Personal Expense Tracker Application

IBM-Project-11414-1659326924

PERSONAL EXPENSE TRACKER APPLICATION

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

PROJECT REPORT

ROLE	NAME	REGISTER NO
TEAM LEADER	GIRIDHARAN K	19BEC156
TEAM MEMBER 1	SHYAMKUMAR G	19BEC155
TEAM MEMBER 2	CHOKKALINGAM S R	19BEC158
TEAM MEMBER 3	DURKESHWARAN S	19BEC207

BACHELORS OF ENGINEERING IN ELECTRONICS AND COMMUNICATION ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY COIMBATORE
COIMBATORE - 641 049

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

1.INTRODUCTION

1.1 Project Overview

Category: Cloud App Development

Team ID: PNT2022TMID15113

Skills Required:

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

Project Description:

In simple words, personal finance entails all the financial decisions and ac vi es that a Finance app makes your life easier by helping you to manage your finances efficiently. A personal finance app will not only help you with budge ng and accounting but also give you helpful insights about money management.

Personal finance applications will ask users to add their expenses and based on their expenses wallet balance will be updated which will be visible to the user. Also, users can get an analysis of their expenditure in graphical forms. They have an op on 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.

1.2 Purpose

Personal finance management is an important part of people's lives. However, everyone does not have the knowledge or me to manage their finances in a proper manner. And, even if a person has me and knowledge, they do not bother with tracking their expenses as they find it tedious and me-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 so ware 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

2.1 Existing problem

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

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

- 1. Manual entry and routing of expense reports (62%)
- 2. Lack of visibility into spend data (42%)
- 3. Inability to enforce travel policies (29%)
- 4. Lost expense reports (24%)
- 5. Lengthy expense approval system and reimbursement cycles (23%)

References

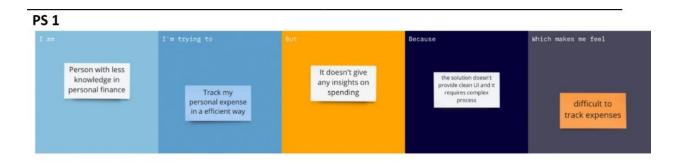
https://www.thebalancemoney.com/is-it-important-to-track-my-expenses-2385679

https://www.icicidirect.com/investonomics/findyourmojo/articles/surprising-benefits-of-tracking-your-expenses Problem Statement Definition:

Personal finance applications will ask users to add their expenses and based on their expenses wallet balance will be updated which will be visible to the user. Also, users can get an analysis of their expenditure in graphical forms. 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.

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 be er understand how they perceive your product or service.



PS₂



3.IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

- 1. Cannot manage their expenditure and savings to keep up with the inflation.
- 2. Identify and eliminate unnecessary expenses and spendings
- 3. To stop living paycheck to paycheck lifestyle

What do they THINK and FEEL?

What do they HEAR?

1.Financial objectives can be achieved easily

2. Easier to manage expenses and better budgetting



What do they SEE?

- 1. Money spent on various sectors
- 2. Dynamic expense management system
- 3. Graph of expense v/s Income

EMPATHY MAP CANVAS

What do they SAY and DO?

