PERSONAL EXPENSE TRACKERAPPLICATION

IBM-Project-18816-1659690473

NALAIYA THIRANPROJECT BASED LEARNINGON PROFESSIONAL READLINESS FOR INNOVATION, EMPLOYNMENT AND ENTERPRENEURSHIP

A PROJECT REPORT BY

JEEVAN N M

KARAN B

GODSON SHALOM KING

CHEKKURU THARUN

BACHELOR OF ENGINEERING IN COMPUTER SCIENCE AND ENGINEERING IN VEL TECH HIGH TECH DR RANGARAJAN DR SAKUNTHALA ENGINEERING COLLEGE CHENNAI

INDEX

PROJECT REPORT-PERSONAL EXPENSE TRACKER (TEAM ID:PNT2022TMID22008)

1. INTRODUCTION:

Personal Income Expense Tracker is to easily manage your finance by recording your monthly incomesand expenses. Sometimes at the end of every month, we usually find a shortage of money due to our unaccounted expenses or our bad spending habits. It is necessary to keep track of our incomes and expenses.

a. Project Overview

Category: CloudApp Development

Skills Required:
IBM cloud , HTML, javascript, IBM object storage, python, kubernates docker IBM db2, IBM container registry

Project Description:

Personal Expense Tracker (PET) is a daily expense management system which is specially designed for non- salaried and salaried personnel for keeping track of their daily expenditure with easy and effective way . Personal expense or finance entails all the financial decisions and activities that a Finance app makes your life easier byhelping you to manage your finances efficiently.

A personal finance app will not only help you with budgeting and accounting but also give you helpful insightsabout money management. Personal expense or 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 optionto set a limit for the amountto be used for that particular month if the limit is exceeded the userwill be notifiedwith an email alert.

b. Purpose

The purpose of the project is to help you control your expenses in order to manage the proper spending of money. About the System The Expense Tracker App was created in a HTML web browser that use JavaScript to give user a great interactive experience when using an app.

When you track your spending, you know where your money goes and you can ensure that your money is used wisely. Tracking your expenditures also allows you to understand why you're in debt and how you got there. This will then help you design a befitting strategy of getting out of debt. Budgeting ensures you're not spending more than you're making, allowing you to plan for short- and long-term expenses. It's an easy, helpful way for people with all types of income and expenses to keep their finances in order.

2.LITERATURE SURVEY

Literature reviewwas carried out to gain knowledge and improve the skills needed to complete this project. This chapter shows the different techniques that have been implemented.

2.1 Existing Problem

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 meettheir needs.

2.2 References

S.No	Paper	Author	Year	Methodand Algorithm	Accuracy
1	Expense ManagerApplicati on	A Velmurugan, et al	2020	This paper's main aim to eliminate the use of sticky notes, spreadsheets and handling of large chunksofdata is successful, thenew experience is hassle-free and veryhandy. It usesthe Core Data Model.	94.02%
2	Cloud basedExpense Tracker	Asthha Wahal,et al	2018	The waterfall model is used.This application will help its users to overcome the wastage of money. It will guide them and aware them abouttheir daily expenses	93.4%
3	Expense Tracker : A	Hrithik Gupta, et al	2020	This application willhelp its users to managethe cost of	89.92%

	Smart Approach to Track Everyday Expense			their daily expenditure. It will guide them and aware them about their dailyexpenses. Waterfall model is used for the project because all the requirementsare clear as this project is not dealing with the clients and hence beforehand planning can be madeabout how to carryout each phase of development.	
4	Budget Tracker Highly Customizable Budgeting Mobile Application	Malikberdi Hezretov	2018	The scrum agile software development methodologywas decided to follow for this projectover the likes of waterfall model and incrementalmodel. The understudiesfrom the universitiesspend a great deal dependent on the information and on which needthey are following	86.79%
5	Spending Tracker: A Smart Approach to Track Daily Expense	Uday Pratap Singh, et al	2021	The waterfall modelis used. This application willhelp its users to managethe cost of their daily expenditure. It will guide them and aware them of theirdaily expenses	91.00%

2.3 Problem Statement Definition

It is tough to keep track of all the financial decisions and activities that a person makes. Traditional expense tracking methods are inconvenient and unreliable. In order to get a quick overview about your total incomes and expenses and control spending , its convenient to digitize the process by havinga personal expensetracker.

Who doesthe problem affect?	Investors, savers,big spenders, debtors, consumers on a tight budget, and shoppers.
What are the boundaries of the problem?	Expense tracking software for employees, students, and regular people.
What is the issue	Being watchful of expenses incurred increases financial strain. Making rash financial decisions could decrease financial security and cause you to go over your budget.
When doesthis issue occur?	When employing improper budgeting methods. When you don't keep track of your expenses, you can't determine how much was actually spent.
Where is the issueoccurring	Working peoplewho struggle to keep track of their expenses.
Why is it important that we fix theproblem?	By designating the income for spending, saving, and giving, resolving this problem promotes accountability andencourages financial planning with purpose. This promotes monetary stability.

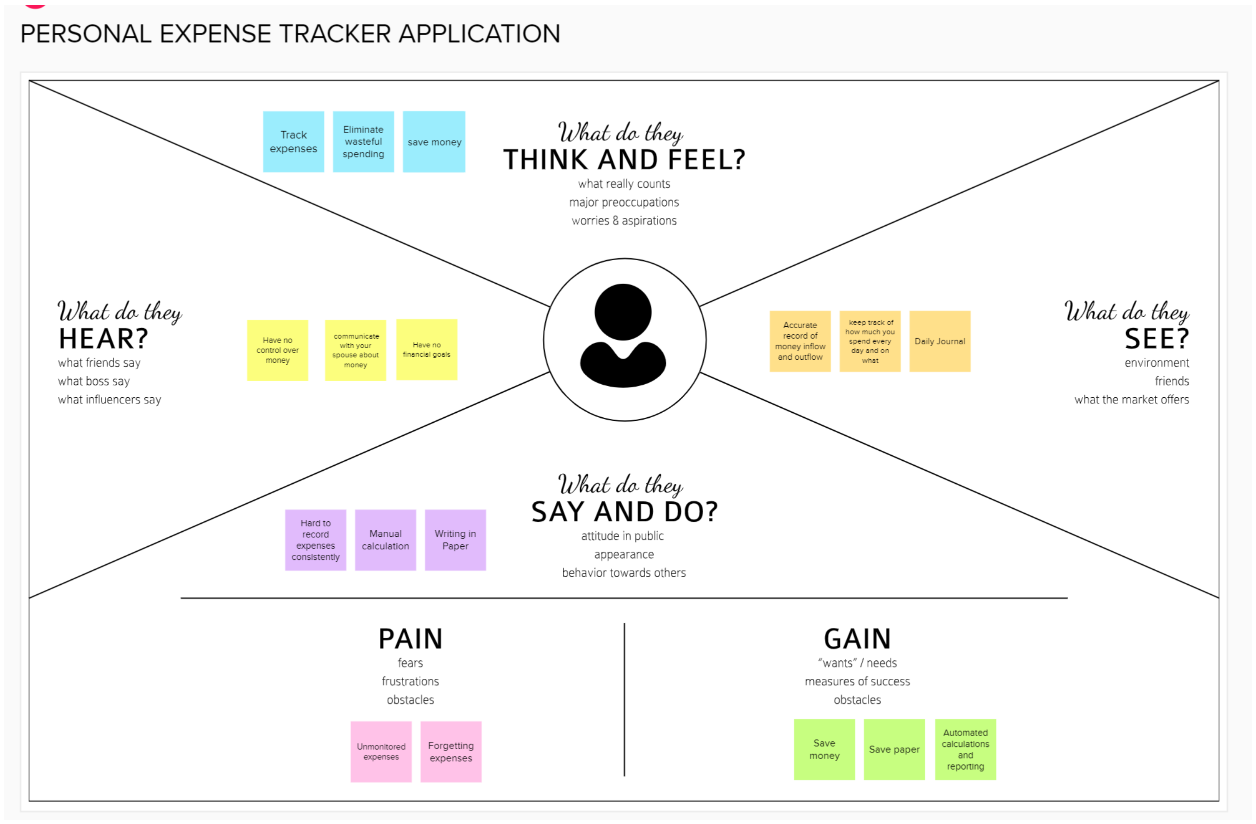
Kevin, who is interested in stock investing, finds it challenging to estimate the cost of stockinvesting. He can easily and effectively plan out his expenses for investing with the aid of expense tracking.

Raj, a novice budgeter, finds it difficult to keep track of and manage his expenses in the midst of his hectic schedule. Setting priorities for his expenses will enable him to reduceirrational spending.

High school student, Ariyan typically receives a meagre allowance from his parents. So hecan spend on both his regular expenses and himself by keeping track of his spending and using good budgeting techniques.

