IBM - NALAIYATHIRAN PROJECT

PERSONAL EXPENSE TRACKER APPLICATION

PROJECT REPORT

INDUSTRY MENTOR: KUSHBOO

FACULTY MENTOR: VANATHI

Submitted by

TEAM ID:PNT2022TMID29605

TEAM LEADER: PAVITHRA T
TEAM MEMBER: PRAVEEN R
TEAM MEMBER: PRAVEENA S
TEAM MEMBER: RABOONI S

In partial fullfillment for the award of the degree
of
BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE ENGINEERING
IN
THANTHAI PERIYAR GOVERNMENT INSTITUDE OF TECHONOLOGY-
VELLORE 632002

TABLE OF CONTENTS

1. Introduction

In today’s busy and expensive lives we are in a great rush to make money. Butat the end of the day we broke off. As we are unknowingly spending money on little and unwanted things. So, we have come over with the idea to track our earnings. DailyExpense Tracker (DET) aims to help everyone who are planning to know their expenses and save from it. DTE is a website in which user can add expenses on daily basis andits table will get generated and at the end based on user expenses report will be generated. User can select date range to calculate his/her expenses come over withthe idea to track our earnings. Personal Expense Tracker aims to help everyone who are planning to know their expenses and save from it. Personal Expense Tracker is awebsite in which user can add expenses on daily basis and at the end, based on user expenses report will be generated. User can select date range to calculate his/her expenses.

1.1 Project Overview

This website is used to track expenses and control spending beyond limits. while input data of expenses in website, we must select category which spent on andadditionally notes can be used to note the details of expenses. By entering those record we can track our expenses. we can generate reports in graphical, pie chat. We can also set limits to particular category which alerts in email when the limits exceed.

1.2 Purpose

At the end of certain period, users does not know where they spent their money and they spend more on needless expenses beyond budgets which leads to financial crisis. To avoid this people needs to track their expenses. While calculating in diary requires lot of manual calculation and lot of time. This is the purpose to go for websiteapplication to track expenses.

2. Literature Survey

2.1 Existing problem

People can’t able to track their expenses and spending more on unnecessary expenses which leads to money crisis. Without tracking people can’t know whether they exceed the limit of their budget. Diary notes requires lots of manual calculation and It reduces the interest to track expenses. User frustrated about they can’t remember where their money goes and can’t handle their cash flow. There is no alerting system about exceeding limits. There can be many disadvantages of using a manual accounting system. Accounting, for any business, can be a complex undertaking. A manual accounting system requires you to understand the accountingprocess in a way that may be unnecessary with a computerized accounting system.

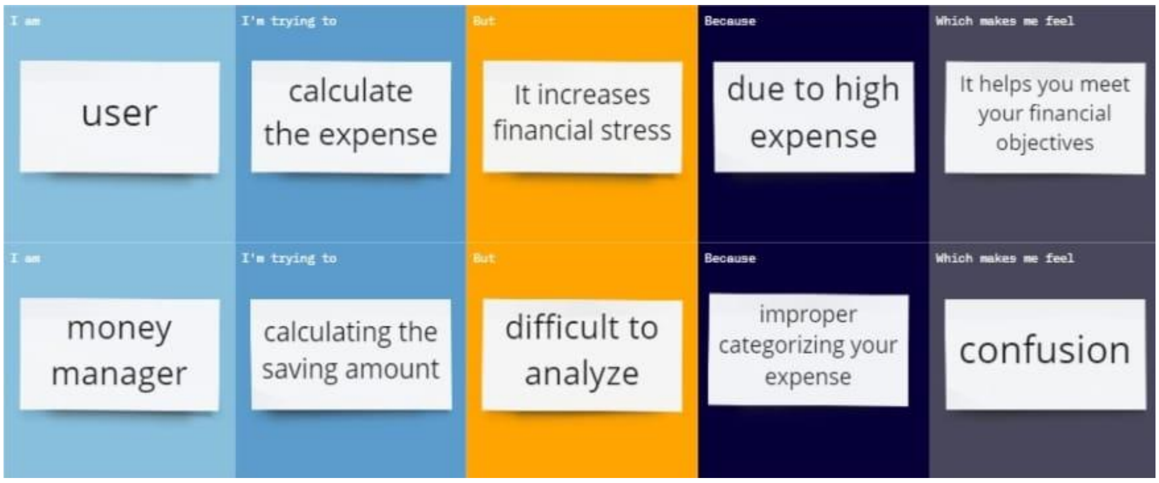
This can be an advantage or a disadvantage, depending on the person doing the bookkeeping; often, a specially trained professional is needed to ensure that accounting is done properly. Unrevealing the complexity of your financial records byhand may be time consuming. Since it takes time to generate reports.