- 1. Alerts on unnecessary expenditure
- 2. Daily expense remainder

PAINS

- 1. Rising Inflation
- 2. Sudden Expenses
- 3. Running out of money

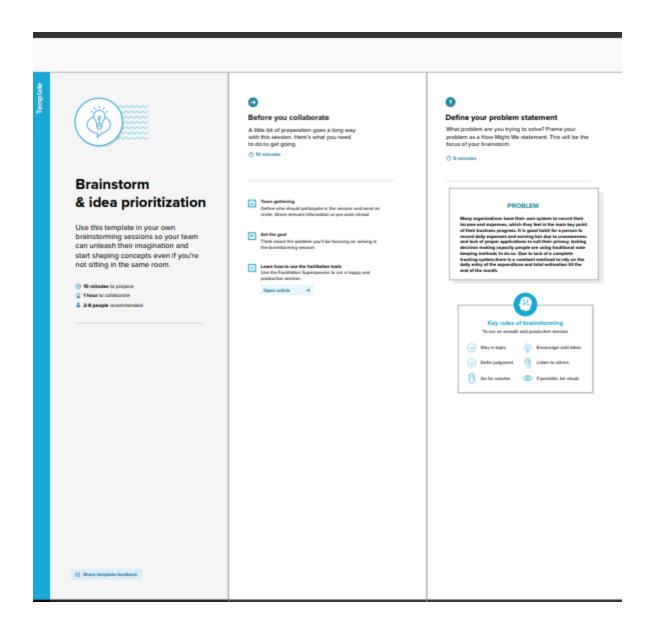
GAINS

- 1. Maintaining a record of all their transactions
- 2. Spending mindfully
- 3. Better Budgetting

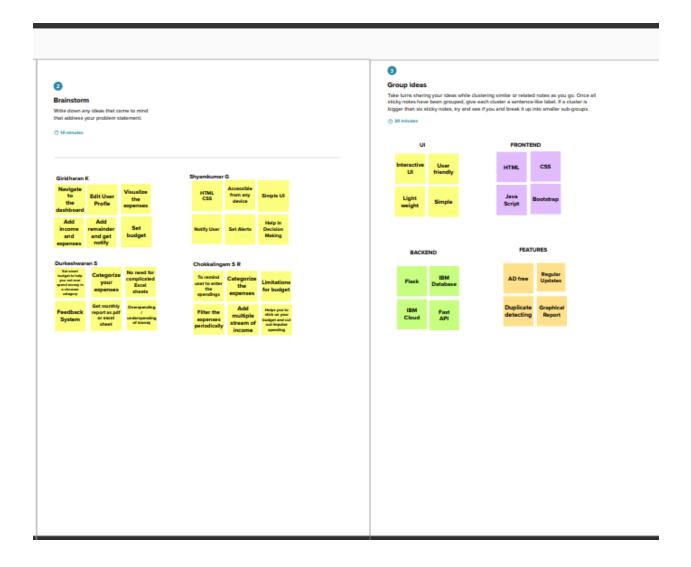


3.2 Ideation & Brainstorming

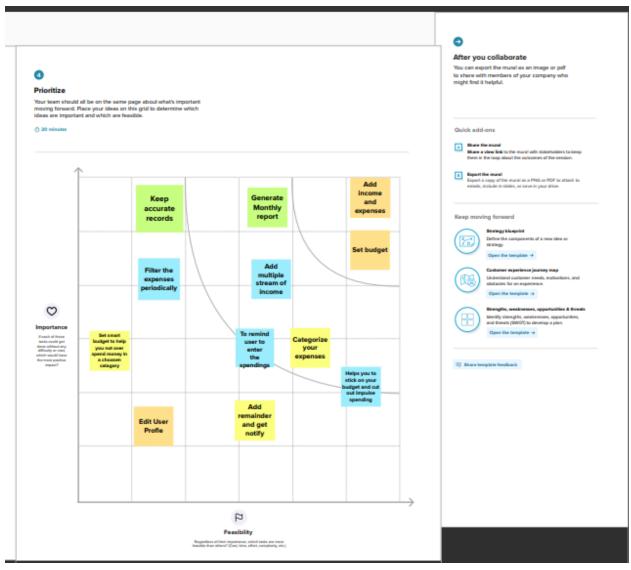
Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping



