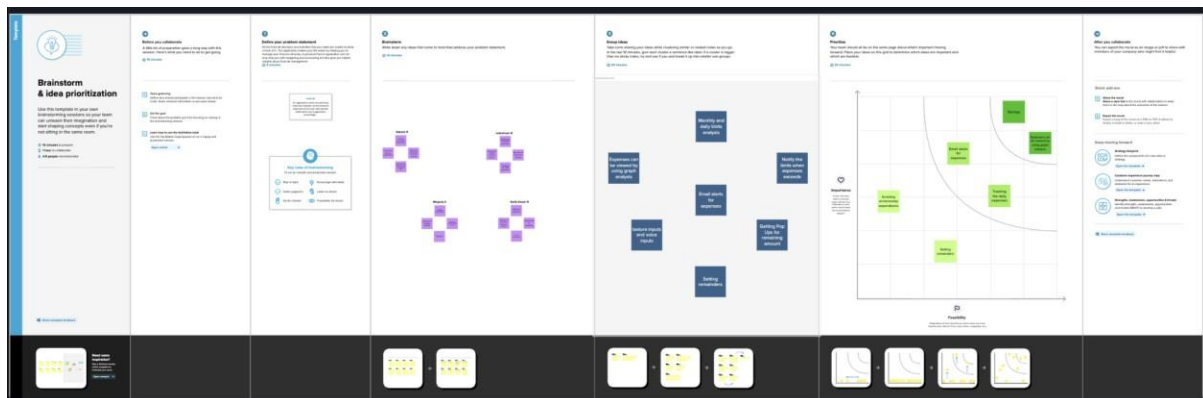
A well-articulated customer problem statement allows us to find the ideal solution for the challenges our customers face. Throughout the process, you'll also be able to empathize with your customers, which helps you better understand how they perceive your product or service.

1. IDEATION & PROPOSED SOLUTION

a. Empathy Map Canvas



B. Ideation & Brainstorming

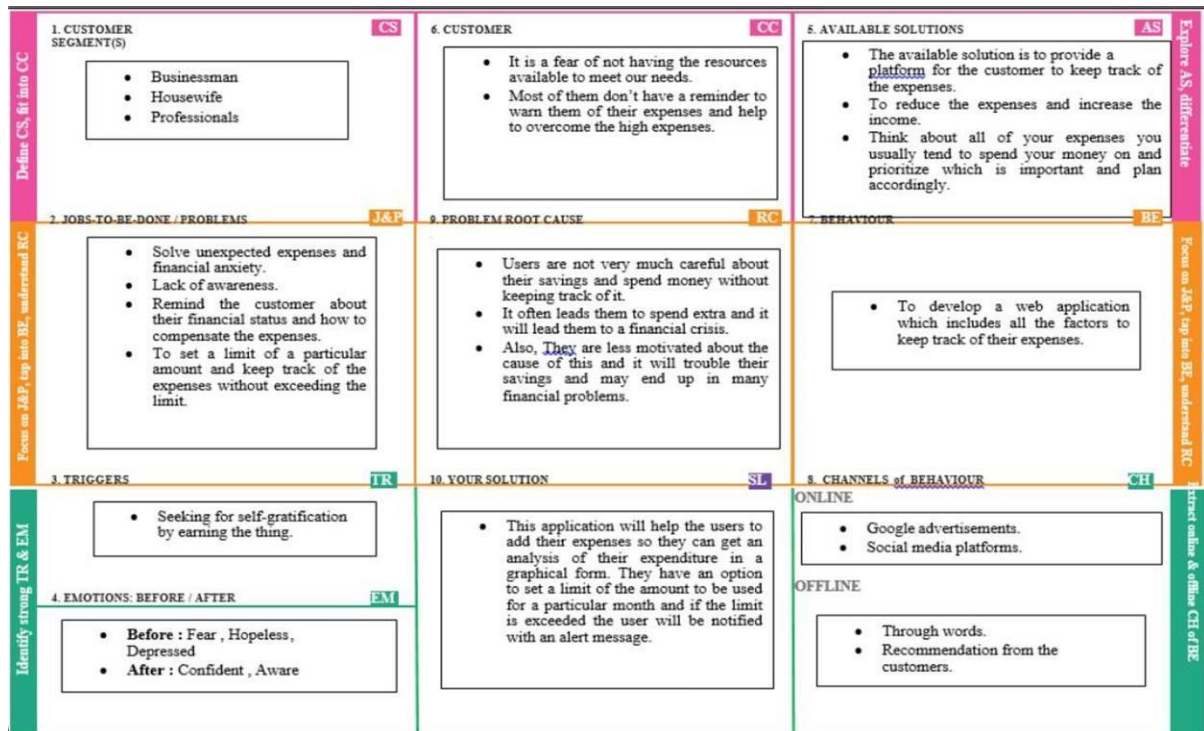


c. Proposed Solution

S.NO	Parameter	Description
1.	Problem Statement (Problem to be solved)	<p>Keeping Proper track of our daily expenses is becoming challenging in today's world. Without the proper money management knowledge people overspend on their wants instead of focusing on their needs.</p> <p>Especially when using online applications for purchasing their requirements consumers tend to over spend. This problem leads to improper distribution of their daily expenses. Without proper knowledge on managing money poor are becoming poorer and rich are becoming richer.</p>
2.	Idea / Solution description	<p>An attempt to develop an app to manage our daily expenses and give us insights on managing our money would be a good idea. This app will be able to track expenses on various online platforms and apps. The app can help with proper budgeting and give alerts when the user over spends or crosses the limit previously set by them. This will lead to proper spending habits and make them knowledgeable about money management. IBM cloud can be used to handle the data safely.</p>
3.	Novelty / Uniqueness	<p>The speciality for the app will be the data security with IBM cloud being used for data storage and this app genuinely helps with the money management. The proper and personalized budgeting of the user's money leads them to trust the app and they wouldn't have to worry about their expenditure on unnecessary things.</p>
4.	Social Impact / Customer Satisfaction	<p>People using the app will be becoming better at their spending habits and will be able to save more than their peers who are not using the app. This application aims to improve the users' savings sustainably and steadily which leads them to trust the app without worrying about the</p>

5.	Business Model (Revenue Model)	This application leads to a business model, the user can be suggested the right products to buy based on their budget and this can lead to targeted business approaching the right consumers. The model leads to systematic and structured expenses of the user and also leads to predictive analysis of the future expenses of the consumer. This model makes the user more careful with expenses as they are provided with the money management insights.
6.	Scalability of the Solution	This application can be created as a multi user model nationwide. The model can also be modified based on the country's law on applications and data security which leads to international implementation of this application by maintaining proper gateway rules. This app when developed for multiple nations can be modified to their requirements. The app can also be modified for a particular group of people or organization.

d. Problem Solution Fit



4. REQUIREMENT ANALYSIS

a. Functional requirements

FR No.	Functional Requirement	Description
FR-1	Register	Registration is the process of the user to complete the application's form. Certain details must be submitted such as e-mail address, password, and password confirmation. The user is identified using these details.
FR-2	Login	The login screen is used to verify the identity of the user. The account can be accessed using the user's registered email address and password.
FR-3	Categories	On the main page, we can see overall revenue and spending, as well as the balance remaining after expenditure, as well as the user's entire categories namely Entertainment, Cloth, Food and Drinks, Health and Fitness and so on.

FR-4	Update Daily Expensive	The user can upload the daily expensive details what they are spending on each day. The details such as cloth, entertainment, food, health etc.,
FR-5	View Expensive Chart	This module used to see a pictorial depiction of all details in the form of a pie chart, where each slice of the pie chart represents that the viewer to gain an approximatention of which category has the highest expenses.
NFR-6	Set Alert	When a user attempts to spend more than the pre-defined amount limit, the app will automatically send an alert if the threshold amount they selected for an alert is exceeded.

b. Non-Functional requirements

NFR No.	Non-Functional Requirement	Description
NFR-1	Usability	The system shall allow the users to access the system with pc using web application. The system uses a web application as an interface. The system is user friendly which makes the system easy.
NFR-2	Security	A security requirement is a statement of needed security functionality that ensures one of many different security properties of software is being satisfied.