2.2 References

S. NO	PAPERTITLE	AUTHOR NAME	PUBLICATION YEAR
1.	Expense Tracker : A Smart Approach to Track Everyday Expense	Hrithik Gupta, Anant Prakash Singh, Navneet Kumar and J. Angelin Blessy	2020
2.	Expense Manager Application	A Velmurugan, J Albert Mayan, P Niranjana and RichardFrancis	2020
3.	Income and ExpenseTracker	P. Thanapal , Mohammed Yaseen Patel, T.P. Lokesh Raj and J. Satheesh Kumar	2015
4.	IRJET- Online Income and Expense Tracker	S. Chandini1, T. Poojitha2, D. Ranjith3, V.J. Mohammed Akram4, M.S. Vani5, V.Rajyalakshmi	2019

2.3 Problem Statement Definition

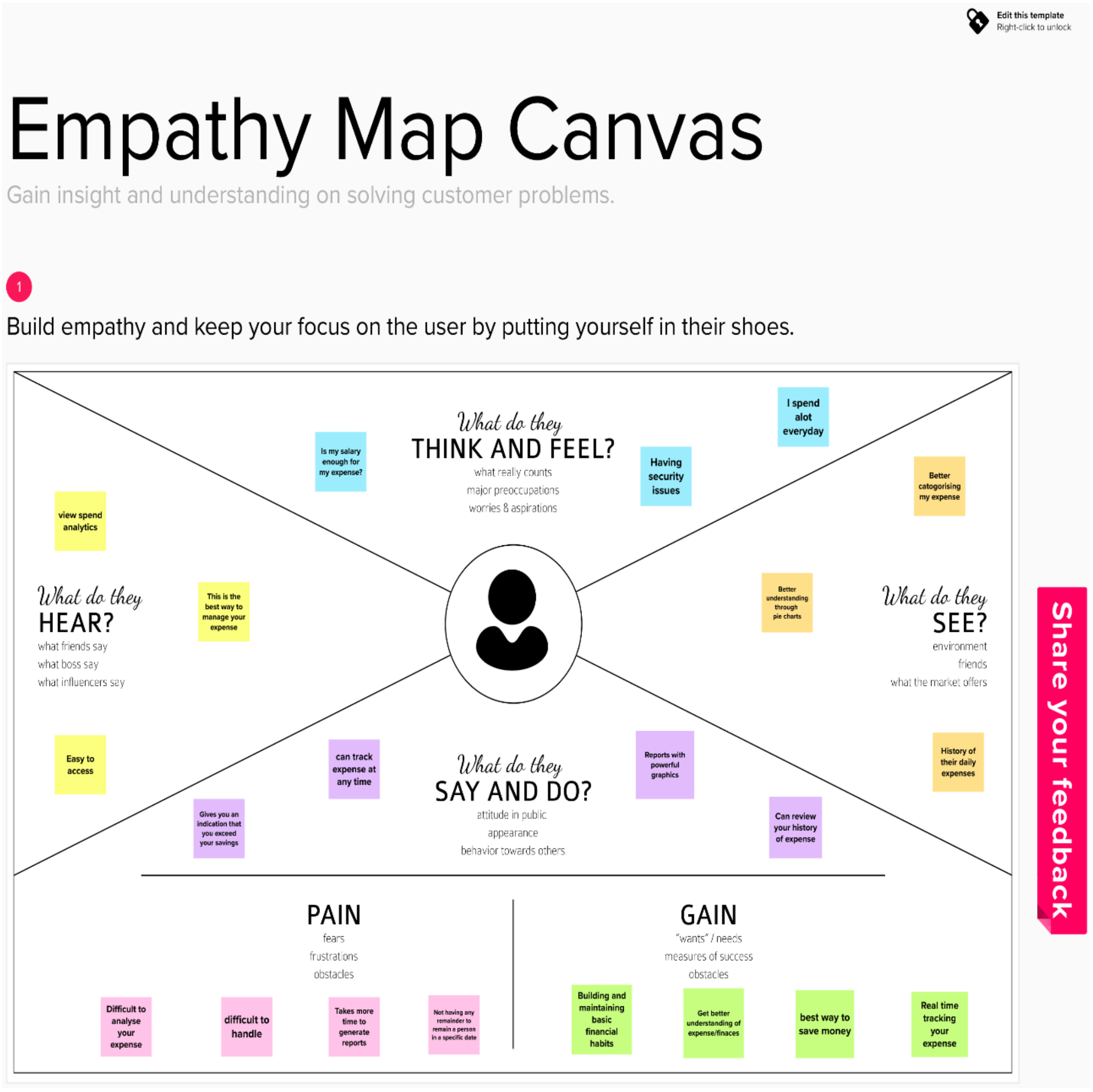
Our project helps the user to keep track their expenses and determine whether they are spending as per their set budget. Potential input the required data such as the expense amount, merchant, category, and date when the expense was made. Which allows users to track their expenses daily, weekly, monthly, and yearly in terms ofsummary, bar graphs, and pie-charts. It is like automated diary which requires no burdenof manual calculation users need to and enables the user to not just keep the control on the expensesbut also to generate and save reports. Users can insert and delete transactions. We cancompare with past expenses. Customized email alerts are used alerts user when limit exceeds. 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.



I am	I'm trying to	But	Because	Which makes me feel
user	calculate the expense	It increases financial stress	due to high expense	It helps you meet your financial objectives
money manager	calculating the saving amount	difficult to analyze	improper categorizing your expense	confusion

3. Ideation and Proposed Solution

3.1 Empathy Map Canvas



Empathy Map Canvas

Gain insight and understanding on solving customer problems.

1

Build empathy and keep your focus on the user by putting yourself in their shoes.