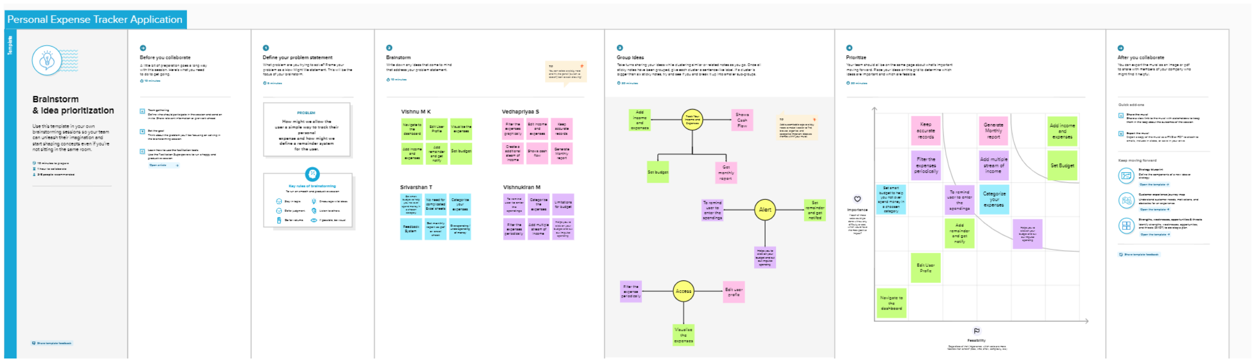
3 .IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

3.1.a Brainstorm



PROBLEM STATEMENT

Many organizations have their own system to record their income and expenses,which they feel is the main key point of their business progress. It is good habit for a personto record daily expenses and earning but due to unawareness and lack of properapplicationsto suit their privacy, lacking decision making capacity people are using traditional note keeping methods to do so. Due to lack of a complete tracking system, there is a 2 constant overload to rely on the daily entry of the expenditure and total estimation till the end of the month.

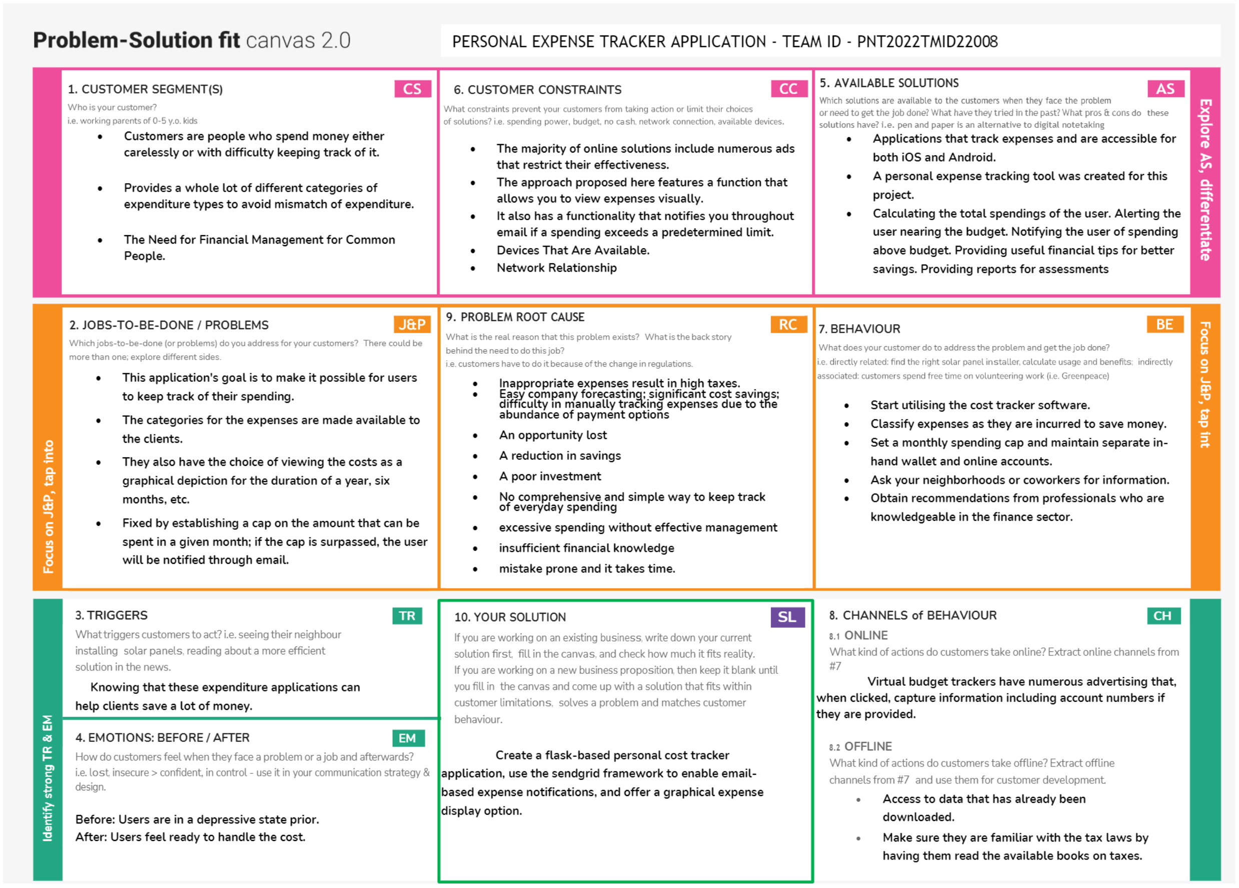
Who doesthe problem affect?	Investors, savers,big spenders, debtors, consumers on a tight budget, and shoppers.
What are the boundaries of the problem?	Expense tracking software for employees, students, and regular people.
What is the issue	Being watchful of expenses incurred increases financial strain. Making rash financial decisions could decrease financial security and cause you to go over your budget.
When doesthis issue occur?	When employing improper budgeting methods. When you don't keep track of your expenses, you can't determine how much was actually spent.
Where is the issueoccurring	Working peoplewho struggle to keep track of their expenses.
Why is it important that we fix theproblem?	By designating the income for spending, saving, and giving, resolving this problem promotes accountability andencourages financial planning with purpose. This promotes monetary stability.

3.3 Proposed Solution

S.NO.	Parameter	Description
1.	Problem Statement	In paper-based expense tracker systemit is difficult to track our monthly expenses manually. In paper-based expensetracker system it is difficult to track our monthly expenses manually. The paper-based expense records may get lost in case of fire accidents, flood etc.
2.	Scalability of theSolution	This application can handle large number of usersand data withhigh performance and security. This application can adapt for both large-scale and small-scale purposes. Easily available in all kindsofdevices.
3.	Idea / Solution description	Daily expensemanagement system whichis specially designed for non-salaried and salaried personnel for keeping trackof their dailyexpenditure with easy and effective way through computerized system whichtends to eliminate manual paper works. Personal finance applications will ask users to add their expenses and based on their expenses wallet balance will be updatedwhich will be visible to the user. 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 emailalert.

4.	Novelty / Uniqueness	The user gets notified when their expense exceedsthe limit and also it remindsthe user when they forgot to make entry. Tracking expenses through SMS. Data analytics on expenses. Future expense prediction
5.	Social Impact / Customer Satisfaction	The application should be able to generate reports of their spending and notify users if they have exceeded their budget. It is designed to be dynamic to produce the prediction. It also provides users’ personal information, their income as well as theirexpenses. This application can create awareness among common people about finance and stuffs. Thisapplication also helpsuser to be financially responsible. It Reduces time rather than entering details manually.
6.	Business Model (Revenue Model)	This Application is provided for free of cost. ButIt will have some advertisement. In premium version there is no advertisement and contains some additional features.

3.4 Problem Solution fit



Problem-Solution fit canvas 2.0

PERSONAL EXPENSE TRACKER APPLICATION - TEAM ID - PNT2022TMID22008

1. CUSTOMER SEGMENT(S) CS

Who is your customer?
i.e. working parents of 0-5 y.o. kids

- Customers are people who spend money either carelessly or with difficulty keeping track of it.
- Provides a whole lot of different categories of expenditure types to avoid mismatch of expenditure.
- The Need for Financial Management for Common People.

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

- The majority of online solutions include numerous ads that restrict their effectiveness.
- The approach proposed here features a function that allows you to view expenses visually.
- It also has a functionality that notifies you throughout email if a spending exceeds a predetermined limit.
- Devices That Are Available.
- Network Relationship

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

- Applications that track expenses and are accessible for both iOS and Android.
- A personal expense tracking tool was created for this project.
- Calculating the total spendings of the user. Alerting the user nearing the budget. Notifying the user of spending above budget. Providing useful financial tips for better savings. Providing reports for assessments

Explore AS, differentiate

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

- This application's goal is to make it possible for users to keep track of their spending.
- The categories for the expenses are made available to the clients.
- They also have the choice of viewing the costs as a graphical depiction for the duration of a year, six months, etc.
- Fixed by establishing a cap on the amount that can be spent in a given month; if the cap is surpassed, the user will be notified through email.

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists? What is the back story behind the need to do this job?
i.e. customers have to do it because of the change in regulations.