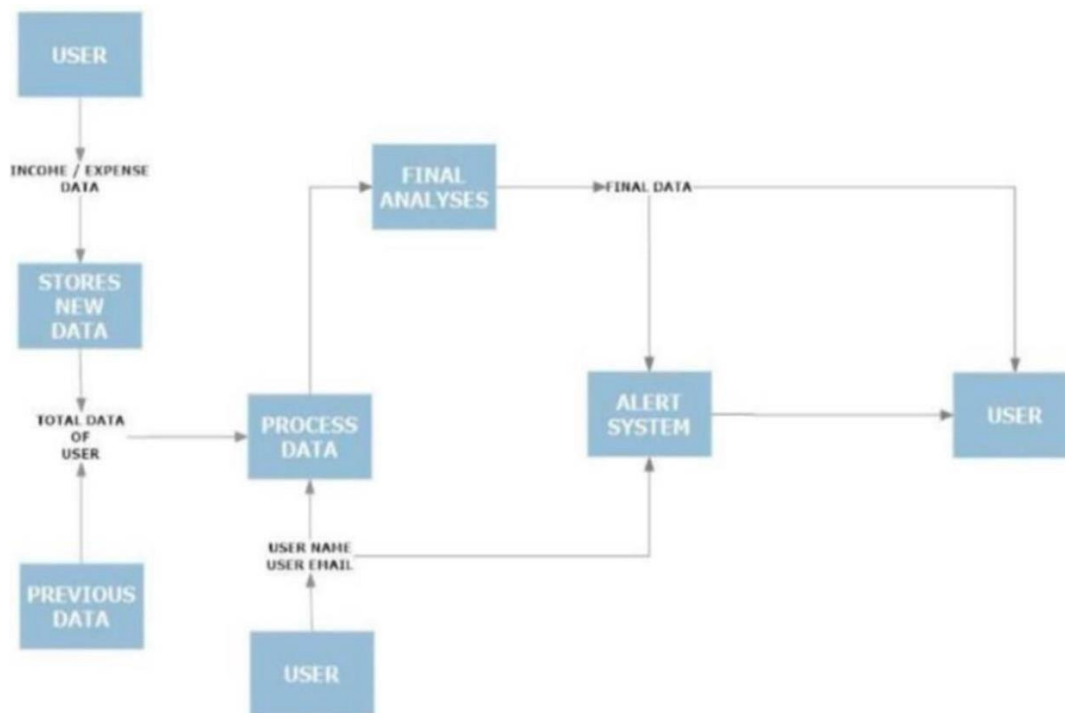
NFR-3	Reliability	he system has to be 100% reliable due to the importance of data and the damages that can be caused by incorrect or incomplete data. The system will run 7 days a week. 24 hours a day.
NFR-4	Performance	The information is refreshed depending upon whether some updates have occurred or not in the application. The system shall respond to the member in not less than two seconds from the time of the request submittal. The system shall be allowed to take more time when doing large processing jobs. Responses to view information shall take no longer than 5 seconds to appear on the screen.
NFR-5	Availability	The system is available 100% for the user and is used 24 hrs a day and 365 days a year. The system shall be operational 24 hours a day and 7 days a week.

NFR-6	Scalability	Scalability is the measure of a system's ability to increase or decrease in performance and cost in response to changes in application and system processing demands.
-------	-------------	---

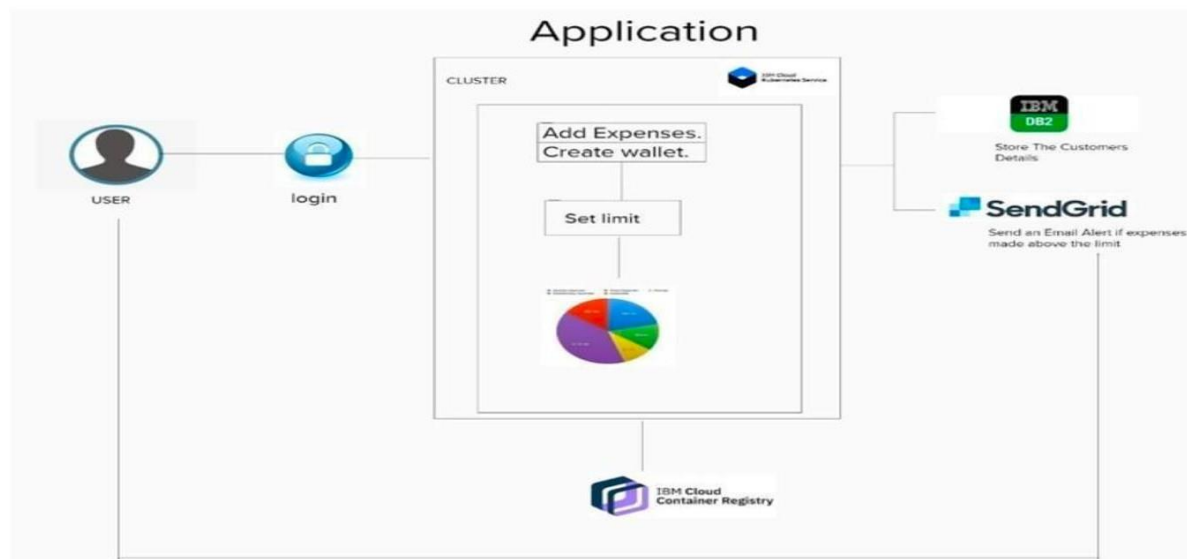
5. PROJECT DESIGN

a. 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.



B. Solution & Technical Architecture



C. User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority
Customer (web user)	Registration	USN-1	As a user, I can register for the application by entering mail id and password	I can access my account/ dashboard	High
		USN-2	As a user, I will receive a confirmation email once I have registered for the email and click application	I can receive a confirmation email	High
USN-3 As a user, I can I can register Low access using mail through mail					
	Login	USN-4	As a user, I can login application by entering application using email and password	I can access the application	High
	Dashboard	USN-5	As a user, I can view my income and expenditure details	I can view my daily expenses	High
Customer care executive		USN-6	As a customer care executive, I can solve the login issue and other issues of the solution at any application	I can provide support	Medium

Administrator	Application	USN-7	As an administrator,I can upgrade or update the application	I can fix the bug	Medium
---------------	-------------	-------	---	-------------------	--------

6.PROJECT PLANNING & SCHEDULING

a. Sprint planning and estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Lokeshwar
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Lokeshwar
Sprint-1	Login	USN-3	As a user, I used my Mail id and password for login	2	High	Ashik Eswar
Sprint-2		USN-4	As a user, I forget my password. Used forget password,	2	High	Aakash
Sprint-2	Dashboard	USN-5	As a user, there is profile tab	1	High	Megaraj
Sprint-2		USN-6	As a user, there is budget tab	2	High	Megaraj
Sprint-3	Budget	USN-7	As a user, I create a budget, update the budget	1	Low	Aakash
Sprint-3		USN-9	As a user, I can enter my expense into category	2	High	Ashik Eswar
Sprint-4	Report	USN-10	As a user, I get a expense report anytime I need	2	High	Lokeshwar
Sprint-4	Chat bot	USN-12	As a admin, chat bot helps to get familiar with application	2	High	Megaraj

MILESTONES Preparation Phase

1.	Preparation Phase	Prerequisites	24 Aug 2022
		Príor Knowledge	25 Aug 2022
		Project Structure	23 Aug 2022
		Project Flow	23 Aug 2022
		Project Objective	22 Aug 2022
		Registration	26 Aug 2022
		Environment Set-up	27 Aug 2022
2.	Ideation Phase	Literature Survey	29 Aug 2022 – 03 Sept 2022
		Empathy Map	5 Sept 2022 - 7 Sept 2022
		Problem Statement	8 Sept 2022 - 10 Sept 2022
		Ideation	12 Sept 2022 – 16 Sept 2022

3.	Project Design Phase -1	Proposed Solution	19 Sept 2022 – 23 Sept 2022
----	-------------------------	-------------------	-----------------------------