What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations

- Is my salary enough for my expense?
- Having security issues
- I spend alot everyday
- Better catogorising my expense

What do they HEAR?
what friends say
what boss say
what influencers say

- view spend analytics
- This is the best way to manage your expense
- Easy to access

What do they SEE?
environment
friends
what the market offers

- Better understanding through pie charts
- History of their daily expenses

What do they SAY AND DO?
attitude in public
appearance
behavior towards others

- can track expense at any time
- Gives you an indication that you exceed your savings
- Reports with powerful graphics
- Can review your history of expense

PAIN
fears
frustrations
obstacles

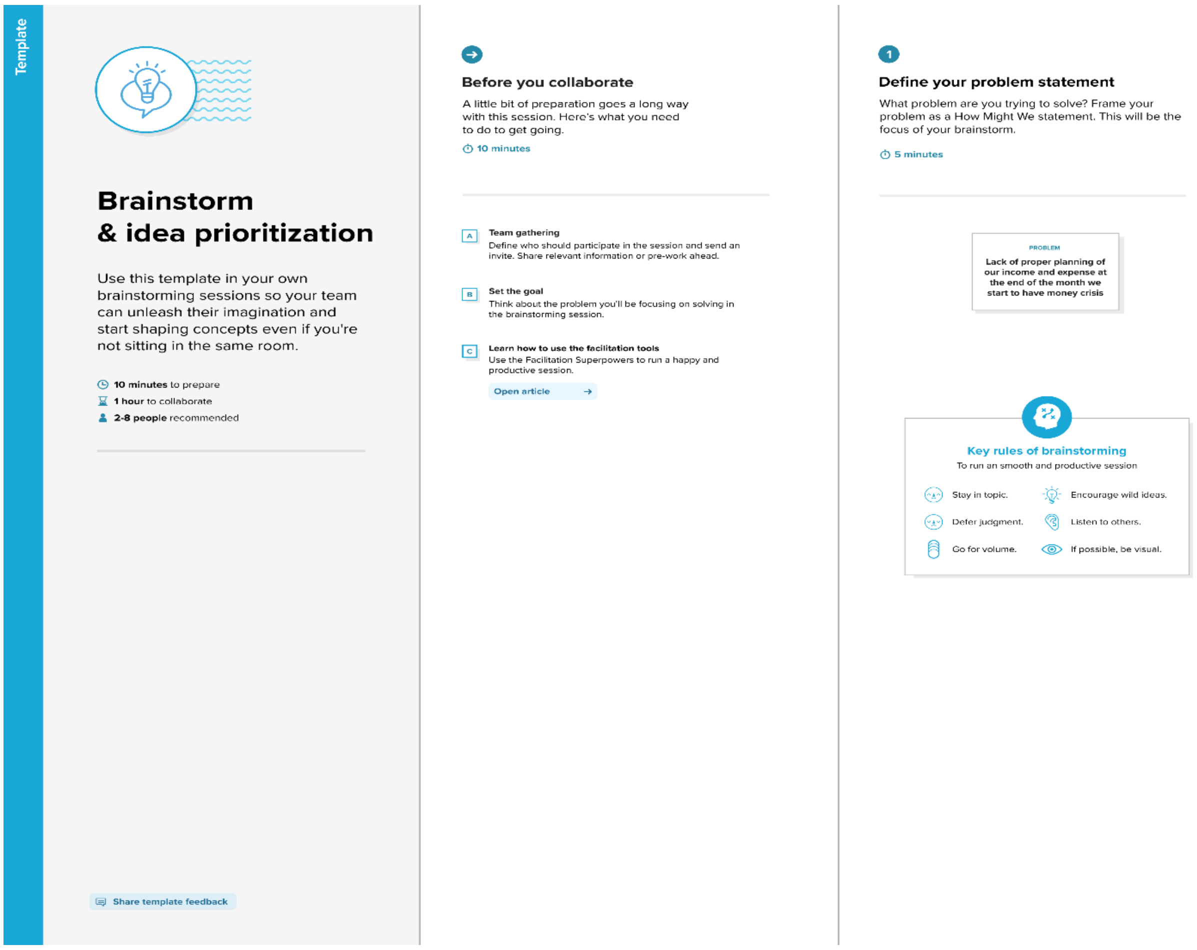
- Difficult to analyse your expense
- difficult to handle
- Takes more time to generate reports
- Not having any remainder to remain a person in a specific date

GAIN
"wants" / needs
measures of success
obstacles

- Building and maintaining basic financial habits
- Get better understanding of expense/finaces
- best way to save money
- Real time tracking your expense