- Inappropriate expenses result in high taxes.
- Easy company forecasting; significant cost savings; difficulty in manually tracking expenses due to the abundance of payment options
- An opportunity lost
- A reduction in savings
- A poor investment
- No comprehensive and simple way to keep track of everyday spending
- excessive spending without effective management
- insufficient financial knowledge
- mistake prone and it takes time.

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

- Start utilising the cost tracker software.
- Classify expenses as they are incurred to save money.
- Set a monthly spending cap and maintain separate in-hand wallet and online accounts.
- Ask your neighborhoods or coworkers for information.
- Obtain recommendations from professionals who are knowledgeable in the finance sector.

Focus on J&P, tap into

Focus on J&P, tap int

3. TRIGGERS TR

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

Knowing that these expenditure applications can help clients save a lot of money.

4. EMOTIONS: BEFORE / AFTER EM

How do customers feel when they face a problem or a job and afterwards? i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

Before: Users are in a depressive state prior.
After: Users feel ready to handle the cost.

10. YOUR SOLUTION SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

Create a flask-based personal cost tracker application, use the sendgrid framework to enable email-based expense notifications, and offer a graphical expense display option.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE

What kind of actions do customers take online? Extract online channels from #7

Virtual budget trackers have numerous advertising that, when clicked, capture information including account numbers if they are provided.

8.2 OFFLINE

What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

- Access to data that has already been downloaded.
- Make sure they are familiar with the tax laws by having them read the available books on taxes.

Identify strong TR & EM

4.FUNCTIONAL REQUIREMENT

4.1 Functional Requirements

FR No.	**Functional Requirement(Epic)**	**Sub Requirement (Story / Sub-Task)**
FR-1	User Registration	Registration through Application Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User monthly expense tentativedata	Data to be registered in the app
FR-4	User monthly income data	Data to be registered in the app
FR-5	Alert / Notification	Alert through E-mail Alert through SMS
FR-6	User Budget Plan	Planning and Tracking of user expenses vs budget limit

4.2 Non Functional Requirements

FR No.	**Non-Functional Requirement**	**Description**
NFR-1	**Usability**	Effectiveness,efficiency, and overallsatisfaction of the user while interacting with the application
NFR-2	**Security**	Authentication, authorization, andencryption of the application
NFR-3	**Reliability**	Probability of failure-free operations in a specified environment for a specified time

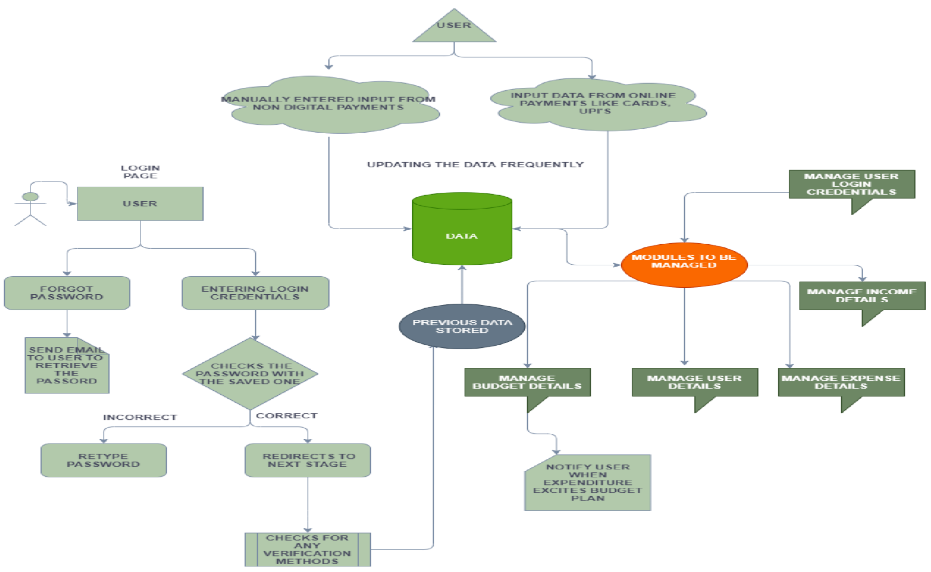
NFR-4	**Performance**	How the application is functioning andhow responsivethe application is tothe end-users
NFR-5	**Availability**	Without near 100% availability, applicationreliability and the usersatisfaction will affectthe solution
NFR-6	**Scalability**	The capacity of the application to handle growth, especiallyin handling moreusers.

5.PRODUCTDESIGN

a. Data Flow Diagrams

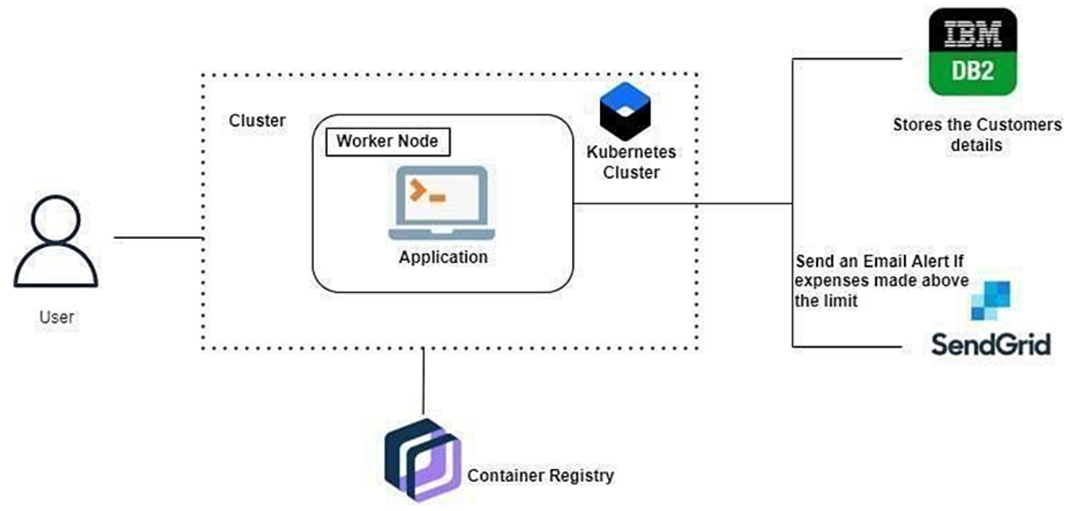
A Data Flow Diagram(DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount ofthe system requirement graphically. It showshow data entersand leaves the system, what changes the information, and where data is stored.

DATA FLOW DIAGRAM:

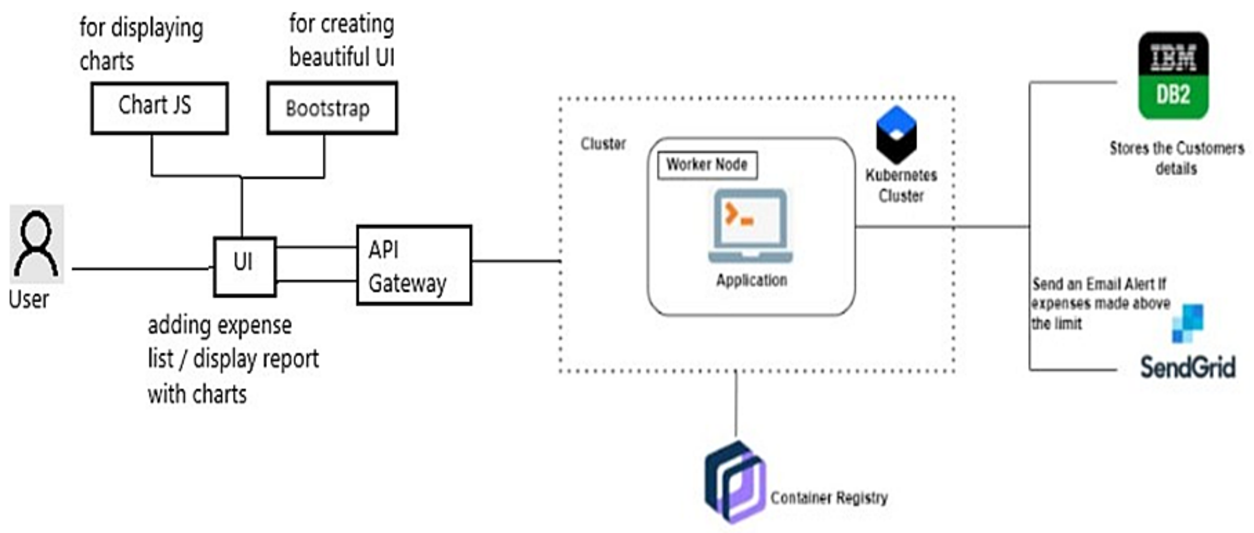

USER
MANUALLY ENTERED INPUT FROM NON DIGITAL PAYMENTS
INPUT DATA FROM ONLINE PAYMENTS LIKE CARDS, UPI'S
UPDATING THE DATA FREQUENTLY
LOGIN PAGE
USER
DATA
MANAGE USER LOGIN CREDENTIALS
MODULES TO BE MANAGED
FORGOT PASSWORD
ENTERING LOGIN CREDENTIALS
MANAGE INCOME DETAILS
PREVIOUS DATA STORED
SEND EMAIL TO USER TO RETRIEVE THE PASSORD
CHECKS THE PASSWORD WITH THE SAVED ONE
MANAGE BUDGET DETAILS
MANAGE USER DETAILS
MANAGE EXPENSE DETAILS
INCORRECT
CORRECT
RETYPE PASSWORD
REDIRECTS TO NEXT STAGE
NOTIFY USER WHEN EXPENDITURE EXCITES BUDGET PLAN
CHECKS FOR ANY VERIFICATION METHODS

b. TechnicalArchitecture

The Deliverable shall include the architectural diagramas below and the information as per the table1 & table 2.