4.	Project Design Phase -2	Customer Journey Map	03 Oct 2022 – 08 Oct 2022
		Requirement Analysis	09 Oct 2022 – 11 Oct 2022
		Data Flow Diagrams	11 Oct 2022 – 14 Oct 2022
		Technology Architecture	15 Oct 2022 - 16 Oct 2022
5.	Project Planning Phase	Milestones & Tasks	17 Oct 2022 – 18 Oct 2022
		Sprint Schedules	19 Oct 2022 – 22 Oct 2022
6.	Project Development Phase	Sprint - 1	24 Oct 2022 – 29 Oct 2022
		Sprint – 2	31 Oct 2022 – 05 Nov 2022
		Sprint – 3	07 Nov 2022 – 12 Nov 2022
		Spínt - 4	14 Nov 2022 -19 Nov 2022

a. Reports from JIRA

i. Backlog

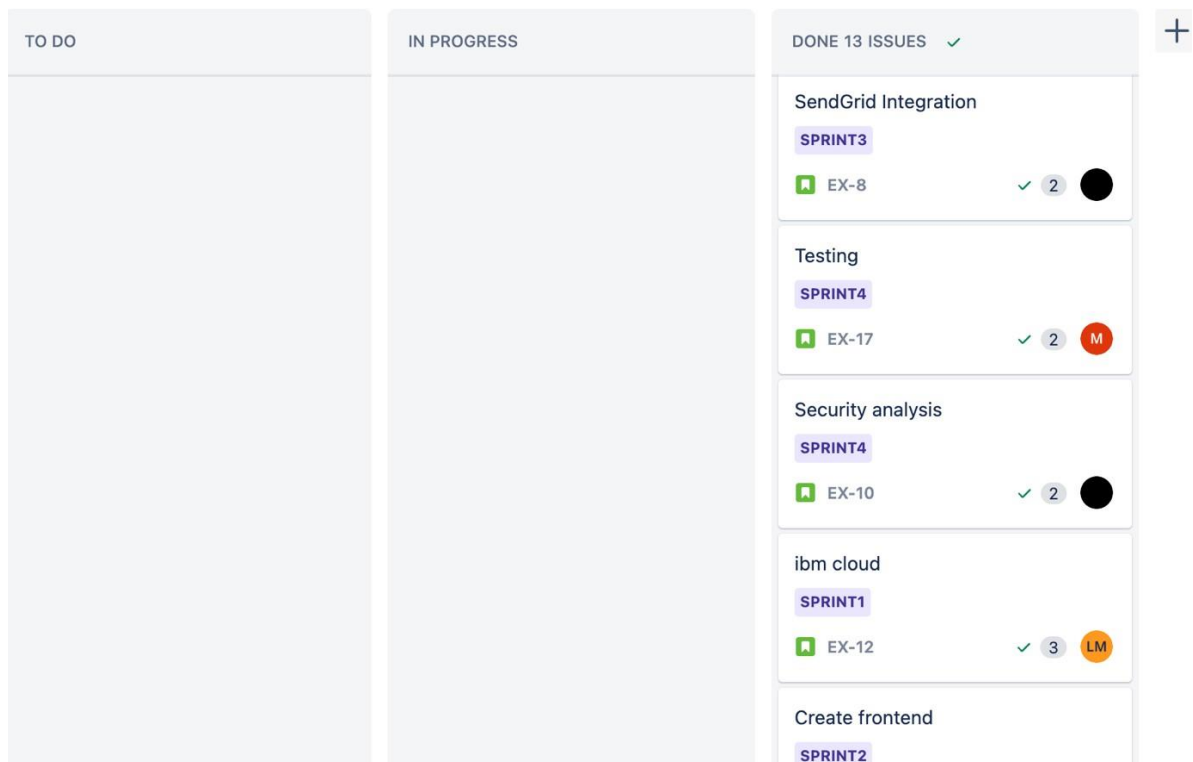
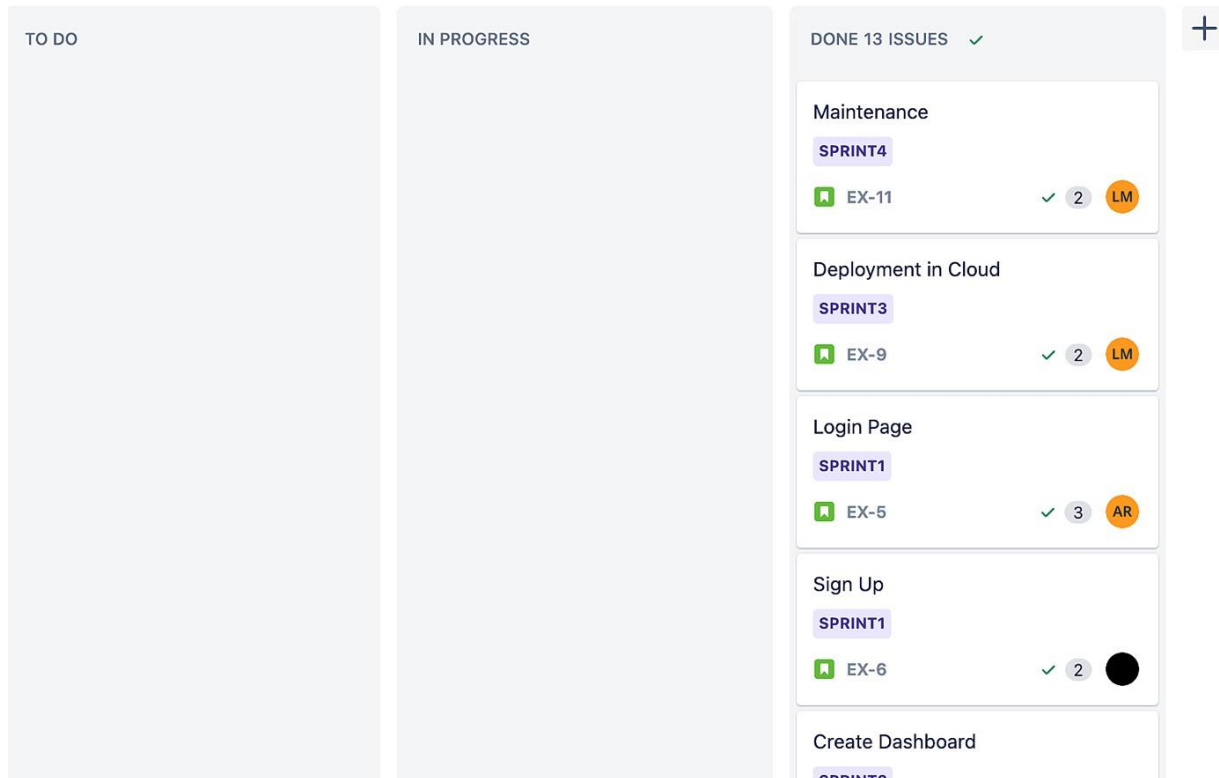
▼ EX Sprint 1 6 Nov – 10 Nov (5 issues)		0 0 14	Complete sprint	...
EX-5 Login Page	SPRINT1	3	DONE	AR
EX-6 Sign Up	SPRINT1	2	DONE	
EX-12 ibm cloud	SPRINT1	3	DONE	LM
EX-14 Dockers	SPRINT1	4	DONE	AR
EX-13 Kubernetes	SPRINT1	2	DONE	M
+ Create issue				

▼ EX Sprint 2 6 Nov – 10 Nov (2 issues)		0 0 4	Complete sprint	...
EX-7 Create Dashboard	SPRINT2	2	DONE	AR
EX-15 Create frontend	SPRINT2	2	DONE	
+ Create issue				

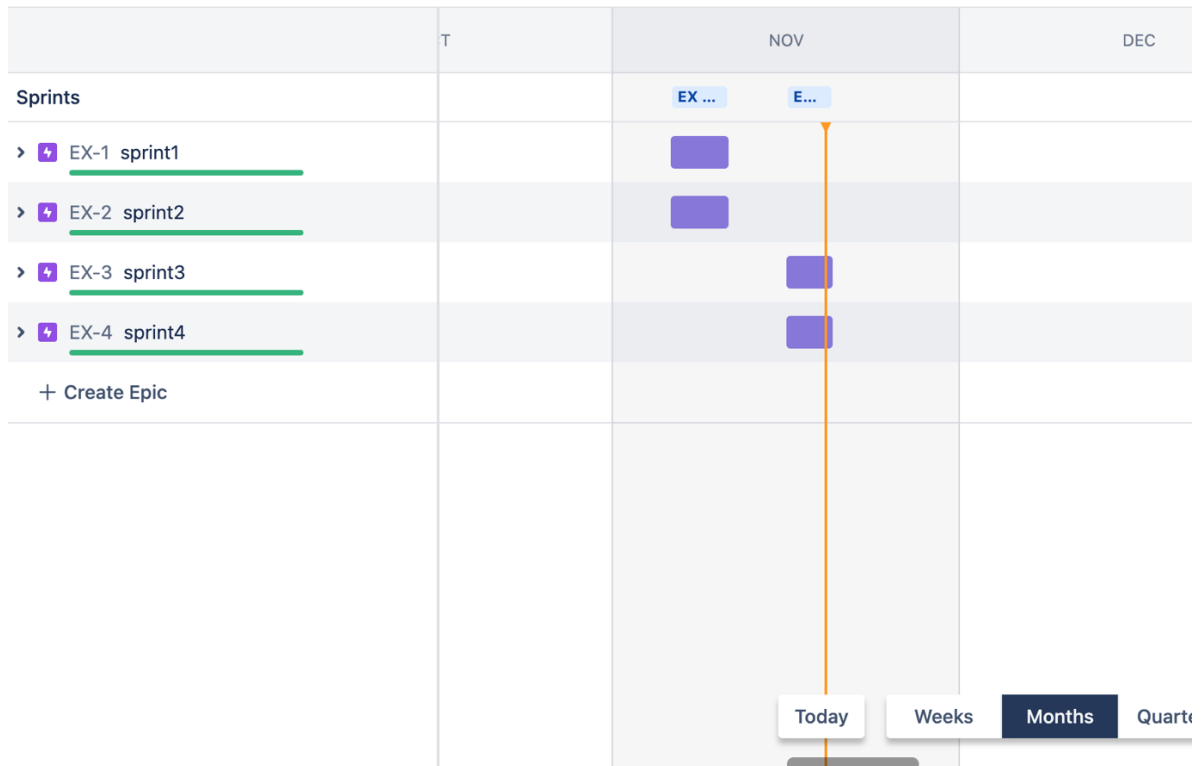
▼ EX Sprint 3 16 Nov – 19 Nov (3 issues)		0 0 7	Complete sprint	...
EX-9 Deployment in Cloud	SPRINT3	2	DONE	LM
EX-8 SendGrid Integration	SPRINT3	2	DONE	
EX-16 Create Backend	SPRINT3	3	DONE	M
+ Create issue				