Step-3: Idea Prioritization



3.3 Proposed Solution

Proposed Solution:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	By tracking expenses and following a plan, a budget makes it easier to pay bills on time, build an emergency fund, and save for major expenses such as a car or home.
2.	Idea / Solution description	This expense tracker is a computerised application which keeps track of all your finances and helps in accounting and budgeting.
3.	Novelty / Uniqueness	The User gets notified once their expense touches 50% 75% 90% & 100% of their limits. Display the costs on a monthly and weekly basis in a pie chart.
4.	Social Impact / Customer Satisfaction	This Application can generate reports of their spendings. It helps the user with their bad spending habits and helps them to save responsibly.
5.	Business Model (Revenue Model)	As this project is intended purely for educational purposes, we keep this application free of cost.
6.	Scalability of the Solution	This Application can handle large numbers of users and data with high performance and security. This application can be used for both large scale and small-scale purposes.

3.4 Problem Solution fit

1. CUSTOMER SEGMENT(S) 6. CUSTOMER CONSTRAINTS CC 5. AVAILABLE SOLUTIONS Explore AS, differentiate Which solutions are available to the customers when they face the problem or need to get job done? Who is your customer? What constraints prevent your 1. People who earn and spend customers from acting or limit money. their choices of solutions? User friendly interface, avoiding misleading ads 2. People who need to track 1. Affordable, stable network their daily expenses. connection 2. Keeping track of user. 2. Authorized Log 9. PROBLEM ROOT CAUSE 7. BEHAVIOUR 2. JOBS-TO-BE-DONE / PROBLEMS What does your customer do to address the problem and get the job done? Which jobs-to-be-done (or What is the real reason that this problem exists? What is problems) do you address for your the back story behind the customers? There could be more than one; explore different sides. need to do this job? 1. User may avoid notification if it is not related. i.e. customers have to do it 1.Remove duplicate transactions. because of the change in 2. User gets frustrated while using 2.Bad user interface. regulations. bad user interface. 3.user data security. 1. No download option user may 3.User may get confused 4.Backup and Restore all not have internet. information. 2. Lack of Financial discipline 8.CHANNELS of BEHAVIOUR 3. TRIGGERS TR 10. YOUR SOLUTION SL СН What triggers customers to act? NLINE If you are working on an existing What kind of actions do customers take i.e. seeing their neighbor business, write down your current online? Extract online channels from #7 installing solar panels, reading solution first, fill in the canvas, and about a more efficient solution check how much it fits reality. 1.User can download reports and can If you are working on a new business share it to other people. 1. Avoiding unwanted expense proposition, then keep it blank until 2.User can have a plan. and bad financial situation. you fill in the canvas and come up with a solution that fits within 2.Guide them and make them customer limitations, solves a problem aware their daily expenses. and matches customer behavior. 4. EMOTIONS: BEFORE / AFTER EM How do customers feel when 1.Generate monthly report and they face a problem or a job forecast budget for the users. and afterwards? 2.Improve users financial i.e. lost, insecure > confident, in management and forecast future control - use it in your budget planning. communication strategy & design. After using the product they are more focused and concentrated about their spending.

4.REQUIREMENT ANALYSIS

4.1 Functional requirement

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through our Website
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Add Expenses	Enter day to day expenditure as input Categorise the expenditure
FR-4	Remainder Mail	Reminds the user once their budget limit crosses 50% 75% 90% 100% of their limit
FR-5	Graph	Creates graph based on the day to day and weekly expenditure
FR-6	Add Salary	User needs to add salary at the start of the month
FR-7	Export CSV	User can export the raw data of their expenses for their own reference in the form of pdf or csv

4.2 Non-Functional requirements

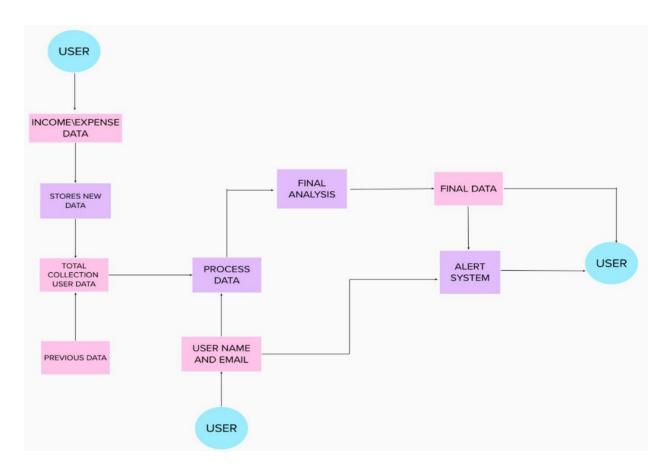
Non-functional Requirements:

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

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	The interface is user friendly and it is easy to use by all type of users.
NFR-2	Security	Every data is secured and encrypted using many encryption algorithms.
NFR-3	Reliability	The transaction must rollback if there is any system failure or network issue. The data is saved when updation of data is failed between the process. Even if there is a failure, the data can be restored within some time.
NFR-4	Performance	The application should not take more than 30 seconds to load. The response is quick even if there is heavy traffic.
NFR-5	Availability	This application is globally available all the time regardless of the traffic.
NFR-6	Scalability	This application is scalable for multiple users as we use docker and kubernetes.

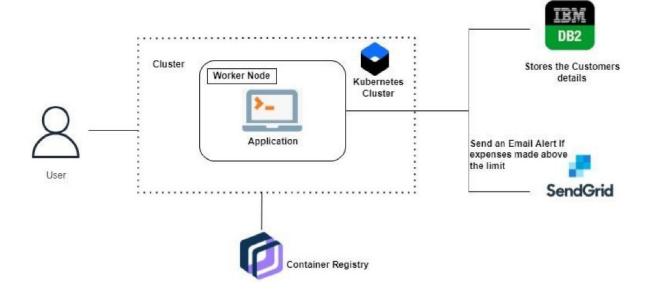
5.PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture

Technical Architecture:



5.3 User Stories

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
	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	
		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	
		USN- 3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	
	Login	USN - 4	As a user, I can log into the application by entering email & password	I can access the application	High	
	Dashboard	USN - 5	As a user I can enter 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 log in issues and other issues of the application.	I can provide support or solution at any time 24*7	Medium	
Administrator	Application	USN - 7	As a administrator I can upgrade or update the application.	I can fix the bug which arises for the customers and users of the application	Medium	

6.PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Team ID	PNT2022TMID15113
Project Name	Personal Expense Tracker Application
Maximum Marks	8 Marks

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

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.	8	High	Giridharan, Shyamkumar
Sprint-1	Login	USN-2	As a user, I can log into the application by entering email & password	8	High	Chokkalingam, Durkeshwaran
Sprint-1	Validating user	USN-3	Checking whether new user or existing user of the application	4	Medium	Giridharan, Shyamkumar
Sprint-2	Add Expense	USN-4	As a user, I can add the day-to-day expense to the application	8	High	Chokkalingam, Durkeshwaran
Sprint-2	Edit and Delete Expense	USN-5	As a user, I can edit and delete the previously created expense	8	High	Giridharan, Shyamkumar

Sprint-2	Creating time- based filters in history.	USN-6	As a user, I can see the time-based history of expenses.	4	Medium	Chokkalingam, Durkeshwaran
Sprint-3	Integrating with pie charts for analysis	USN-7	As a user, I can view diagrammatic representation of expenses	8	High	Giridharan, Shyamkumar
Sprint-3	Enabling limit feature	USN-8	As a user, I can set monthly limit to expenses	4	Medium	Chokkalingam, Durkeshwaran
Sprint-3	Sending Email Alerts	USN-9	As a user, I will receive a mail if I cross the limit	8	High	Giridharan, Shyamkumar
Sprint-4	Testing	USN-9	Testing the application with various tools	10	High	Chokkalingam, Durkeshwaran
Sprint-4	Deployment	USN-9	Deployment of the application	10	High	Giridharan, Shyamkumar