3.2 Ideation & Brainstorming

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



Template

Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Share template feedback

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

1 **Define your problem statement**

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM

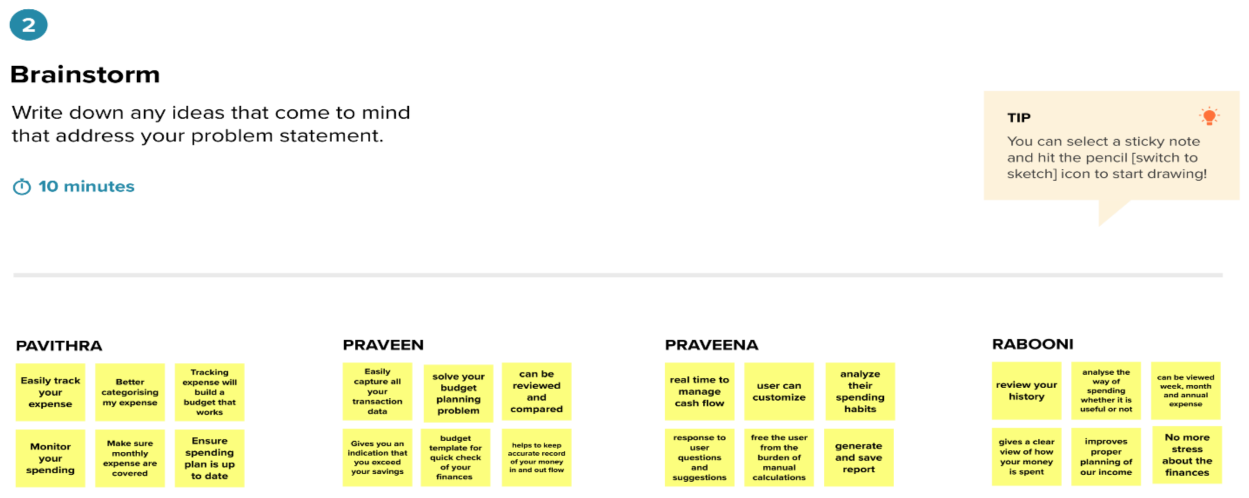
Lack of proper planning of our income and expense at the end of the month we start to have money crisis

Key rules of brainstorming

To run an smooth and productive session

Stay in topic.
Encourage wild ideas.
Defer judgment.
Listen to others.
Go for volume.
If possible, be visual.

Step-2: Brainstorm, Idea Listing


2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

PAVITHRA

- Easily track your expense
- Better categorising my expense
- Tracking expense will build a budget that works
- Monitor your spending
- Make sure monthly expense are covered
- Ensure spending plan is up to date

PRAVEEN

- Easily capture all your transaction data
- solve your budget planning problem
- can be reviewed and compared
- Gives you an indication that you exceed your savings
- budget template for quick check of your finances
- helps to keep accurate record of your money in and out flow

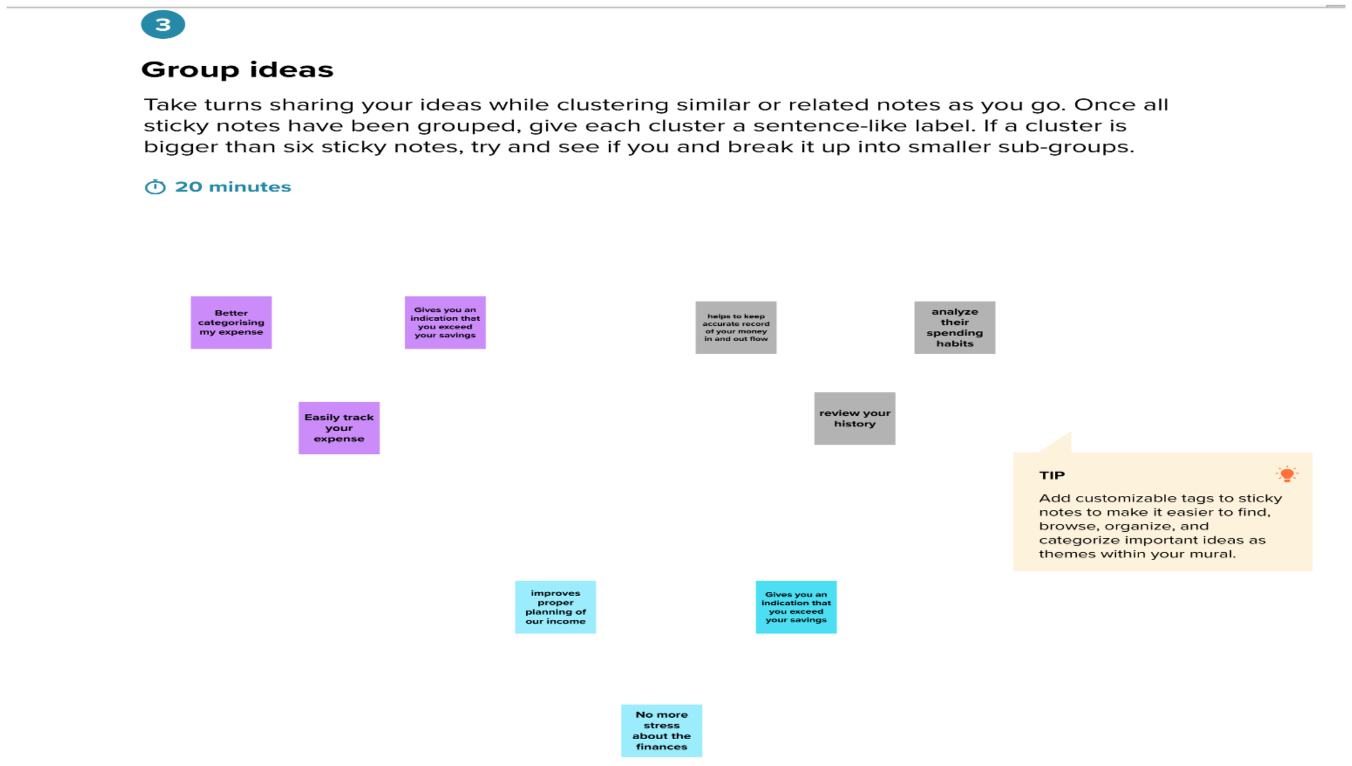
PRAVEENA

- real time to manage cash flow
- user can customize
- analyze their spending habits
- response to user questions and suggestions
- free the user from the burden of manual calculations
- generate and save report