▼ EX Sprint 4 16 Nov – 19 Nov (3 issues)		0 0 6	Complete sprint	...
EX-11 Maintenance	SPRINT4	2	DONE	LM
EX-17 Testing	SPRINT4	2	DONE	M
EX-10 Security analysis	SPRINT4	2	DONE	
+ Create issue				

ii. Board

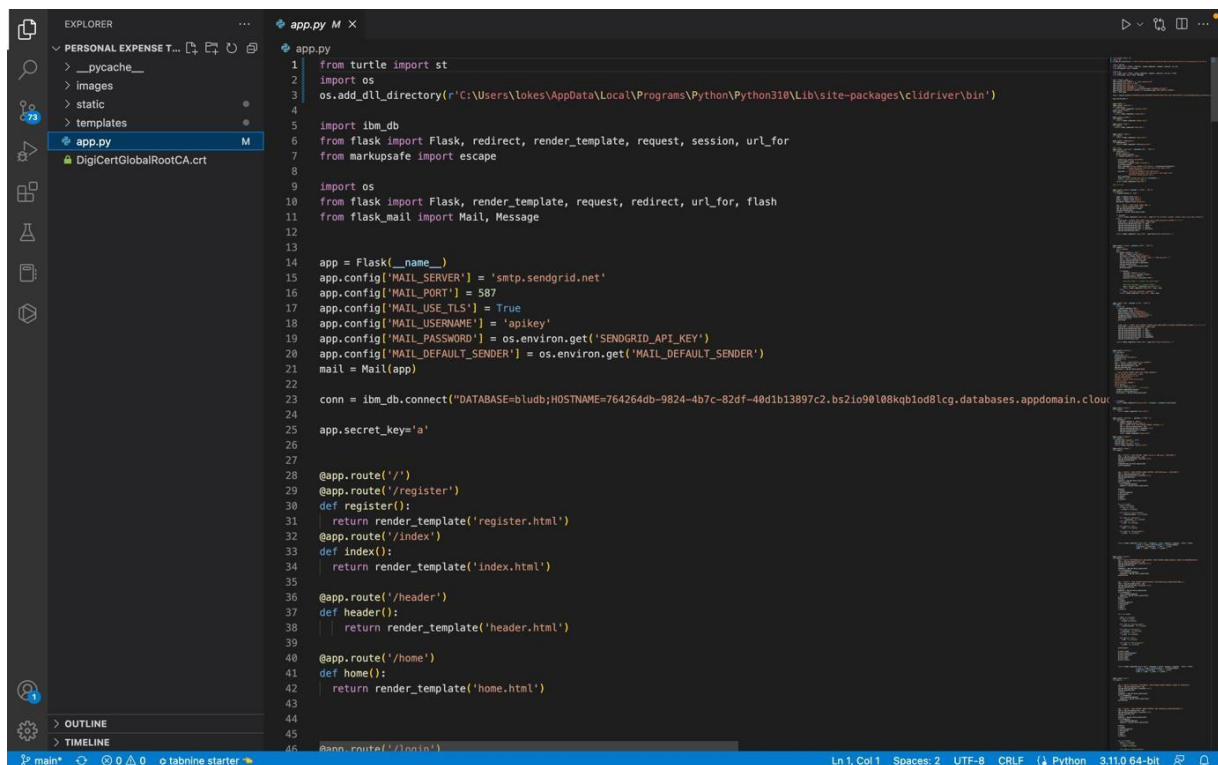


iii. Road Map

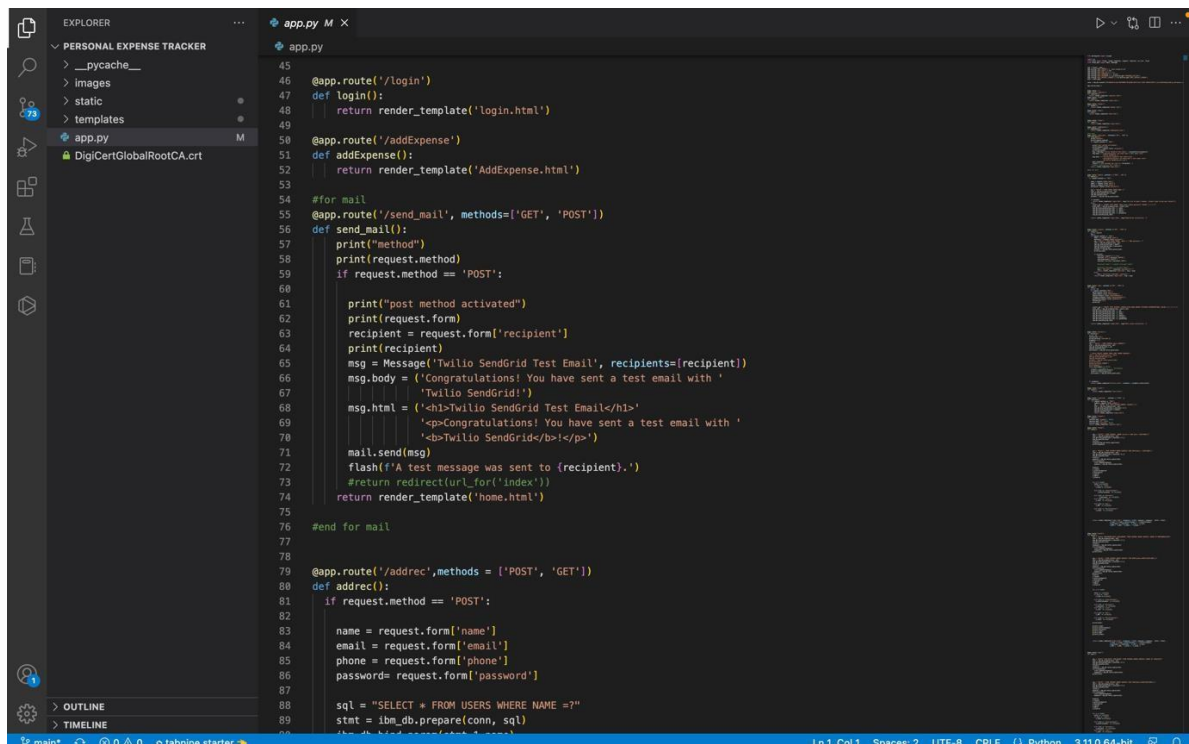


7. CODING & SOLUTIONING

app.py:



```
1 from turtle import st
2 import os
3 os.add_dll_directory(r'C:\Users\lokes\AppData\Local\Programs\Python\Python310\Lib\site-packages\clidriver\bin')
4
5 import ibm_db
6 from flask import Flask, redirect, render_template, request, session, url_for
7 from markupsafe import escape
8
9 import os
10 from flask import Flask, render_template, request, redirect, url_for, flash
11 from flask_mail import Mail, Message
12
13
14 app = Flask(__name__)
15 app.config['MAIL_SERVER'] = 'smtp.sendgrid.net'
16 app.config['MAIL_PORT'] = 587
17 app.config['MAIL_USE_TLS'] = True
18 app.config['MAIL_USERNAME'] = 'apikey'
19 app.config['MAIL_PASSWORD'] = os.environ.get('SENDGRID_API_KEY')
20 app.config['MAIL_DEFAULT_SENDER'] = os.environ.get('MAIL_DEFAULT_SENDER')
21 mail = Mail(app)
22
23 conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=764264db-9824-4b7c-82df-40d1b1397c2.bs2io90l08kqb1od8lcg.databases.apppdomain.cloud;UID=bludb;PWD=bludb")
24
25 app.secret_key='a'
26
27
28 @app.route('/')
29 @app.route('/register')
30 def register():
31     return render_template('register.html')
32 @app.route('/index')
33 def index():
34     return render_template('index.html')
35
36 @app.route('/header')
37 def header():
38     return render_template('header.html')
39
40 @app.route('/home')
41 def home():
42     return render_template('home.html')
43
44
45 @app.route('/login')
```



```
45
46 @app.route('/login')
47 def login():
48     return render_template('login.html')
49
50 @app.route('/addExpense')
51 def addExpense():
52     return render_template('AddExpense.html')
53
54 #for mail
55 @app.route('/send_mail', methods=['GET', 'POST'])
56 def send_mail():
57     print("method")
58     print(request.method)
59     if request.method == 'POST':
60
61         print("post method activated")
62         print(request.form)
63         recipient = request.form['recipient']
64         print(recipient)
65         msg = Message('Twilio SendGrid Test Email', recipients=[recipient])
66         msg.body = f'Congratulations! You have sent a test email with '
67             f' {request.form["recipient"]} '
68             f' {request.form["password"]} '
69             f' {request.form["phone"]} '
70             f' {request.form["email"]} '
71             f' {request.form["name"]} '
72         msg.html = f'<h1>Twilio SendGrid Test Email</h1>'
73             f'<p>Congratulations! You have sent a test email with '
74             f' {request.form["recipient"]} '
75             f' {request.form["password"]} '
76             f' {request.form["phone"]} '
77             f' {request.form["email"]} '
78             f' {request.form["name"]} '
79             f' {request.form["password"]} '
79         mail.send(msg)
80         flash(f'A test message was sent to {recipient}.')
81         #return redirect(url_for('index'))
82         return render_template('home.html')
83
84 #end for mail
85
86 @app.route('/addrec', methods = ['POST', 'GET'])
87 def addrec():
88     if request.method == 'POST':
89
90         name = request.form['name']
91         email = request.form['email']
92         phone = request.form['phone']
93         password= request.form['password']
94
95         sql = "SELECT * FROM USERS WHERE NAME =?"
96         stmt = ibm_db.prepare(conn, sql)