Project Tracker, Velocity & Burndown Chart: (4 Marks)

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	26 Oct 2022	31 Oct 2022	20	31 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022

Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

Velocity

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

$$AV = 20/6 = 3.33$$

6.2 Sprint Delivery Schedule

Project Tracker, Velocity & Burndown Chart: (4 Marks)

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	26 Oct 2022	31 Oct 2022	20	31 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022

Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

7.CODING & SOLUTIONING

7.1 Feature 1

We have added the data visualization methods for expenditure. The pie chart have been used to represent the monthly expenses. The pie chart is apictorial representation of data that makes it possible to visualize therelationships between the parts and the whole of a variable. For example, it is possible to understand the industry count or percentage of a variable levelfrom the

division by areas or sectors. The recommended use for pie charts is twodimensional, as three-dimensional use can be confusing.

The dimensions form sectors of the measurement values; they can have one or two sizes and up to two measures. The first dimension is used to define the angle of each sector that makes up the chart and the second-dimension optionally determines the radius of each sector. Additionally, these plots are useful for comparing data over a fixed period since they do not show changes over me. Therefore, their use should be considered if:

- You are looking to categorize and compare a set of data.
- 1. You only have positive values.
- 2. You have less than seven categories since a larger number canmake it difficult to perceive each segment.

CODE:

```
from flask import Flask, render_template, request, redirect, url_for, session,
flash
import ibm_db
import re
import os
from sendgrid import SendGridAPIClient
from sendgrid.helpers.mail import Mail
# app
app = Flask(__name__)
app.config['SESSION_TYPE'] = 'filesystem'
app.secret_key = '987654321789456123'
# mail
try:
    sg =
SendGridAPIClient('SG.gVahi3nxRaCjJLxP6_q0UA.6h0U46O1j2HQMZhMiDqEs87ayBg2WVJ76mDs
m5YmL9I')
except Exception as e:
   print(e)
#dbconn
def dbconn():
   conn = None
   try:
        conn = ibm db.connect("DATABASE=bludb;HOSTNAME=3883e7e4-18f5-4afe-be8c-
fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;\
PORT=31498;PROTOCOL=TCPIP;UID=xgm40296;PWD=89gtA0GUu8Dxxjxa;SECURITY=SSL;SSLServi
ceCertificate=DigiCertGlobalRootCA.crt", "", "")
    except Exception as e:
        print(e)
   return conn
conn = dbconn()
```

```
@app.route('/')
def welcom():
    return render template('homepage.html')
# signup
@app.route('/signup')
def signup():
    return render_template('signup.html')
@app.route('/signup', methods=['POST','GET'])
def signupp():
    if request.method == 'POST':
        new email = request.form['email']
        new_username = request.form['user']
        new password = request.form['pass']
        repass = request.form['pass'] == request.form['repass']
        if repass == False:
            flash("passwords do not match")
            return render_template('signup.html')
        try:
            curr1 = "INSERT INTO USERS (email, username, pass)
VALUES('{}','{}','{}')".format(request.form["email"], request.form["user"],
request.form["pass"])
            ibm_db.exec_immediate(conn,curr1)
        except Exception as ee:
            print(ee)
            flash("username already exists!!!")
            return render_template("signup.html")
        curr2 = "SELECT * FROM USERS WHERE username =
('{}')".format(new_username,)
        cur2e = ibm db.exec immediate(conn,curr2)
        cur2ex = ibm db.fetch tuple(cur2e)
        books = dict(id=cur2ex[0], email=cur2ex[1], username=cur2ex[2],
password=cur2ex[3] )
        flash("Registration successful. Please login")
        return render_template('login.html'), 201