RABOONI

- review your history
- analyse the way of spending whether it is useful or not
- can be viewed week, month and annual expense
- gives a clear view of how your money is spent
- improves proper planning of our income
- No more stress about the finances

Step-3: Idea Grouping


3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

- Better categorising my expense
- Gives you an indication that you exceed your savings
- Easily track your expense

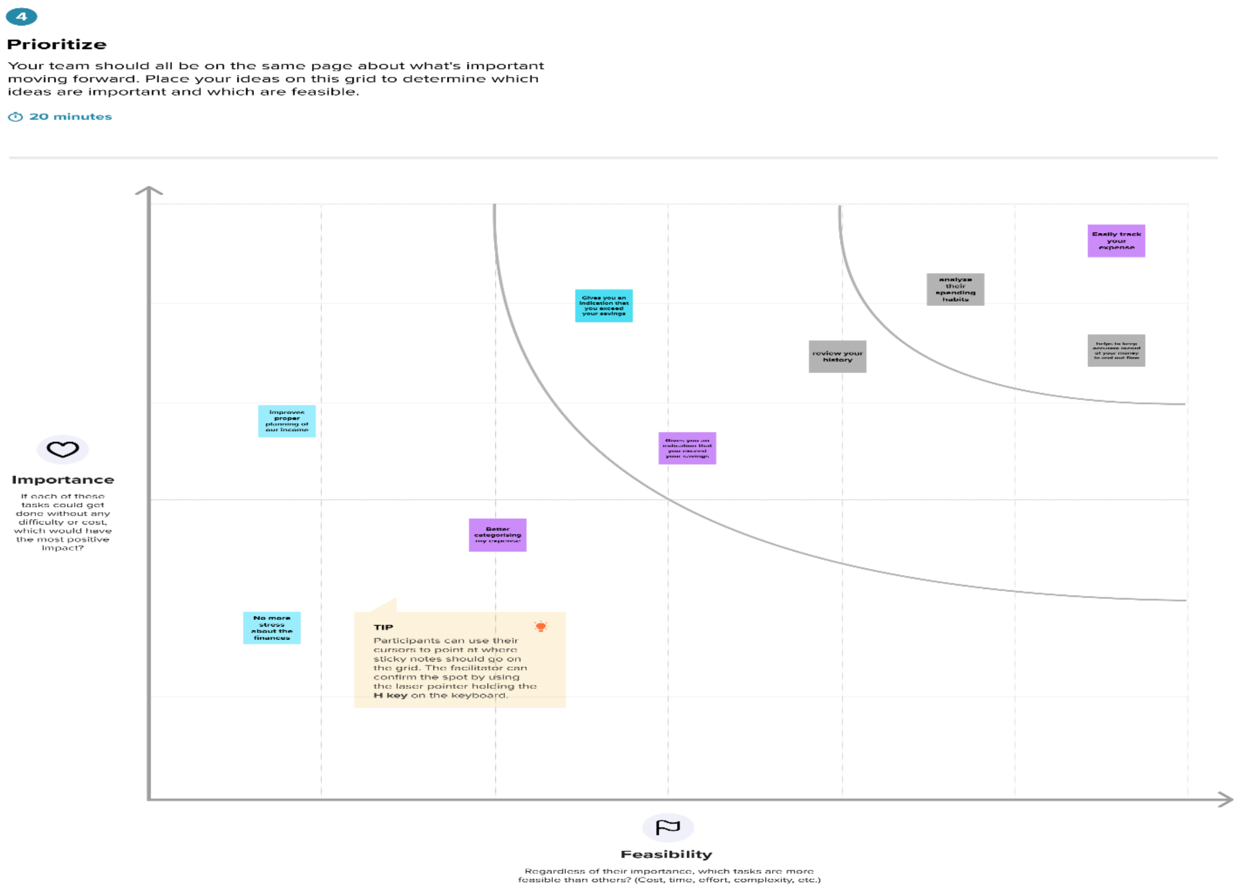
- helps to keep accurate record of your money in and out flow
- analyze their spending habits
- review your history

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

- improves proper planning of our income
- Gives you an indication that you exceed your savings
- No more stress about the finances

Step-4: Idea Prioritization


4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.

Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

Feasibility

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

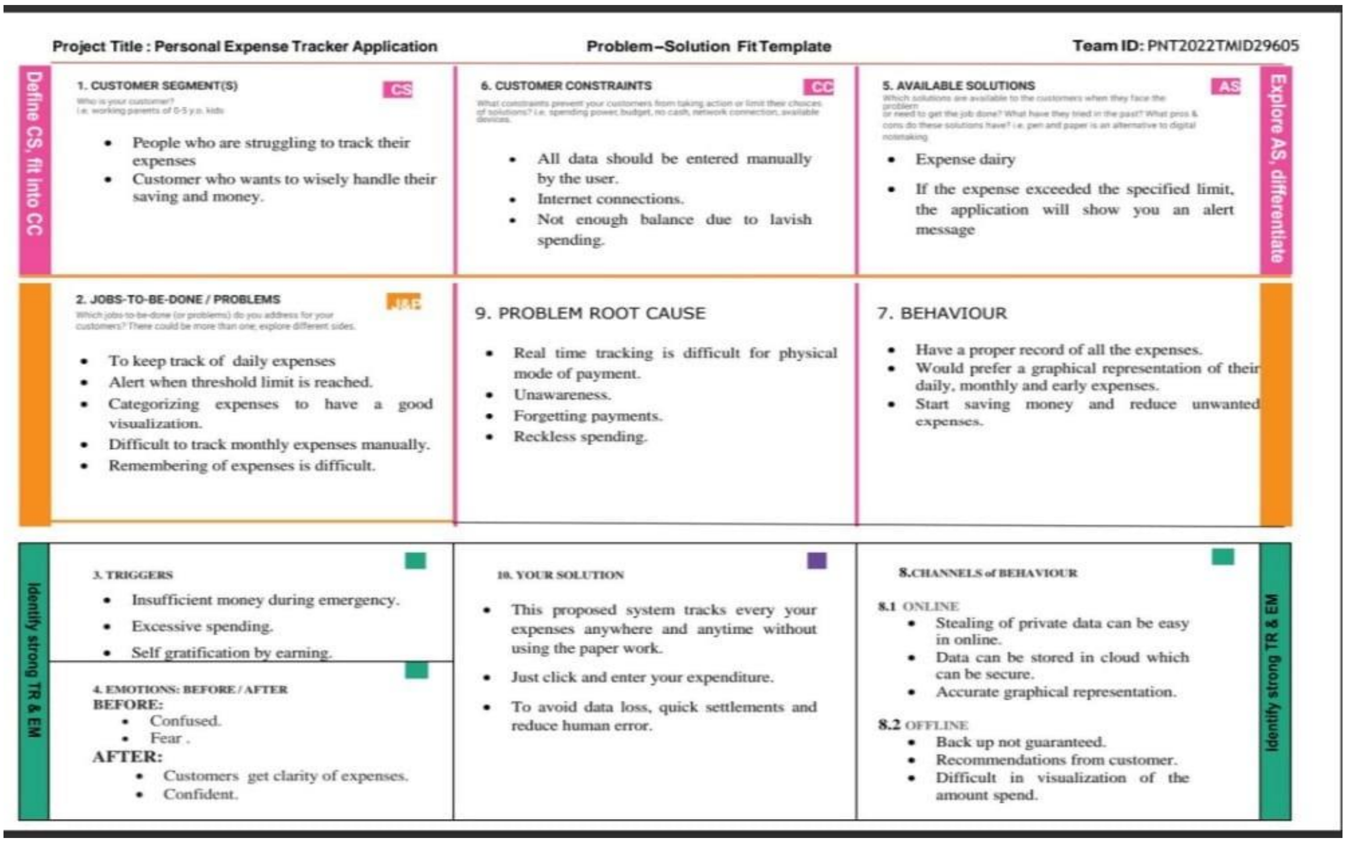
Easily track your expense
analyze their spending habits
Gives you an indication that you exceed your savings
review your history
helps to keep accurate record of your money in and out flow
Improves proper planning of our income
Gives you an indication that you exceed your savings
Better categorising my expense
No more stress about the finances

3.3 Proposed solution

S.no	Parameter	Description
1.	Problem Statement (Problem to be solved)	Attempting to manage the expenses of an individual in an efficient and manageable manner, as compared to the traditional way of expense tracking.
2.	Idea/Solution description	An expense tracker app allows you to monitor and categorize your expenses across different bank and investment accounts and credit cards. Some of these apps also offer budgeting tools, credit monitoring, mileage tracking, receipt keeping, and advice to grow your net worth.

3.	Uniqueness/Novelty	The application gives the user a chance to plan his/her monthly expenses at the start of the month. Besides this, the user gets a notification .when he/she exceeds the limit that is set.
4.	Social Impact / Customer Satisfaction	With such applications, the public will start to plan their expenses better leading to their own financial stability. With more users, this application will ensure that financial state of our society improves.
5.	Business Model (Revenue Model)	The application can be provided based on user required feature and the cost depends on the usage.
6.	Scalability of the Solution	Since the application takes the same set of input from all the users and does not perform many complex computations, it will be easy for us to scale the application to a larger set of users.

3.4 Problem solution fit



Project Title : Personal Expense Tracker Application

Problem–Solution Fit Template

Team ID: PNT2022TMID29605

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

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

- People who are struggling to track their expenses
- Customer who wants to wisely handle their saving and money.

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.

- All data should be entered manually by the user.
- Internet connections.
- Not enough balance due to lavish spending.

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

- Expense dairy
- If the expense exceeded the specified limit, the application will show you an alert message

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.

- To keep track of daily expenses
- Alert when threshold limit is reached.
- Categorizing expenses to have a good visualization.
- Difficult to track monthly expenses manually.
- Remembering of expenses is difficult.

9. PROBLEM ROOT CAUSE

- Real time tracking is difficult for physical mode of payment.
- Unawareness.
- Forgetting payments.
- Reckless spending.

7. BEHAVIOUR

- Have a proper record of all the expenses.
- Would prefer a graphical representation of their daily, monthly and early expenses.
- Start saving money and reduce unwanted expenses.