```

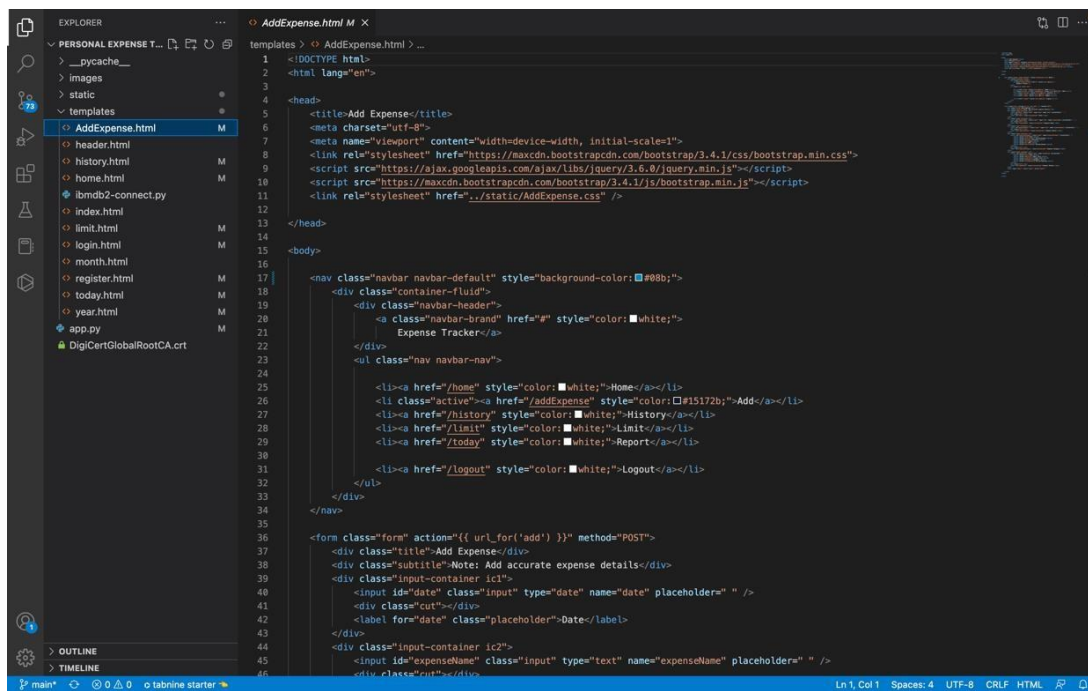
The screenshot shows the VS Code editor with the file explorer on the left displaying a project named 'PERSONAL EXPENSE TRACKER'. The file explorer lists various HTML templates and Python files. The main editor window shows the 'app.py' file, specifically the 'register' function. The function logic is as follows:

```
def register():  
    global user_id  
    msg = ''  
    if request.method == 'POST':  
        email = request.form['email']  
        password = request.form['password']  
        sql = "SELECT * FROM USERS WHERE email = ? AND password = ?"  
        stmt = ibm_db.prepare(conn, sql)  
        ibm_db.bind_param(stmt, 1, email)  
        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['USERID']  
            user_id = account['USERID']  
            session['username'] = account['NAME']  
            #session['name'] = request.form.get("name")  
            #session['username'] = account['Name']  
            msg = 'Welcome*' + session['username'] + '*'!  
            return render_template('home.html', msg = msg)  
        else:  
            msg = 'Incorrect username / password !'  
            return render_template('login.html', msg = msg)  
    else:  
        return render_template('login.html', msg="You are already a member, please login using your details")  
    else:  
        insert_sql = "INSERT INTO USERS (Name,email,phone,password) VALUES (?, ?, ?, ?)"  
        prep_stmt = ibm_db.prepare(conn, insert_sql)  
        ibm_db.bind_param(prepare_stmt, 1, name)  
        ibm_db.bind_param(prepare_stmt, 2, email)  
        ibm_db.bind_param(prepare_stmt, 3, phone)  
        ibm_db.bind_param(prepare_stmt, 4, password)  
        ibm_db.execute(prepare_stmt)  
        return render_template('login.html', msg="Registered successfully..")
```

The screenshot shows the VS Code editor with the file explorer on the left. The main editor window shows the 'app.py' file, specifically the 'add' and 'history' functions. The function logic is as follows:

```
@app.route('/add', methods=['GET', 'POST'])  
def add():  
    global id  
    if request.method == 'POST':  
        date = request.form['date']  
        name = request.form['expenseName']  
        amount = request.form['expenseAmount']  
        category = request.form['expenseCategory']  
        paymentmethod = request.form['payMethod']  
        id = session['id']  
        print(id)  
        insert_sql = "INSERT INTO EXPENSE (USERID,DATE,NAME,AMOUNT,CATEGORY,PAYMENTMETHOD) VALUES (?, ?, ?, ?, ?, ?)"  
        prep_stmt = ibm_db.prepare(conn, insert_sql)  
        ibm_db.bind_param(prepare_stmt, 1, id)  
        ibm_db.bind_param(prepare_stmt, 2, date)  
        ibm_db.bind_param(prepare_stmt, 3, name)  
        ibm_db.bind_param(prepare_stmt, 4, amount)  
        ibm_db.bind_param(prepare_stmt, 5, category)  
        ibm_db.bind_param(prepare_stmt, 6, paymentmethod)  
        ibm_db.execute(prepare_stmt)  
        return render_template('today.html', msg="Data saved successfully..")  
  
@app.route('/history')  
def history():  
    global id  
    id = session['id']  
    print(session['username'])  
    students = []  
    total = 0  
    sql = "SELECT * FROM EXPENSE WHERE USERID=?"  
    stmt = ibm_db.prepare(conn, sql)  
    ibm_db.bind_param(stmt, 1, id)  
    ibm_db.execute(stmt)  
    dictionary = ibm_db.fetch_both(stmt)  
    '''limit="SELECT AMOUNT FROM LIMIT WHERE USERID=?"  
    stmt = ibm_db.prepare(conn, limit)  
    ibm_db.bind_param(stmt, 1, id)  
    ibm_db.execute(stmt)
```