# login
@app.route('/login')
```

```
def login():
    return render_template('login.html')
@app.route('/login', methods=['POST','GET'])
def loginn():
   if request.method == 'POST':
        cursorr = ibm_db.exec_immediate(conn, "SELECT * FROM USERS where username
 ('{}')".format(request.form['user']))
        currex = ibm_db.fetch_tuple(cursorr)
        print(currex)
        if type(currex) is bool:
            flash("user does not exist")
            return redirect(url_for('signup'))
        if currex[2] == request.form['user']:
            if currex[3] == request.form['pass']:
                session["user"] = request.form['user']
                return redirect(url_for('dash'))
            else:
                flash("password incorrect go back")
                return redirect(url_for("loginn"))
        return render_template("login.html")
# dashboard
@app.route('/dashboard')
def dash():
   if "user" in session:
        usr = session['user']
       ur = ibm_db.exec_immediate(conn, "SELECT user_id FROM USERS where username
 ('{}')".format(usr))
        uid = ibm_db.fetch_tuple(ur)
        for i in uid:
            x = int(i)
        curexp = ibm_db.exec_immediate(conn, "SELECT sum(expense) FROM EXPENSES
where user_id = ('{}')".format(x))
        totexp = ibm_db.fetch_tuple(curexp)
        for j in totexp:
            y = int(j)
        curey = ibm_db.exec_immediate(conn, "SELECT email FROM USERS where
username = ('{}')".format(usr))
```

```
curem = ibm db.fetch tuple(curey)
        for k in curem:
            email = str(k)
        if y > 10000:
            message = Mail(
            from email='admin@123.com',
            to emails=email,
            subject='Your Expenses are getting higher',
            html_content='Your expenses have crossed the monthly budget')
            response = sg.send(message)
            print(response.status code)
            print(response.body)
            print(response.headers)
        curall = ibm_db.exec_immediate(conn, "SELECT expense, expense_detail FROM
EXPENSES where user_id = ('{}')".format(x))
        curem = ibm_db.fetch_assoc(curall)
       items_list = []
       while curem != False:
            items list.append(curem)
            curem = ibm_db.fetch_assoc(curall)
       #items_list = [{'1': 'Hello', '2': 'World'}, {'1': 'World', '2':
'Hello'}]
       return render_template('dashboard.html',totexp = y,columns=['EXPENSE',
EXPENSE_DETAIL'], items=items_list)
   else:
       flash("user not logged in")
       return redirect(url_for("login"))
# add expense
@app.route('/dashboard', methods=['POST','GET'])
def addexpense():
   if request.method == 'POST':
       usr = session['user']
       ur = ibm_db.exec_immediate(conn, "SELECT user_id FROM USERS where username
= ('{}')".format(usr))
       uid = ibm db.fetch tuple(ur)
       for i in uid:
           x = int(i)
```

7.2 Feature 2

Email notifications will be sent to the users once they cross the expenditure limit through send grid mail system. Most notifications are transactional, meaning a recipient's ac on or account activity triggers them. But some notifications are marketing related, encouraging the recipient to take a specific ac on. Ecommerce product notifications inform recipients about new products or discounts. Plus, unlike general marketing emails, these are highly personalized and focus on a single product. For example, if a customer views an item on your website and that item goes on sale, you can send the customer a notification to let them know this is the best me to buy. Users can also opt into receiving notifications when an out-of-stock item is back in stock.

Notification emails tend to perform well because the content is highly relevant to the recipient. But the only way for the recipient to know this is if you state the content clearly in the subject line.

For example, the subject line "New Sign-in to Your Account" gets straight to the point, letting the user know why you sent this notification.