Solution Architecture


for displaying charts
for creating beautiful UI
Chart JS
Bootstrap
User
UI
API Gateway
adding expense list / display report with charts
Cluster
Worker Node
Application
Kubernetes Cluster
IBM DB2
Stores the Customers details
Send an Email Alert If expenses made above the limit
SendGrid
Container Registry

C. User Stories

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

User Type	Functional Requirement (Epic)	User StoryNumber	User Story / Task	Acceptancecriteria	Priority	Release
Customer (Mobileuser & web 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	
		USN-2	As a user,I will receive confirmation email once I have	I can receive confirmation email & click confirm	High	

			registered for the application			
		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 userI 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 solvethe 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

Use the below template to create productbacklog and sprintschedule.

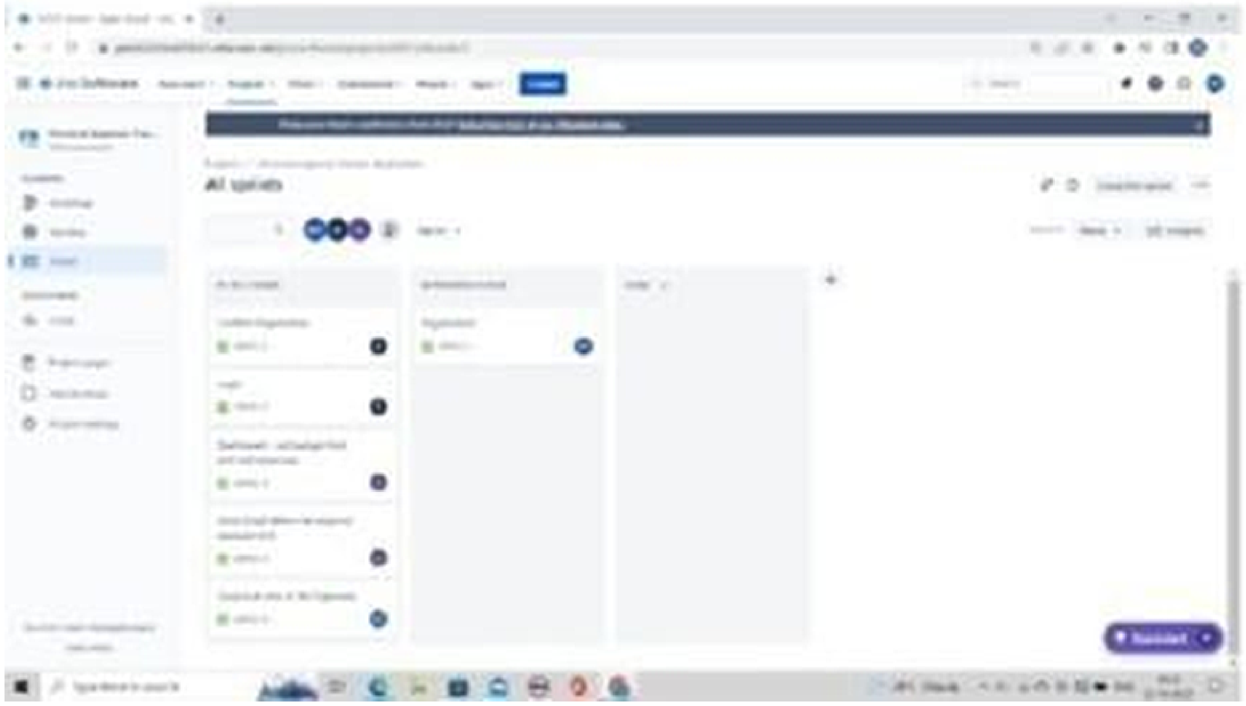
Sprint	Functional Requirement (Epic)	User Story Number	User Story/ Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user,I can registerfor the application by entering my email, password, age and confirm the password.	8	High	JEEVAN, KARAN, GODSON, THARUN
Sprint-1	Login	USN-2	As a user,I can log into the application by entering myemail and password	8	High	JEEVAN, KARAN, GODSON, THARUN
Sprint-2	Set limit and update	USN-3	As a user, I can be able to set theamount limit and able to update	5	Medium	JEEVAN, KARAN, GODSON, THARUN

Sprint-3	Adding expenses	USN-4	As a user, I can add the expenses of spending moneywith many information	5	Medium	JEEVAN, KARAN, GODSON, THARUN
Sprint-4	Producing output.Changethe limit.	USN-5	As a user, I can see an old expense and if I need to set a monthly limit, we can set it onthe application.	5	Medium	JEEVAN, KARAN, GODSON, THARUN

6.2 Sprint Delivery Schedule

Sprint	**Total Story Points**	**Duration**	**Sprint Start Date**	**Sprint End Date(Planned)**	**Story Points Completed (as onPlanned End Date)**	**Sprint Release Date(Actual)**
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 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