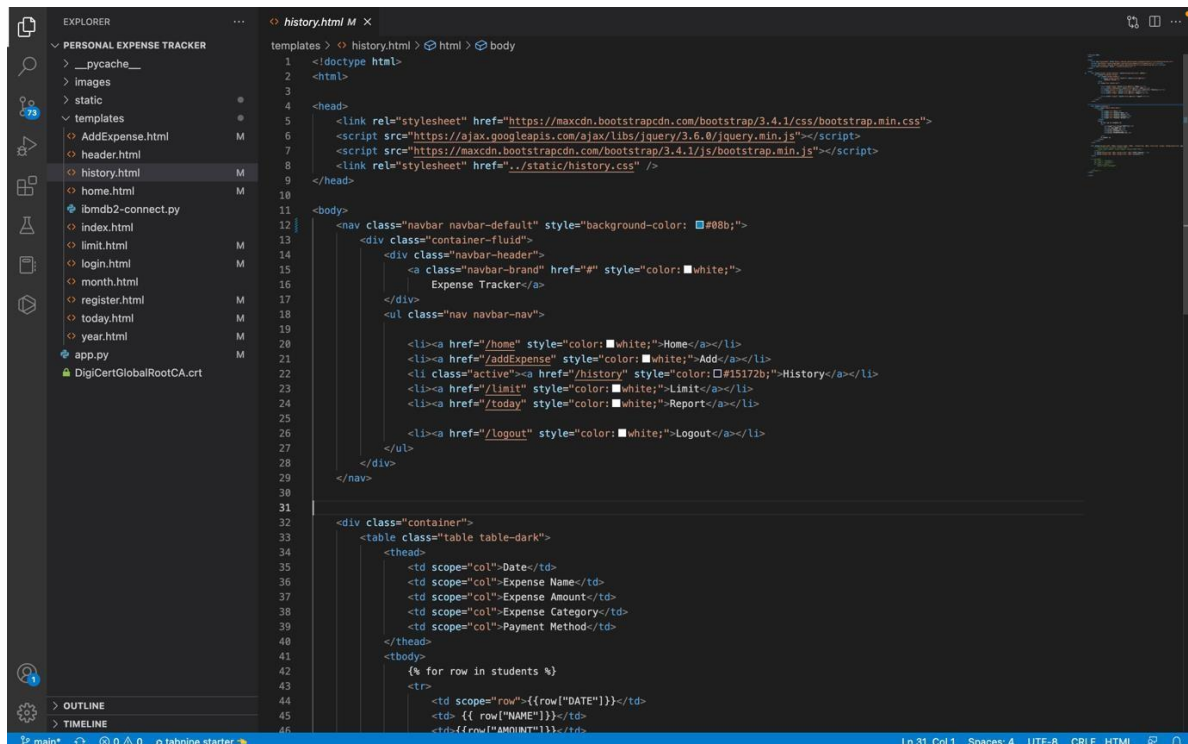
AddExpense.html



The screenshot shows the VS Code editor with the 'AddExpense.html' file open. The Explorer sidebar on the left shows a project structure for 'PERSONAL EXPENSE TRACKER' with files like 'header.html', 'home.html', 'index.html', 'limit.html', 'login.html', 'month.html', 'register.html', 'today.html', 'year.html', and 'app.py'. The main editor displays the HTML code for 'AddExpense.html', which includes a Bootstrap navbar, a form for adding an expense, and links to external CSS and JS files.

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <title>Add Expense</title>
6   <meta charset="utf-8">
7   <meta name="viewport" content="width=device-width, initial-scale=1">
8   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
9   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
10  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
11  <link rel="stylesheet" href="../static/AddExpense.css" />
12 </head>
13
14 <body>
15
16   <nav class="navbar navbar-default" style="background-color: #008b;">
17     <div class="container-fluid">
18       <div class="navbar-header">
19         <a class="navbar-brand" href="#" style="color: white;">Expense Tracker</a>
20       </div>
21       <ul class="nav navbar-nav">
22         <li><a href="/home" style="color: white;">Home</a></li>
23         <li class="active"><a href="/addExpense" style="color: #d1152b;">Add</a></li>
24         <li><a href="/history" style="color: white;">History</a></li>
25         <li><a href="/limit" style="color: white;">Limit</a></li>
26         <li><a href="/today" style="color: white;">Report</a></li>
27
28         <li><a href="/logout" style="color: white;">Logout</a></li>
29       </ul>
30     </div>
31   </nav>
32
33   <form class="form" action="{url_for('add')}" method="POST">
34     <div class="title">Add Expense</div>
35     <div class="sub-title">Note: Add accurate expense details</div>
36     <div class="input-container ic1">
37       <input id="date" class="input" type="date" name="date" placeholder=" " />
38       <div class="cut"></div>
39       <label form="date" class="placeholder">Date</label>
40     </div>
41     <div class="input-container ic2">
42       <input id="expenseName" class="input" type="text" name="expenseName" placeholder=" " />
43       <div class="cut"></div>
44     </div>
45   </form>
46
```

History.html



The screenshot shows the VS Code editor with the 'history.html' file open. The Explorer sidebar on the left shows the same project structure as the previous image. The main editor displays the HTML code for 'history.html', which includes a Bootstrap navbar and a table to display the history of expenses.

```
1 <!doctype html>
2 <html>
3
4 <head>
5   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
6   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
7   <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
8   <link rel="stylesheet" href="../static/history.css" />
9 </head>
10
11 <body>
12
13   <nav class="navbar navbar-default" style="background-color: #008b;">
14     <div class="container-fluid">
15       <div class="navbar-header">
16         <a class="navbar-brand" href="#" style="color: white;">Expense Tracker</a>
17       </div>
18       <ul class="nav navbar-nav">
19         <li><a href="/home" style="color: white;">Home</a></li>
20         <li><a href="/addExpense" style="color: white;">Add</a></li>
21         <li class="active"><a href="/history" style="color: #d1152b;">History</a></li>
22         <li><a href="/limit" style="color: white;">Limit</a></li>
23         <li><a href="/today" style="color: white;">Report</a></li>
24
25         <li><a href="/logout" style="color: white;">Logout</a></li>
26       </ul>
27     </div>
28   </nav>
29
30
31
32   <div class="container">
33     <table class="table table-dark">
34       <thead>
35         <tr>
36           <th>Date</th>
37           <th>Expense Name</th>
38           <th>Expense Amount</th>
39           <th>Expense Category</th>
40           <th>Payment Method</th>
41         </tr>
42       </thead>
43       <tbody>
44         <tr>
45           <td>{{ row["DATE"] }}</td>
46           <td>{{ row["NAME"] }}</td>
47           <td>{{ row["AMOUNT"] }}</td>
48         </tr>
49       </tbody>
50     </table>
51   </div>
52
```


home.html

```
1 <!doctype html>
2 <html>
3
4 <head>
5   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
6   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
7   <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
8   <link rel="stylesheet" href="static/home.css" />
9   <style>
10     @media only screen and (max-width: 1130px) {
11       .image-cus {
12         flex-direction: column;
13         align-items: center;
14       }
15       .title-cus {
16         text-align: center;
17         margin: 0%;
18       }
19       .logo-cus {
20         margin-left: 7%;
21       }
22     }
23   </style>
24 </head>
25
26 <body>
27   <nav class="navbar navbar-default" style="background-color: #008b;">
28     <div class="container-fluid">
29       <div class="navbar-header">
30         <a class="navbar-brand" href="#" style="color: white;">
31           Expense Tracker</a>
32       </div>
33       <ul class="nav navbar-nav">
34
35         <li class="active"><a href="/home" style="color: #008b;">Home</a></li>
36         <li><a href="/addExpense" style="color: white;">Add</a></li>
37         <li><a href="/history" style="color: white;">History</a></li>
38         <li><a href="/limit" style="color: white;">Limit</a></li>
39         <li><a href="/today" style="color: white;">Report</a></li>
40
41         <li><a href="/logout" style="color: white;">Logout</a></li>
42       </ul>
43     </div>
44   </nav>
45
46
```

limit.html