Identify strong TR & EM

3. TRIGGERS

- Insufficient money during emergency.
- Excessive spending.
- Self gratification by earning.

4. EMOTIONS: BEFORE / AFTER

BEFORE:
- Confused.
- Fear .

AFTER:
- Customers get clarity of expenses.
- Confident.

10. YOUR SOLUTION

- This proposed system tracks every your expenses anywhere and anytime without using the paper work.
- Just click and enter your expenditure.
- To avoid data loss, quick settlements and reduce human error.

8.CHANNELS of BEHAVIOUR

8.1 ONLINE
- Stealing of private data can be easy in online.
- Data can be stored in cloud which can be secure.
- Accurate graphical representation.

8.2 OFFLINE
- Back up not guaranteed.
- Recommendations from customer.
- Difficult in visualization of the amount spend.

Identify strong TR & EM

4. Requirement Analysis

4.1 Functional Requirements

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration Form for collecting details.
FR-2	User Login	Enter User and Password.
FR-3	Forgot Password	Reseting the password by sending an OTP to user's mail.
FR-4	Calendar	Personal Expense Tracker Application must allow user to add the data to their expenses.
FR-5	Dashboard	User can add the expense and can evaluate them using the provided options.

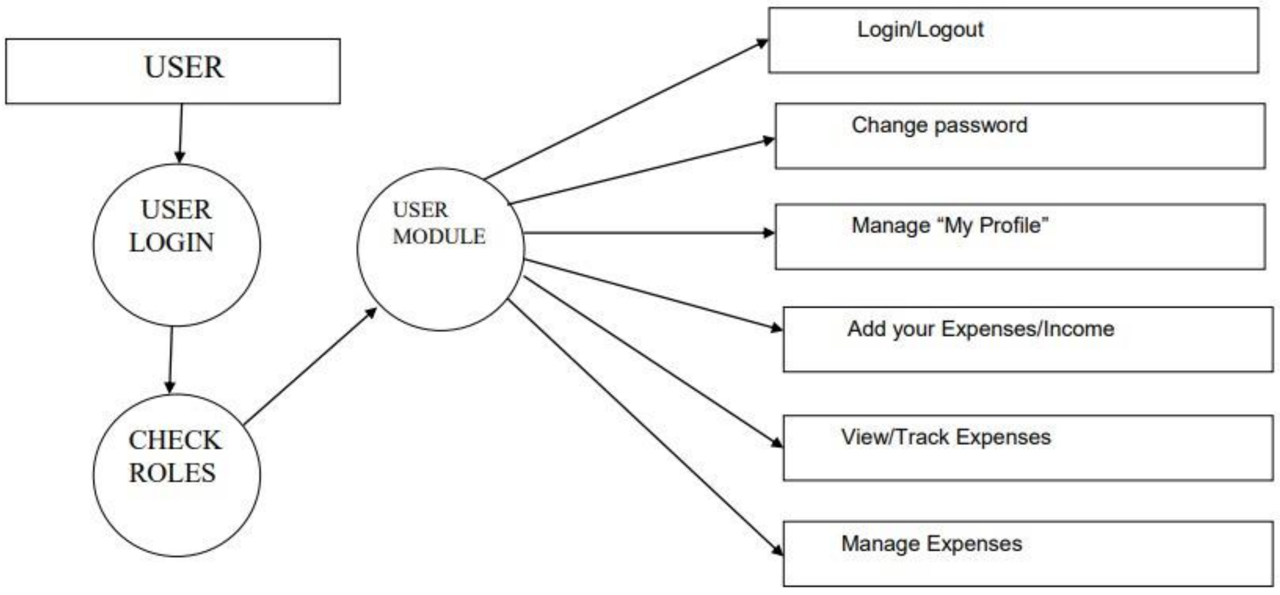
4.2 Non-functional Requirements

NFR No.	Non-Functional Requirement	Description
NFR-1	Usability	Customer can use the application in almost all the web browsers. Application is with good looking and detailed UI, which makes it more friendly to use.
NFR-2	Security	Customers are asked to create an account for themselves using their email which is protected with an 8-character long password, making it more secure.
NFR-3	Reliability	Each data record is stored on well built efficient database schemes .There is no risk of data loss.
NFR-4	Performance	Customer will have a smooth experience while using the application, as it is simple and is well optimized.
NFR-5	Availability	Application is available 24/7 as it is hosted on IBM Cloud.
NFR-6	Scalability	The ability to appropriately handle increasing demands. In future, may be cross-platform mobile applications can be developed as the user base grows.