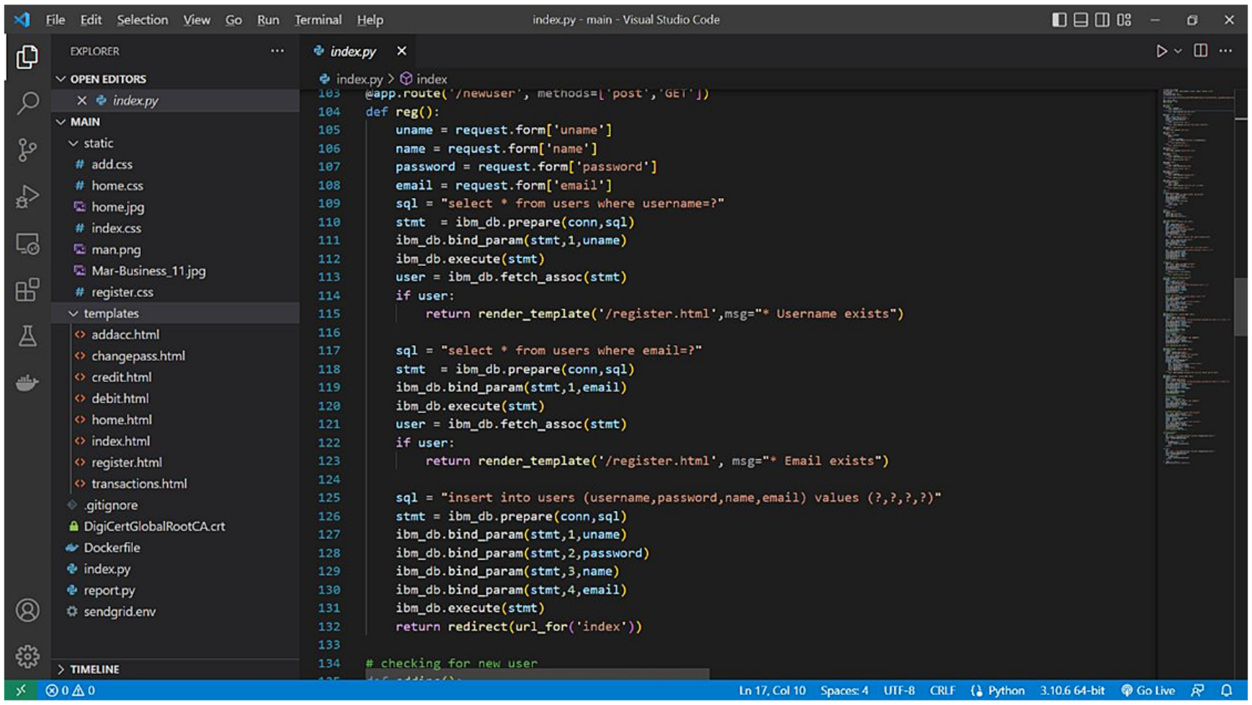
6.3 Reports from JIRA



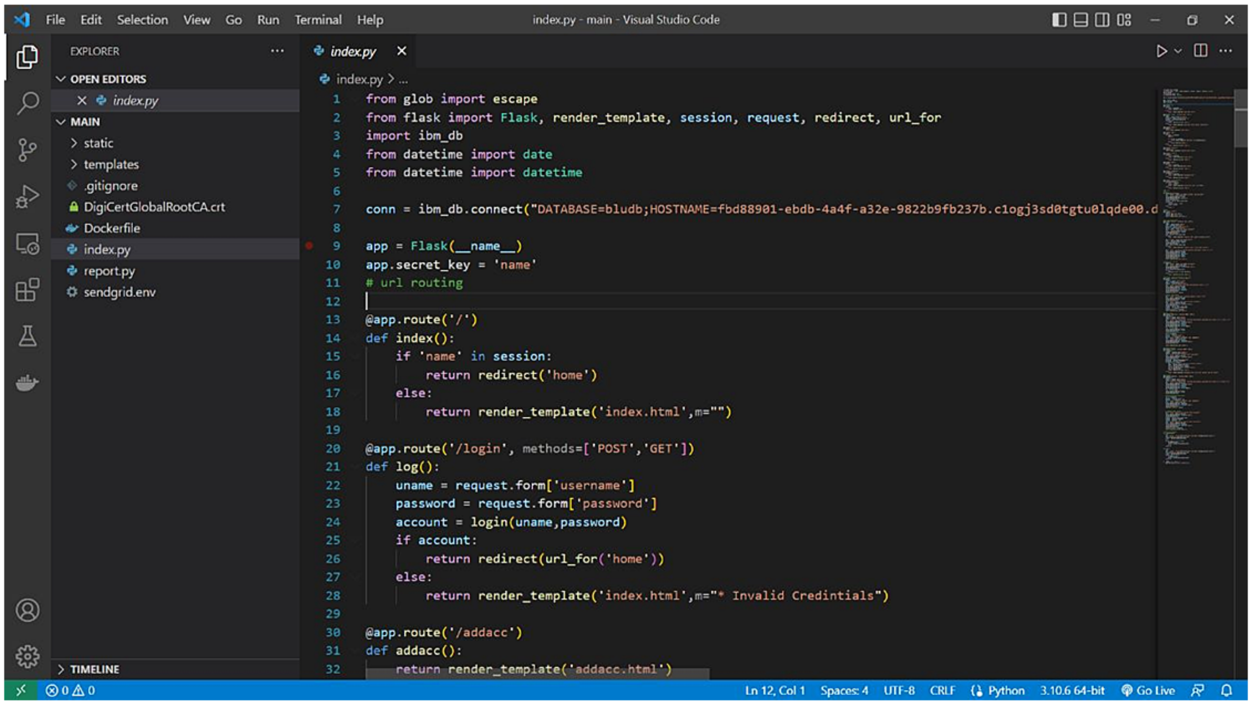
CODING & SOLUTION

a. **Feature 1**

   i. Expense and revenue tracking.
   ii. Managing transaction receipts and records.
   iii. Paying taxes in time.
   iv. Processing payment and invoices.
   v. Create in-depth reports.66

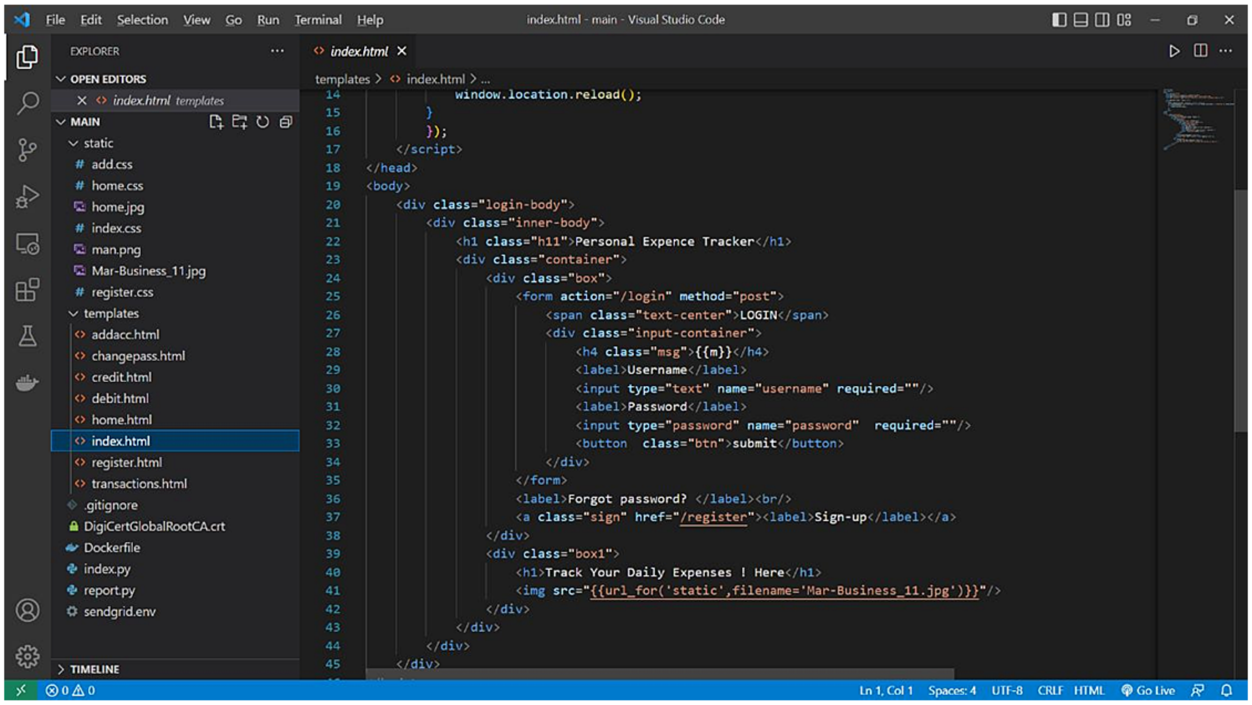


```python
@app.route('/newuser', methods=['post','GET'])
def reg():
    uname = request.form['uname']
    name = request.form['name']
    password = request.form['password']
    email = request.form['email']
    sql = "select * from users where username=?"
    stmt  = ibm_db.prepare(conn,sql)
    ibm_db.bind_param(stmt,1,uname)
    ibm_db.execute(stmt)
    user = ibm_db.fetch_assoc(stmt)
    if user:
        return render_template('/register.html',msg="* Username exists")

    sql = "select * from users where email=?"
    stmt  = ibm_db.prepare(conn,sql)
    ibm_db.bind_param(stmt,1,email)
    ibm_db.execute(stmt)
    user = ibm_db.fetch_assoc(stmt)
    if user:
        return render_template('/register.html', msg="* Email exists")

    sql = "insert into users (username,password,name,email) values (?,?,?,?)"
    stmt = ibm_db.prepare(conn,sql)
    ibm_db.bind_param(stmt,1,uname)
    ibm_db.bind_param(stmt,2,password)
    ibm_db.bind_param(stmt,3,name)
    ibm_db.bind_param(stmt,4,email)
    ibm_db.execute(stmt)
    return redirect(url_for('index'))

# checking for new user
```



```python
from glob import escape
from flask import Flask, render_template, session, request, redirect, url_for
import ibm_db
from datetime import date
from datetime import datetime

conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.d

app = Flask(__name__)
app.secret_key = 'name'
# url routing

@app.route('/')
def index():
    if 'name' in session:
        return redirect('home')
    else:
        return render_template('index.html',m="")

@app.route('/login', methods=['POST','GET'])
def log():
    uname = request.form['username']
    password = request.form['password']
    account = login(uname,password)
    if account:
        return redirect(url_for('home'))
    else:
        return render_template('index.html',m="* Invalid Credintials")

@app.route('/addacc')
def addacc():
    return render_template('addacc.html')
```

b. **Feature 2**



```html
            window.location.reload();
        }
        });
    </script>
</head>
<body>
    <div class="login-body">
        <div class="inner-body">
            <h1 class="h11">Personal Expence Tracker</h1>
            <div class="container">
                <div class="box">
                    <form action="/login" method="post">
                        <span class="text-center">LOGIN</span>
                        <div class="input-container">
                            <h4 class="msg">{{m}}</h4>
                            <label>Username</label>
                            <input type="text" name="username" required=""/>
                            <label>Password</label>
                            <input type="password" name="password"  required=""/>
                            <button  class="btn">submit</button>
                        </div>
                    </form>
                    <label>Forgot password? </label><br/>
                    <a class="sign" href="/register"><label>Sign-up</label></a>
                </div>
                <div class="box1">
                    <h1>Track Your Daily Expenses ! Here</h1>
                    <img src="{{url_for('static',filename='Mar-Business_11.jpg')}}"/>
                </div>
            </div>
        </div>
    </div>
```

7.3 Database Schema

Tables:

1)Admin

id INT NOT NULL GENERATED
ALWAYS AS IDENTITY,username
VARCHAR(32) NOT NULL,email
VARCHAR(32) NOT
NULL,password VARCHAR(32)
NOT NULL

2)Expense

id INT NOT NULL GENERATED
ALWAYS AS IDENTITY,userid INT
NOT NULL, date TIMESTAMP(12)
NOT NULL,expensename
VARCHAR(32) NOT NULL,
amountVARCHAR(32) NOT NULL
paymode VARCHAR(32) NOT
NULL,
category VARCHAR(32) NOT NULL