```
1 <!doctype html>
2 <html>
3
4 <head>
5   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
6   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
7   <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
8   <link rel="stylesheet" href="static/history.css" />
9   </head>
10
11 <body>
12   <nav class="navbar navbar-default" style="background-color: #008b;">
13     <div class="container-fluid">
14       <div class="navbar-header">
15         <a class="navbar-brand" href="#" style="color: white;">
16           Expense Tracker</a>
17       </div>
18       <ul class="nav navbar-nav">
19
20         <li><a href="/home" style="color: white;">Home</a></li>
21         <li><a href="/addExpense" style="color: white;">Add</a></li>
22         <li><a href="/history" style="color: white;">History</a></li>
23         <li class="active"><a href="/limit" style="color: #f1572b;">Limit</a></li>
24         <li><a href="/today" style="color: white;">Report</a></li>
25
26         <li><a href="/logout" style="color: white;">Logout</a></li>
27       </ul>
28     </div>
29   </nav>
30   <!-- Currently your MONTHLY limit is { (y)} -->
31   <style>
32     body {
33       font-family: 'Poppins', sans-serif;
34       overflow: hidden;
35     }
36   </style>
37
38   <form action="/limitnum" method="POST">
39
40     <p> ENTER the MONTHLY LIMIT to avoid over EXPENSES</p> <br/>
41
42     <input type="number" name="number" required/> <span>
43     <button class="btn btn-warning" type="submit">ENTER</button>
44
45   </form>
46
```

login.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <title>Expense Tracker - Login</title>
6   <meta charset="utf-8">
7   <meta name="viewport" content="width=device-width, initial-scale=1">
8   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
9   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
10  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
11  <link rel="stylesheet" href="static/login.css" />
12 </head>
13
14 <body>
15   <div>
16     <h4> {{msg}} </h4>
17   </div>
18   <div>
19     <form action="{{ url_for('signIn') }}" method="POST" class="form">
20       <div class="title">Login</div>
21       <div class="input-container ic1">
22         <input id="email" class="input" type="text" name="email" placeholder=" " />
23         <div class="cut"></div>
24         <label for="email" class="placeholder">Email</label>
25       </div>
26       <div class="input-container ic2">
27         <input id="password" class="input" type="password" name="password" placeholder=" " />
28         <div class="cut"></div>
29         <label for="password" class="placeholder">Password</label>
30       </div>
31       <input type="submit" class="submit" value="Login">
32       <div class="register">
33         <p>New User? <a id="loki" href="/register">Register Here</a></p>
34       </div>
35     </form>
36   </div>
37 </body>
38
39 </html>
```

month.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css" integrity="sha384-TX8t2Z...>
6   <script src="https://cdn.jsdelivr.net/npm/chart.js@3.2.1/dist/chart.min.js"></script>
7   <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTPuJ...>
8   <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js" integrity="sha384-9/reFTGAW83EW2RDu2S0VkaI...>
9   <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js" integrity="sha384-B4gt1jrGCTJh4AqTIPSdU0I...>
10  <link rel="stylesheet" href="https://pro.fontawesome.com/releases/v5.10.0/css/all.css" integrity="sha384-AyMEC3Yw5CVb3ZcuHTOA...>
11 </head>
12
13 <body>
14   <nav class="navbar sticky-top navbar-expand-lg navbar-light" style="background-color: #15172b;>
15     <a class="navbar-brand" href="#" style="color: white;">EXPENSE TRACKER</a>
16     <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="
17     <span class="navbar-toggler-icon"></span>
18   </button>
19
20   <div class="collapse navbar-collapse" id="navbarSupportedContent">
21     <ul class="navbar-nav mr-auto">
22       <li class="nav-item">
23         <a class="nav-link" href="/home" style="color: white;">Home <span class="sr-only">(current)</span></a>
24       </li>
25       <li class="nav-item">
26         <a class="nav-link" href="/add" style="color: white;">Add</a>
27       </li>
28       <li class="nav-item">
29         <a class="nav-link" href="/history" style="color: white;">History</a>
30       </li>
31       <li class="nav-item">
32         <a class="nav-link" href="/limit" style="color: white;">Limit</a>
33       </li>
34       <li class="nav-item active dropdown">
35         <a class="nav-link dropdown-toggle" style="color: white;" href="#" id="navbarDropdown" role="button" data-tc
36         Report
37       </a>
38       <div class="dropdown-menu" aria-labelledby="navbarDropdown">
39         <a class="dropdown-item" href="/today">Today</a>
40         <a class="dropdown-item" href="/month">Month</a>
41       </div>
42     </ul>
43   </div>
44 </body>
```



```
49 </div>
50 </li>
51 <li class="nav-item">
52 <a class="nav-link" href="/logout" style="color: white;">Logout</a>
53 </li>
54 </ul>
55 </div>
56 </nav>
57
58 <div class="container">
59
60 <div class="row">
61 <div class="col-md-5">
62 <h3 class="mt-5">MONTH Expense Breakdown</h3>
63
64 <div class="card shadow mb-2 bg-white rounded-pill">
65 <div class="card-body">
66 <div class="row">
67 <div class="col-md-6"> DATE </div>
68 <div class="col-md-6"> AMOUNT </div>
69 </div>
70 </div>
71 </div>
72 <div class="card shadow mb-2 bg-white rounded-bottom">
73 <div class="card-body">
74 <div class="row">
75 <div id="ttline" class="col-md-6">{{row[0]}}</div>
76 <div id="tamount" class="col-md-6"> {{row[1] }} </div>
77 </div>
78 </div>
79 </div>
80 </div>
81 </div>
82 </div>
83 </div>
84 </div>
85 </div>
86 </div>
87 </div>
88 </div>
89 </div>
90 </div>
91 </div>
92 </div>
93 </div>
```

register.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5 <meta charset="UTF-8">
6 <meta http-equiv="X-UA-Compatible" content="IE=edge">
7 <meta name="viewport" content="width=device-width, initial-scale=1.0">
8 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
9 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
10 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
11 <link rel="stylesheet" href="static/AddExpense.css" />
12 <title>Expense Tracker - Register</title>
13 </head>
14
15 <body>
16
17 <form action="{{ url_for('addrec') }}" class="form" method="POST">
18 <div class="title">Register</div>
19 <div class="input-container ic1">
20 <input id="name" class="input" type="text" name="name" placeholder=" " />
21 <div class="cut"></div>
22 <label for="name" class="placeholder">Name</label>
23 </div>
24 <div class="input-container ic2">
25 <input id="email" class="input" type="email" name="email" placeholder=" " />
26 <div class="cut"></div>
27 <label for="email" class="placeholder">Email</label>
28 </div>
29 <div class="input-container ic2">
30 <input id="phone" class="input" type="text" name="phone" placeholder=" " />
31 <div class="cut"></div>
32 <label for="phone" class="placeholder">Phone</label>
33 </div>
34 <div class="input-container ic2">
35 <input id="password" class="input" type="password" name="password" placeholder=" " />
36 <div class="cut"></div>
37 <label for="password" class="placeholder">Password</label>
38 </div>
39 <input type="submit" class="submit" value="Register">
40 <div class="login">
41 <p>Already a user? <a id="loki" href="/login">Login Here</a></p>
42 </div>
43 </form>
44
45
```

today.html

![Screenshot of VS Code editor showing the 'today.html' file in the 'templates' directory. The Explorer sidebar on the left shows a project structure with files like 'AddExpense.html', 'header.html', 'home.html', 'index.html', 'limit.html', 'login.html', 'month.html', 'register.html', 'today.html', and 'year.html'. The main editor area displays the HTML code for 'today.html', which includes a Bootstrap navbar and a dropdown menu. The code is as follows: <pre>1 <!DOCTYPE html>
2 <html lang=](https://code.jquery.com/jquery-3.5.1.slim.min.js)

year.html

![Screenshot of VS Code editor showing the 'year.html' file in the 'templates' directory. The Explorer sidebar on the left shows the same project structure as the first screenshot, with 'year.html' selected. The main editor area displays the HTML code for 'year.html', which is identical to the 'today.html' code shown above. The code is as follows: <pre>1 <!DOCTYPE html>
2 <html lang=](https://code.jquery.com/jquery-3.5.1.slim.min.js)

```

62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
<div class="container">
  <div class="row">
    <div class="col-md-5">
      <h3 class="mt-5">Year Expense Breakdown</h3>
      <div class="card shadow mb-2 bg-white rounded-pill">
        <div class="card-body">
          <div class="row">
            <div class="col-md-6"> YEAR </div>
            <div class="col-md-6"> AMOUNT </div>
          </div>
        </div>
      </div>
    </div>
    <div class="col-md-5">
      <div class="card shadow mb-2 bg-white rounded-bottom">
        <div class="card-body">
          <div class="row">
            <div id="tttime" class="col-md-6">{{row[0]}}</div>
            <div id="tanount" class="col-md-6">{{row[1]}}</div>
          </div>
        </div>
      </div>
    </div>
  </div>
  <div class="row">
    <div class="col-md-5">
      <h3 class="mt-5">Expense Breakdown By Category</h3>
      <div class="card shadow mb-2 bg-white rounded-bottom">
        <div class="card-body">
          <div class="row">
            <div class="col-md-6">Food</div>
            <div id="tfood" class="col-md-6">{{t_food}}</div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>

```

Database Schema

Tables :

1.Admin:

IBM Db2 on Cloud

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

Find schemas or tables Refresh

Tables New table +

Name	Schema	Properties
EXPENSE	DKH61031	...
LIMIT	DKH61031	...
USERS	DKH61031	...