7.3 Database Schema

Tables:

1) USERS

user_id INT NOT NULL GENERATED ALWAYS AS IDENTITY, username VARCHAR(255) NOT NULL, email VARCHAR(255) NOT NULL, password VARCHAR(255) NOT NULL

2)EXPENSES

expenseid INT NOT NULL GENERATED ALWAYS AS IDENTITY, userid INT NOT NULL, expensedetail VARCHAR(255) NOT NULL, expense INT NOT NULL, primary key - expenseid foreign key - user_id

8.1 Test Cases

Test case ID	Feature Type	Componet	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status
LoginPage_TC_001	Functional	Home Page	Verify user is able to see t e Login/Signup popup when user clicked on My account button	None	Go to website Home page appears	Username: test password: 123456	Login/Signup popup should display	Working as expected	Pass
LoginPage_TC_O02	UI	Home Page	Verify the UI elements in Logir/Signup popup	Home	1.Go to website 2.Enter details and click login	Username: test password: 123456	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 e.Last password? Recovery pass link	Working as expected	Pass
LoginPage_TC_O03	Functional	Home page	Verify user is able to log into application with Valid credentials	Username & password	Go to website Enter details and click login	Username: test password: 123456	User should navigate to user account homepage	Working as expected	Pass
LoginPage_TC_004	Functional	Login page	Verify user is able to log into application with InValid credendials	Username & password	1.Go to website 2.Enter details and click login	Username: test password: 123456	Application should show 'Incorret mail or password ' validation message.	Working as expected	Pass
LoginPage_TC_O04	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: test password: 123456	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass
LoginPage_TC_O05	Functional	Login page	Verify user is able to log into application with InValid crede ntials	Login first	1.Go to website 2.Enter details and click login	Username: test password: 123456	Application should show 'incorrect email or password ' validation message.	Working as expected	Pass
AddExpensePag _OO5_TC	Functional	Add Expense page	Verify whether user is able to add expense or not	Have some expense to add	Add date, expense name and other details. Check if the expense gets added	add rent = 6000	Application adds expenses	Working as expected	Pass

8.2 User Acceptance Testing

2. Defect Analysis

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

were resolve	u					
Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subt	otal
By Design	10	5	2	3	20	
Duplicate	1	0	3	0	4	
External	2	3	0	1	6	
Fixed	11	2	4	20	37	
Not Reproduced	0	0	0	0	0	
Skipped	0	0	0	0	0	
Won't Fix	0	5	2	1	8	
Totals 24		14	13	26	75	
Section			Total Cases	Not Tested	Fail	Pass
Print Engine		7	0	0	7	
Client Application		29	0	0	29	

Print Engine	7	0	0	7
Client Application	29	0	0	29
Security	4	0	0	4

			15	
Outsource Shipping	6	0	0	6
Exception Reporting	7	0	0	2
Final Report Output	5	0	0	5
Version Control	1	0	0	1

9.RESULTS

9.1 Performance Metrics

- Tracking income and expenses: Monitoring the income and tracking all expenditures (through bank accounts, mobile wallets, and credit & debit cards).
- 2. Transaction Receipts: Capture and organize your payment receipts to keep track of your expenditure.
- 3. Organizing Taxes: Import your documents to the expense tracking app, and it will streamline your income and expenses under the appropriate tax categories.
- 4. 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 tracking app sends reminders for payments and automatically matches the payments with invoices.
- 5. Reports: The expense tracking app generates and sends reports to give a detailed insight about profits, losses, budgets, income, balance sheets, etc.,
- 6. E-commerce integration: Integrate your expense tracking app with

- your eCommerce store and track your sales through payments received via multiple payment methods.
- 7. Vendors and Contractors: Manage and track all the payments to the vendors and contractors added to the mobile app.
- 8. Access control: Increase your team productivity by providing access control to particular users through custom permissions.
- 9. Track Projects: Determine project profitability by tracking labour costs, payroll, expenses, etc., of your ongoing project.
- 10. 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.
- 11. 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.
- 12. 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

- Achieve your business goals with a tailored mobile app that perfectly fits your business.
- 2. **Scale-up** at the pace your business is growing.
- 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 can 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

13.APPENDIX

Source Code Github Link: https://github.com/IBM-EPBL/IBM-Project-11414-1659326924

Project Demo Video Link:

https://youtu.be/_T67FR1f2F0