3)Limit

id INT NOT NULL
GENERATED
ALWAYS AS
IDENTITY,userid
VARCHAR(32) NOT NULL,
limit
VARCHAR(32) NOT NULL

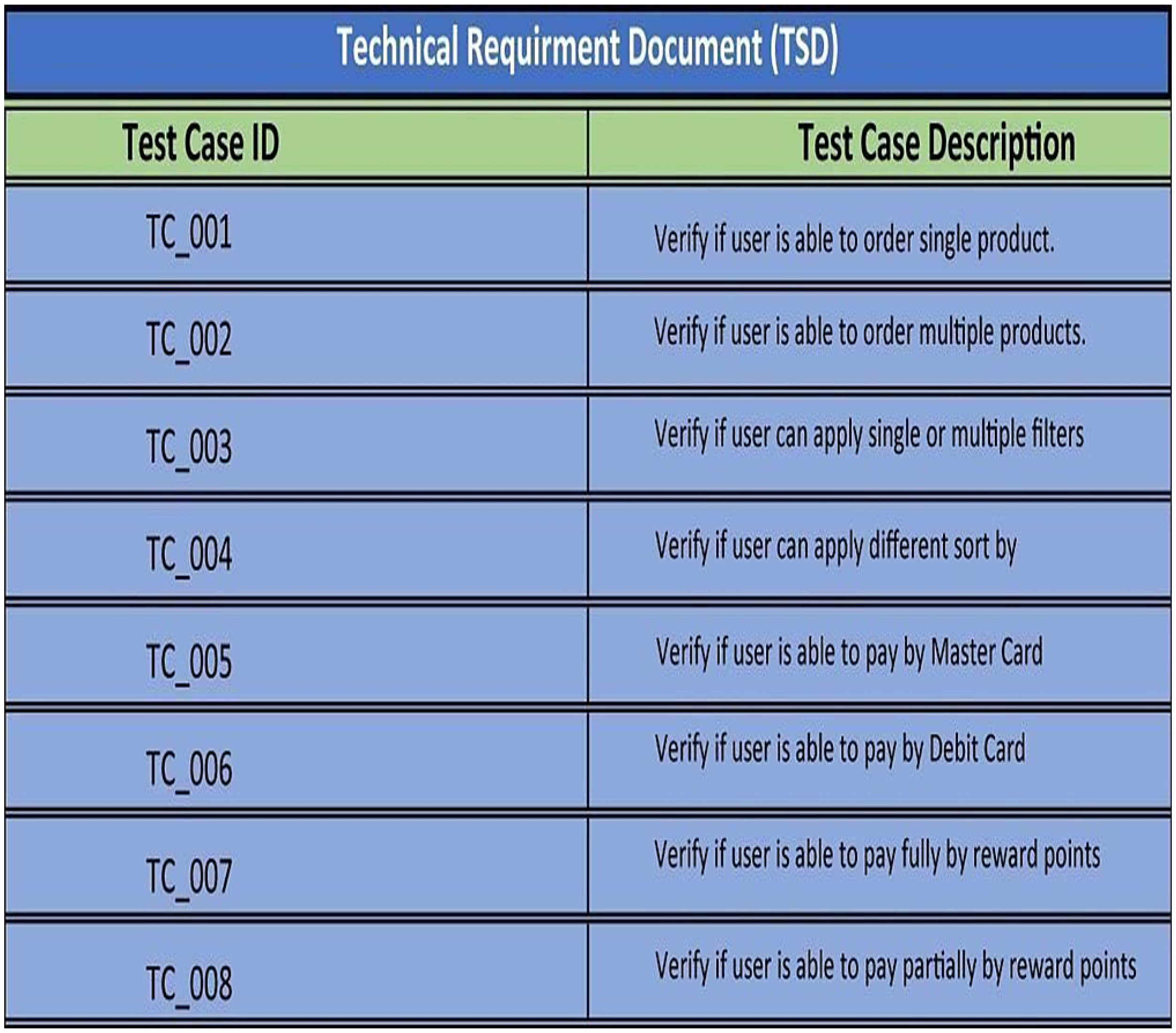
8.TESTING

8.1 Test Cases

Test Case ID	Purpose	TestCases	Result
TC1	Authentication	Password with length less than 4 characters	Password cannot be less than 4 characters
TC2	Authentication	User name with length lessthan 2 characters	User name cannot be less than 2 characters
TC3	Authentication	Valid user name with minimum 2	User name accepted

		characters	
			cannot be less
		Password field	Password cannotbe
		valid password	Password accepted
		Password and Confirm Password did notmatch	Please enter password
		Confirm Password field	Please enter password

8.2 UserAcceptance Testing



Technical Requirment Document (TSD)

Test Case ID	Test Case Description
TC_001	Verify if user is able to order single product.
TC_002	Verify if user is able to order multiple products.
TC_003	Verify if user can apply single or multiple filters
TC_004	Verify if user can apply different sort by
TC_005	Verify if user is able to pay by Master Card
TC_006	Verify if user is able to pay by Debit Card
TC_007	Verify if user is able to pay fully by reward points
TC_008	Verify if user is able to pay partially by reward points

9.RESULTS

9.1 Performance Metrics

a. Tracking income and expenses: Monitoring the income and tracking all expenditures (through bank accounts, mobilewallets, and credit& debit cards).

b. Transaction Receipts: Capture and organize your payment receipts to keep track of your expenditure.

c. Organizing Taxes: Import your documents to the expensetracking app, and it will streamline your income and expenses under the appropriate tax categories.

d. 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 matchesthe paymentswith invoices.

e. Reports: The expense tracking app generates and sends reports to give a detailed insight about profits, losses,budgets, income, balance sheets, etc.

f. E-commerce integration: Integrate your expense trackingapp with your eCommerce storeand track your sales throughpayments received via multiple paymentmethods.

g. Vendorsand Contractors: Manage and track all the payments to the vendorsand contractors added to the mobile app

h. Access control: Increase your team productivity by providing access control to particular users through custompermissions
i. Track Projects: Determine project profitability by tracking labor costs, payroll, expenses, etc., of your ongoing project
j. 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 purchaseorders
k. In-depth insights and analytics: Provides in-built tools to generate reports with easy-to- understand visuals and graphics to gain insightsabout the performance of your business.
l. 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

10.1 Advantages

1. **Maintaining Financial Control**

When if comes to personal finance, being out of control is not something anybodywould strive for. There's nothingfinancially worse than feeling like you don't have any idea what'sgoing on with your money.

The good news is, when you make an effort to record every financial transaction you make, you are essentially, taking the reinson anything and everything involving your money. At anyone time, you will know exactly how much money is sitting in your bank account,and how much you can spend. In other words, when you track your expenses, you take completecontrol over your finances.

Holding YourselfAccountable

If you haveany plans on saving, investing, getting out of debt, or building wealth, what is holding you accountable. I mean, we can all set financialgoals, and have financial dreams,but if you aren't trackingyour expenses, there is nothingto hold you accountable when youmake a bad financial decision.1. Susceptible to costly human errors

Didyou know that up to 9 out of 10 spreadsheets consistof human errors?

Unfortunately, even thesmallest of mistakesin a spreadsheet can cause catastrophic consequences. Fidelity MagellanFund once suffereda $2.6 billion overstatement when an accountant accidentally omitted the minus sign on a net capitalloss of $1.3 billion.

There is always a greater chanceof human error with manualprocesses, especially when itcomes to complex data sets, such as those involved with expense management. Failure to accurately track you company's expenditure and pay invoices on time can wreak havoc on your business'sbottom line.

2. **Lack of collaboration and access**

Because Excel spreadsheets are a single file, only one user at a time may access and modify the data. It can also be challenging to collaborate with other departments because you have to manually shareor email a copy of the relevantspreadsheet with yourcolleagues.

When it comes to expense management data, however, these Excel spreadsheets are frequently shared and proofed acrossnumerous teams and departments. To guarantee thateveryone is viewing the current version, users must be rigorous about version control and sharing when updates are made.

3. **Time-consuming manual processes**

The quantityof expense management data you need to review,analyse, and trackwill grow as your business evolves. The only way tovalidate your data when using Excel spreadsheets, however,is to manually double check andre-enter any inaccurate information. This is a time-consuming and labour-intensive task.
As a result, Excel spreadsheets slow workers down and reduce accuracy by requiring themto perform repetitive processes that could be simplified or automated using expense management and invoicing software.

4. **Inaccuracy leads to slower decisionmaking**

There’s no denying that manual processeswhich increase the chances of inaccuracy lead toslower decision making within companies. Extracting expense data and invoicesfrom different departments, as well as consolidating them and summarising the information, is incredibly time consuming.

Because spreadsheets are prone to inaccuracies, everyone involved in processing the information must double-check the data as much as possible, which can further slow theprocess.