Total: 3, selected: 0

Table definition

USERS No statistics available

Name	Data type	Nullable	Length	Scale
USERID	INTEGER	N		0
NAME	VARCHAR	N	255	0
EMAIL	VARCHAR	N	255	0
PHONE	VARCHAR	N	255	0
PASSWORD	VARCHAR	N	255	0

View data

2.Expense:

The screenshot shows the IBM Db2 on Cloud web interface. The 'Tables' tab is selected, displaying a list of tables: EXPENSE, LIMIT, and USERS, all belonging to the DKH61031 schema. The 'Table definition' panel on the right shows the structure of the 'EXPENSE' table.

Name	Data type	Nullable	Length	Scale
USERID	INTEGER	N		0
DATE	DATE	N	4	0
NAME	VARCHAR	N	255	0
AMOUNT	DECIMAL	N	5	0
CATEGORY	VARCHAR	N	255	0
PAYMENTMETHOD	VARCHAR	N	255	0

At the bottom of the interface, the system tray shows a temperature of 30°C, weather 'Haze', and the time 12:14 on 19-11-2022.

3.LIMIT

The screenshot shows the IBM Db2 on Cloud web interface. The 'Tables' tab is selected, displaying a list of tables: EXPENSE, LIMIT, and USERS, all belonging to the DKH61031 schema. The 'Table definition' panel on the right shows the structure of the 'LIMIT' table.

Name	Data type	Nullable	Length	Scale
USERID	INTEGER	N		0
AMOUNT	DECIMAL	N	5	0

At the bottom of the interface, the system tray shows a temperature of 30°C, weather 'Haze', and the time 12:15 on 19-11-2022.

8. TESTING:

a. TestCases:

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID	Executed By
LoginPage_TC_001	Functional	Home Page	Verify user is able to see the Login/Signup popup when user clicked on life account button	None	1.Go to Homepage	Username: Rahul password: 123	Login/Signup popup should display	Working as expected	Pass				Magaraj S
LoginPage_TC_002	UI	Home Page	Verify the UI elements in Login/Signup popup	Home	1.Go to website 2.Enter details and click login	Username: Rahul password: 123	Application should show below UI elements: a. email text box b. password text box c. Login button with orange colour d. New customer? Create account link	Working as expected	Pass				Ashik Eswar N
LoginPage_TC_003	Functional	Home page	Verify user is able to log into application with Valid credentials	Username and password	1.Go to website 2.Enter details and click login	Username: Rahul password: 123	User should navigate to user account homepage	Working as expected	Pass				Lokeshwar B
LoginPage_TC_004	Functional	Login page	Verify user is able to log into application with Invalid credentials	Username and password	1.Go to website 2.Enter details and click login	Username: Rahul password: 1234	Application should show 'Incorrect email or password' validation message	Working as expected	Pass				Aakash R
LoginPage_TC_005	Functional	Login page	Verify user is able to log into application with Invalid credentials	Login First	1.Go to website 2.Enter details and click login	Username: Rahul password: 1234	Application should show 'Incorrect email or password' validation message	Working as expected	Pass				Magaraj S
ExpensePage_TC_06	Functional	Expense Page	Verify whether user is able to add expense	Add some expenses	1.Go to website 2.Enter details and click login	Food:2000	Application adds expenses	Working as expected	Pass				Lokeshwar B
ExpensePage_TC_07	Functional	Expense Page	Verify whether user is able to add expense	Add some expenses	1.Go to website 2.Enter details and click login	Cloths:500	Application adds expenses	Working as expected	Pass				Aakash R
ExpensePage_TC_08	Functional	Expense Page	Verify whether user is able to add expense	Add some expenses	1.Go to website 2.Enter details and click login	Update Food:2400	Application updates expenses	Working as expected	Pass				Magaraj S
AddLimit_TC_007	Functional	Add Limit	Verify whether user is able to add limit or not	Add Limit	1.Add Limit	Limit:50000	Application adds limits for expenses	Working as expected	Pass				Ashik Eswar N
GenerateReport_TC_008	Functional	Generate report	Verify whether the report is generating or not	Report Generation	1.Generate Report	Report	Application generates reports	Working as expected	Pass				Lokeshwar B

b.User Acceptance Testing

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the Personal Expense Tracker Application project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

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

Logout	2	0	0	2
--------	---	---	---	---

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

Resolution	Severity1	Severity2	Severity3	Severity4	Subtotal
By Design	10	4	2	3	20
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	2	1	8
Totals	22	14	11	22	51

3. TestCaseAnalysis

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

Limit	3	0	0	3
Adding expenses	9	0	0	9
History	4	0	0	4
Final report	2	0	0	2

9.RESULTS

a. Performance Metrics

- i. Tracking income and expenses: Monitoring the income and tracking all expenditures (through bank accounts, mobile wallets, and credit & debit cards).
- ii. Transaction Receipts: Capture and organize your payment receipts to keep track of your expenditure.
- iii. Organizing Taxes: Import your documents to the expense tracking app, and it will streamline your income and expenses under the appropriate tax categories.
- iv. Payments & Invoices: Accept and pay from credit cards, debit cards, net banking, mobile wallets, and bank transfers, and track the status of your invoices and bills in the mobile app itself. Also, the trackingapp sends remindersfor payments an d automatically matches the payments with invoices.
- v. Reports: The expense tracking app generates and sends reports to give a detailed insight about profits, losses, budgets, income, balance sheets, etc.,
- vi. Ecommerce integration: Integrateyour expense trackingapp wit h your eCommerce store and track your sales through payments received via multiple payment methods.
- vii. Vendors and Contractors: Manage and track all the payments to the vendors and contractors added to the mobile app.
- viii. Access control: Increase your team productivity by providing access control to particular users through custom permissions.
- ix. Track Projects: Determine project profitability by tracking labor costs, payroll, expenses, etc., of your ongoing project.
- x. Inventory tracking: An expense tracking app can do it all. Right from tracking products or the cost of goods, sending alert notifications when the product is running out of stock or the product is not selling, to purchase orders.
- xi. In-depth insights and analytics: Provides in-built tools to generate reports with easy-to-understand visuals and graphics to gain insights about the performance of your business.
- xii. Recurrent Expenses: Rely on your budgeting app to track, streamline, and automate all the recurrent expenses and remind you on a timely basis.

10. ADVANTAGES & DISADVANTAGES

1. **Achieve your business goals** with a tailored mobile app that perfectly fits your business.
2. **Scale-up** at the pace your business is growing.
3. Deliver an **outstanding** customer experience through additional control over the app.
4. Control the **security** of your business and customer data
5. Open **direct marketing channels** with no extra costs with methods such as push notifications.
6. **Boost the productivity** of all the processes within the organization.
7. Increase **efficiency** and **customer satisfaction** with an app aligned to their needs.
8. **Seamlessly integrate** with existing infrastructure.
9. Ability to provide **valuable insights**.
10. Optimize sales processes to generate **more revenue** through enhanced data collection.

11. CONCLUSION

From this project, we are able to manage and keep tracking the daily expenses as well as income. While making this project, we gained a lot of experience of working as a team. We discovered various predicted and unpredicted problems and we enjoyed a lot solving them as a team. We adopted things like video tutorials, text tutorials, internet and learning materials to make our project complete.

12. FUTURE

The project assists well to record the income and expenses in general. However, this project has some limitations:

1. The application is unable to maintain the backup of data once it is uninstalled.
2. This application does not provide higher decision capability.

To further enhance the capability of this application, we recommend the following features to be incorporated into the system:

3. Multiple language interface.
4. Provide backup and recovery of data.
5. Provide better user interface for user.
6. Mobile apps advantage.

13. APPENDIX

Source Code Github Link : <https://github.com/IBM-EPBL/IBM-Project-23819-1659931639>

Demo Drive Link: https://drive.google.com/file/d/1xOn0Mb-cuOeBCqx-_AZFygXw2zKu9XOZ/view?usp=share_link