5. **Lack of version control**

The sharingof Excel spreadsheets from team to team might lead to concerns with the data’sversion and validity. You should considerwho had the most recentaccess to the data. Who did what to the spreadsheet and when? Can you confirmthat the calculations are correct? If you don’t trust the answers, you may need to start all over again.

6. **Data isn’tupdated in real-time**

Excel spreadsheets don’t update in real-time, so each updaterequires manual input. Because Excel spreadsheets can be difficult to modify, they are usually updated at the endof the day or every few days. Typically, this entails keeping daily paper records andthen manually entering them to update the Excel spreadsheet at a later date. Not only is this a waste of time, but it also raises the likelihood of data being entered inaccurately or decisionsbeing made based on out-of-date information.

7. **Increased potentialto lose important data**

If a spreadsheet owneris unfamiliar with best practices for data storageand backup, theymight keep just one version of their spreadsheet in a single location, such as on their desktop.

In the event of a technical issue, however, there’sno guarantee of complete data recovery, meaninga company could lose all of their vital data in a split-second.

a. Improved customer service
b. Cloud-based solution
c. Order Fulfillment
d. Harness Customer Loyalty and Retention

10.2 Disadvantages

1. **Susceptible to costly human errors**

Did you know that up to 9 out of 10 spreadsheets consistof human errors?

Unfortunately, even the smallest of mistakes in a spreadsheet can cause catastrophic consequences. Fidelity Magellan Fund once suffered a $2.6 billion overstatement when an accountant accidentally omittedthe minus sign on a net capitalloss of $1.3 billion.

There is always a greater chance of human error with manual processes, especially when it comes tocomplexdata sets, such as those involved with expense management. Failure to accuratelytrack you company’s expenditure and pay invoices on time can wreak havoc on your business’s bottom line.

2. **Lack of collaboration and access**

Because Excel spreadsheets are a single file, only one user at a time may access and modify thedata. It can also be challenging to collaborate with other departments because you have to manuallyshare or email a copy of the relevant spreadsheet with your colleagues.

When it comes to expense management data, however, these Excel spreadsheets are frequently shared and proofed across numerous teams and departments. To guarantee that everyone is viewingthe current version, users must be rigorous about version control and sharing when updates are made.

3. **Time-consuming manualprocesses**

The quantity of expense management data you need to review, analyse, and track will grow as yourbusiness evolves. The only way to validate your data when using Excel spreadsheets, however, is tomanually double check and re-enter any inaccurate information. This is a time-consuming and labour-intensive task.

As a result, Excel spreadsheets slow workers down and reduce accuracy by requiring them to perform repetitive processes that could be simplified or automated using expense management andinvoicing software.

4. **Inaccuracy leads to slowerdecision making**

There’s no denying that manual processes which increase the chances of inaccuracy lead to slowerdecision making within companies. Extracting expense data and invoices from different departments, as well as consolidating them and summarising the information, is incredibly time consuming.

Because spreadsheets are prone to inaccuracies, everyone involved in processing the informationmust double-check the dataas much as possible, which can further slow the process.

5. **Lack of version control**

The sharing of Excel spreadsheets from team to team might lead to concerns with the data’s versionand validity. You should consider who had the most recent access to the data. Who did what to the spreadsheet and when? Can you confirm that the calculations are correct? If you don’t trust the answers, you may need to start allover again.

6. **Data isn’t updated in real-time**

Excel spreadsheets don’t update in real-time, so each update requires manual input. Because Excel spreadsheets can be difficult to modify, they are usually updated at the end of the day or every few days. Typically, this entails keeping daily paper records and then manually entering them to update the Excel spreadsheet at a laterdate. Not only is this a waste of time,but it also raises the likelihood
of data being enteredinaccurately or decisionsbeing made based on out-of-date

information.

7. **Increased potentialto lose important data**

If a spreadsheet owner is unfamiliar with best practices for data storage and backup, they mightkeep just one version of their spreadsheet in a single location, such as on their desktop.

11.CONCLUSION

Taking proper care of our record is crucial in every business, no matter how big or little, we must understand. We must educate ourselves about the idea of effective inventory management and its applications because we can see that managers do not fully grasp it. A company's inventory management system is one of the reasons for its failure. Many customs to combat failure are present, and we can start from this point. Modern technologies can support us in managing and keeping an eye on our inventory. We may learn, put new ideas into practice, and assess our company**.**

12.FUTURE SCOPE

a. It will have various options to keep record (for example Food, Travelling Fuel, Salary etc.).

b. Automatically it will keep on sendingnotifications for our daily expenditure.

c. In today’s busy and expensive life, we are in a great rush to make moneys, but at the end of the month we broke off. As we are unknowingly spending money on title and unwanted things. So, we have come over with the plan to follow our profit.

d. Here user can define their own categories for expense type like food,clothing, rent and bills where they have to enter the money that has been spend and likewise can add some data in extra data to indicate the expense.

APPENDIX

SOURCE CODE:

LOGIN PAGE

```
            <!DOCTYPE html>
    <html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="viewport" content="width=device-width, initial-
scale=1.0">
        <link rel="stylesheet" type="text/css" href="{{
url_for('static',filename='index.css')}}">

        <title>Expense Tracker | Login</title>
        <script>
            window.addEventListener( "pageshow", function ( event ) {
            var historyTraversal = event.persisted || ( typeof window.performance
!= "undefined" && window.performance.navigation.type === 2 );
            if ( historyTraversal ) {
                window.location.reload();
            }
            });
        </script>
    </head>
    <body>
        <div class="login-body">
            <div class="inner-body">
                <h1 class="h11">Personal Expence Tracker</h1>
                <div class="container">
```

```
                    <div class="box">
                        <form action="/login" method="post">
                            <span class="text-center">LOGIN</span>
                            <div class="input-container">
                                <h4 class="msg">{{m}}</h4>
                                <label>Username</label>
                                <input type="text" name="username" required=""/>
                                <label>Password</label>
                                <input type="password" name="password"  required=""/>
                                <button  class="btn">submit</button>
                            </div>
                        </form>
                        <label>Forgot password? </label><br/>
                        <a class="sign" href="/register"><label>Sign-up</label></a>
                    </div>
                    <div class="box1">
                        <h1>Track Your Daily Expenses ! Here</h1>
                        <img src="{{url_for('static',filename='Mar-
Business_11.jpg')}}"/>
                    </div>
                </div>
            </div>
        </div>
    </body>
    </html>
```

HOME PAGE

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" type="text/css" href="{{url_for('static',filename='home.css')}}"/>
    <title>Expense Tracker | Home</title>
    <script>
        window.addEventListener( "pageshow", function ( event ) {
        var historyTraversal = event.persisted || ( typeof window.performance != "undefined" &&
window.performance.navigation.type === 2 );
        if ( historyTraversal ) {
            window.location.reload();
        }
        });
    </script>
</head>
<body>
    <div class="home-container">
        <h1 class="h11">Personal Expence Tracker</h1>
        <div class="inner-home">
            <nav class="side_bar">
                <ul>
                    <li class="avator"><img src="{{url_for('static',filename='man.png')}}"/></li>
                    <li>Welcome {{session.get('name')}}</li>
                    <li>Expense till date {{session.get('expense')}}</li>
                    <li>Your Available Balance is {{session.get('income')}}</li>
                    <li><a href="/transactions">Transactions</a></li>
                    <li><a href="/changepass">Change Password</a></li>
                    <li><a href="/logout">logout</a></li>
                </ul>
            </nav>
            <div class="rightside">
                <div class="right-top">
                    <h2>Track your bank accounts with us</h2>
                    <a href="/credit"><button class="btn">Add income</button></a>
                    <a href="/debit"><button class="btn">Add Expense</button></a>
                </div>
                <div class="home_img">
                    <img  src="{{url_for('static',filename='home.jpg')}}"/>
                </div>
        <div class="transaction_table">
          <h2>Your previous Transactions</h2>
          <table>
            <thead>
              <th>Amount</th>
              <th>Reason</th>
              <th>Type</th>
              <th>Date</th>
              <th>TIME</th>
            </thead>
            {% set a=0%}
            {% for x in tra %}
            <tr>
              {% if a < 5 %}
                <td>{{x.AMOUNT}}</td>
                <td>{{x.REASON}}</td>
                <td>{{x.TYPE}}</td>
                <td>{{x.DATE}}</td>
                <td>{{x.TIME}}</td>
                {% set a=a+1 %}
              {% endif %}
            </tr>

            {% endfor %}
          </table>
        </div>
      </div>
    </div>
  </div>
</body>
</html>
```

The python code to Connect with DB

```
from glob import escape
from flask import Flask, render_template, session, request, redirect, url_for
```

```
import ibm_db
from datetime import date
from datetime import datetime

conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=fbd88901-ebdb-4a4f-a32e-
9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32731;SECURITY=SS
L;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=brd83146;PWD=86sieylZYORbSY3i"
,",")

app = Flask(__name__)
app.secret_key = 'name'
# url routing

@app.route('/')
def index():
    if 'name' in session:
        return redirect('home')
    else:
        return render_template('index.html',m="")

@app.route('/login', methods=['POST','GET'])
def log():
    uname = request.form['username']
    password = request.form['password']
    account = login(uname,password)
    if account:
        return redirect(url_for('home'))
    else:
        return render_template('index.html',m="* Invalid Credintials")

@app.route('/addacc')
def addacc():
    return render_template('addacc.html')

@app.route('/home')
def home():
    if 'name' in session:
        addinc()
        sql = "select expense from expenses where username=?"
        stmt = ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,session['name'])
        ibm_db.execute(stmt)
        expense = ibm_db.fetch_assoc(stmt)
        expense = expense['EXPENSE']
        session['expense'] = expense
        if 'income' in session:
            return render_template('home.html',tra=transactions())
        else:
            return redirect('addacc')
    else:
        return redirect(url_for('index'))

@app.route('/register')
def register():
    return render_template('register.html',msg="")

@app.route('/credit')
def credit():
    if 'name' in session:
        return render_template('credit.html')
    else:
        return redirect(url_for('index'))

@app.route('/debit')
def debit():
    if 'name' in session:
        return render_template('debit.html')
    else:
        return redirect(url_for('index'))

@app.route('/changepass')
def changepass():
    if 'name' in session:
        return render_template('changepass.html')
    else:
        return redirect(url_for('index'))

@app.route('/transactions')
def transactionsfull():
    if 'name' in session:
        transs = tran()
        return render_template('transactions.html',tra =transs )
    else:
        return redirect(url_for('index'))

# API call
#login
def login(name,password):
    sql = "SELECT * FROM users WHERE username=? and password=?"
    stmt = ibm_db.prepare(conn,sql)
    ibm_db.bind_param(stmt,1,name)
    ibm_db.bind_param(stmt,2,password)
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    if account:
        session['name'] = name
        return account
    else:
        return 0

#logout
@app.route('/logout')
def logout():
    session.pop('name',None)
    session.pop('income',None)
    return redirect(url_for('index'))
```

```
# new user registration
@app.route('/newuser', methods=['post','GET'])
def reg():
    uname = request.form['uname']
    name = request.form['name']
    password = request.form['password']
    email = request.form['email']
    sql = "select * from users where username=?"
    stmt  = ibm_db.prepare(conn,sql)
    ibm_db.bind_param(stmt,1,uname)
    ibm_db.execute(stmt)
    user = ibm_db.fetch_assoc(stmt)
    if user:
        return render_template('/register.html',msg="* Username exists")

    sql = "select * from users where email=?"
    stmt  = ibm_db.prepare(conn,sql)
    ibm_db.bind_param(stmt,1,email)
    ibm_db.execute(stmt)
    user = ibm_db.fetch_assoc(stmt)
    if user:
        return render_template('/register.html', msg="* Email exists")

    sql = "insert into users (username,password,name,email) values (?,?,?,?)"
    stmt = ibm_db.prepare(conn,sql)
    ibm_db.bind_param(stmt,1,uname)
    ibm_db.bind_param(stmt,2,password)
    ibm_db.bind_param(stmt,3,name)
    ibm_db.bind_param(stmt,4,email)
    ibm_db.execute(stmt)
    return redirect(url_for('index'))

# checking for new user
def addinc():
    sql1= "SELECT * FROM income WHERE username=?"
    stmt = ibm_db.prepare(conn,sql1)
    ibm_db.bind_param(stmt,1,session.get('name'))
    ibm_db.execute(stmt)
    inc = ibm_db.fetch_assoc(stmt)
    if inc:
        session['income'] = inc['BALANCE']
    else:
        return redirect(url_for('addacc'))

# inserting income and balance for new user
@app.route('/addincome', methods=['POST'] )
def addin():
    income = request.form['income']
    name = request.form['uname']
    balance = request.form['balance']
    sql = "insert into income (username,income,balance) values (?,?,?)"
    stmt = ibm_db.prepare(conn,sql)
    ibm_db.bind_param(stmt,1,name)
    ibm_db.bind_param(stmt,2,income)
    ibm_db.bind_param(stmt,3,balance)
    ibm_db.execute(stmt)
    session['income'] = balance

    #adding expense data
    sql = "insert into expenses (username,expense) values (?,'0')"
    stmt = ibm_db.prepare(conn,sql)
    ibm_db.bind_param(stmt,1,name)
    ibm_db.execute(stmt)
    return redirect(url_for('home'))

#addding credit
@app.route('/addcredit', methods=['POST','GET'])
def addcredit():
    amount = request.form['amount']
    reason = request.form['reason']
    sql = "insert into transactions (username,reason,amount,type,date,time) values
(?,?,?,'credit',?,?)"
```

```
    stmt = ibm_db.prepare(conn,sql)
    ibm_db.bind_param(stmt,1,session['name'])
    ibm_db.bind_param(stmt,3,amount)
    ibm_db.bind_param(stmt,2,reason)
    ibm_db.bind_param(stmt,4,date.today())
    tt = datetime.now()
    time = tt.strftime("%H:%M:%S")
    ibm_db.bind_param(stmt,5,time)
    ibm_db.execute(stmt)
    #updating balance after credit
    acc = int(amount)
    new_var = session['income']
    balance =  int(new_var) + acc
    sql = "update INCOME set BALANCE=? where USERNAME=?"
    stmt = ibm_db.prepare(conn,sql)
    ibm_db.bind_param(stmt,1,balance)
    ibm_db.bind_param(stmt,2,session['name'])
    ibm_db.execute(stmt)

    return redirect(url_for('credit'))

#changing password
@app.route('/change', methods=['POST','GET'])
def chanpass():
    oldpass = request.form['oldpass']
    newpass = request.form['newpass']
    sql = "select password from users where username=?"
    stmt = ibm_db.prepare(conn,sql)
    ibm_db.bind_param(stmt,1,session['name'])
    ibm_db.execute(stmt)
    password = ibm_db.fetch_assoc(stmt)
    password = password['PASSWORD']
    if (password == oldpass):
        sql = "update users set password=? where username=?"
        stmt = ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,newpass)
        ibm_db.bind_param(stmt,2,session['name'])
        ibm_db.execute(stmt)
        return redirect(url_for('home'))
    else:
        return render_template('changepass.html',msg = "old password does not match")

#adding debit
@app.route('/adddebit', methods=['POST','GET'])
def adddebit():
    amount = request.form['amount']
    reason = request.form['reason']
    sql = "insert into TRANSACTIONS (username,reason,amount,type,date,time) values
(?,?,?,'debit',?,?)"
    stmt = ibm_db.prepare(conn,sql)
    ibm_db.bind_param(stmt,1,session['name'])
    ibm_db.bind_param(stmt,3,amount)
    ibm_db.bind_param(stmt,2,reason)
    ibm_db.bind_param(stmt,4,date.today())
    tt = datetime.now()
    time = tt.strftime("%H:%M:%S")
    ibm_db.bind_param(stmt,5,time)
    ibm_db.execute(stmt)

    #updating balance after debit
    acc = int(amount)
    new_var = session['income']
    balance =  int(new_var) - acc
    sql = "update INCOME set BALANCE=? where USERNAME=?"
    stmt = ibm_db.prepare(conn,sql)
    ibm_db.bind_param(stmt,1,balance)
    ibm_db.bind_param(stmt,2,session['name'])
    ibm_db.execute(stmt)
    session['income'] = balance



    # updating debit amount
    sql = "select expense from expenses where username=?"
    stmt = ibm_db.prepare(conn,sql)
    ibm_db.bind_param(stmt,1,session['name'])
    ibm_db.execute(stmt)
    expense = ibm_db.fetch_assoc(stmt)
    expense = expense['EXPENSE']
    expense = expense + acc
    sql = "update expenses set expense=? where username=?"
    stmt = ibm_db.prepare(conn,sql)
    ibm_db.bind_param(stmt,1,expense)
    ibm_db.bind_param(stmt,2,session['name'])
    ibm_db.execute(stmt)
    session['expense']=expense
    return redirect(url_for('debit'))

# transaction details
def transactions():
    t=[]
    sql= f"select * from TRANSACTIONS where username='{escape(session['name'])}'"
    stmt=ibm_db.exec_immediate(conn,sql)
    trans = ibm_db.fetch_assoc(stmt)
    a=1
    while trans != False and a<6:
        t.append(trans)
        trans = ibm_db.fetch_assoc(stmt)
        a +=1
    return t

def tran():
    t=[]
    sql= f"select * from TRANSACTIONS where username='{escape(session['name'])}'"
    stmt=ibm_db.exec_immediate(conn,sql)
    trans = ibm_db.fetch_assoc(stmt)
    while trans != False:
        t.append(trans)
        trans = ibm_db.fetch_assoc(stmt)
    return t

if __name__ == '__main__':
    app.run(debug=True,use_reloader=True)
```

CONCLUSION

Thus we conclude with our final deliverables of the project Personal expense tracker.

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-18816-1659690473

DEMO LINK:

DEMO VIDEO