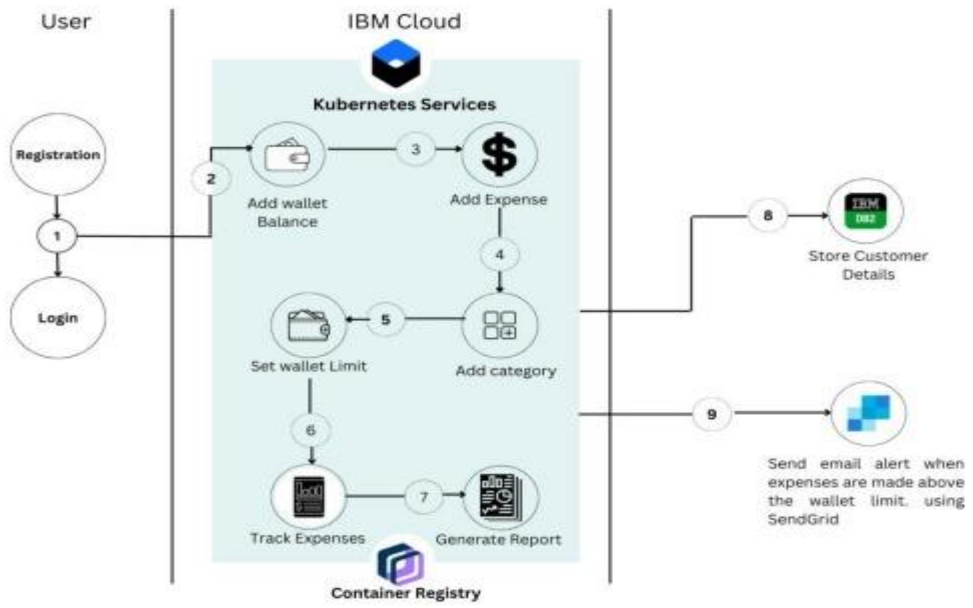
5.Project Design

5.1 Data-Flow Diagrams

A Data flow diagram is a traditional representation of the information flows with in 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 ,where data is stored**.**



5.2 Solution & Technical Architecture



5.3 User stories

Table-1 : Components & Technologies

S.No	Component	Description	Technology
1.	User Interface	User Interacts with application e.g., Web UI, Mobile app, chatbot	HTML, CSS, JavaScript
2.	Registration and Login	To develop the application to connect the count.	Python Docker
3.	Application Logic-1	The application contains the sign-in/sign-up where the user will log in to the main dashboard.	Java/Python
4.	Wallet Dashboard	IOBM cloud Kubernetes service provides a native Kubernetes experience that secure and easy to use . this tool is used to load-balance, scale and monitor the containers.	IBM Cloud Kubernetes services.
5.	Tracking of Expenses	IBM container registry enables to store and distribute the docker images in a managed, private registry.	IBM Cloud Container Registry

6.	Database	The income and expense data are stored in the MySQL database.	MySQL
7.	Cloud Database	With the use of cloud database service on Cloud, the user data are stored in a well secured manner.	IBM DB2,IBM Cloudant etc.,

Table-2: Application Characteristics

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Flask is an open source framework written in Python to implement this application	Python-Flask
2.	Security Implementations	The user accounts are configured to only allow access from users with specific privileges. This application provides high security to the user financial data. It can be done by using the container registry in IBM cloud database.	IBM DB2
3.	Scalable Architecture	Three-tier architecture- user server, application server and cloud server. This Application is anytime accessible .Kubernetes services, the crudest form of load balancing.	Python, IBM Cloud services
4.	Availability	The most basic type of load balancing is load distribution. The Docker load balancer runs on every node and can load balance requests across any of the containers on any of the hosts in the cluster.	Kubernetes and Docker
5.	Performance	The performance will be high. Because there will be no network traffics in the application.	IBM Container Registry

4.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority
Customer (Mobile user & web user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirmingmy 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 FacebookLogin	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 thelog 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 updatethe application.	I can fix the bug which arises for the customersand users of the application	Medium

5. Project planning & scheduling

5.1 Sprint planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
1	Registration	PETAS-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	Low	Rabooni
1	Registration	PETAS-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Praveena
1	Registration	PETAS-4	As a user, I can register for the application through Gmail	1	High	Pavithra
1	Login	PETAS-5	As a user, I can log into the application by entering email & Password	2	Low	Praveen
2	Workspace	PETAS-3	Workspace for personal expense tracking	2	High	Praveena
2	Charts	PETAS-7	Creating various graphs and statistics of customer's data	1	High	Pavithra
2	Connecting DB	PETAS-8	Linking database with dashboard	2	High	Praveen
3	Connecting DB	PETAS-9	Making dashboard interactive with JS	2	High	Rabooni
3	Sendgrid	PETAS-16	Using send grid to send mail to user about their expenses.	1	High	Praveen

3	Sendgrid	PETAS-17	Integrating both frontend and backend.	2	Low	Praveena
4.	Docker	PETAS-18	Creating image of website using docker	2	High	Pavithra
4.	Cloud Registry	PETAS-19	Uploading docker image to IBM cloud registry	2	High	Praveena
4.	Kubernetes	PETAS-20	Create container using the docker image and hosting the site.	2	High	Praveen
4.	Exposing	PETAS-21	Exposing IP/Ports for the site.	2	High	Rabooni

5.2 Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed	Sprint Release Date (Actual)
Sprint-1	10	6 Days	23 Oct 2022	28 Oct 2022	10	29 Oct 2022
Sprint-2	10	6 Days	04 Oct 2022	04 Nov 2022	10	05 Nov 2022
Sprint-3	10	6 Days	06 Nov 2022	11 Nov 2022	10	12 Nov 2022
Sprint-4	10	6 Days	13 Nov 2022	18 Nov 2022	10	19 Nov 2022

Velocity

We have a 6-day sprint duration and the velocity of the team is 10 (points per sprint).Calculating the team's average velocity (AV) per iteration unit (story points per day)

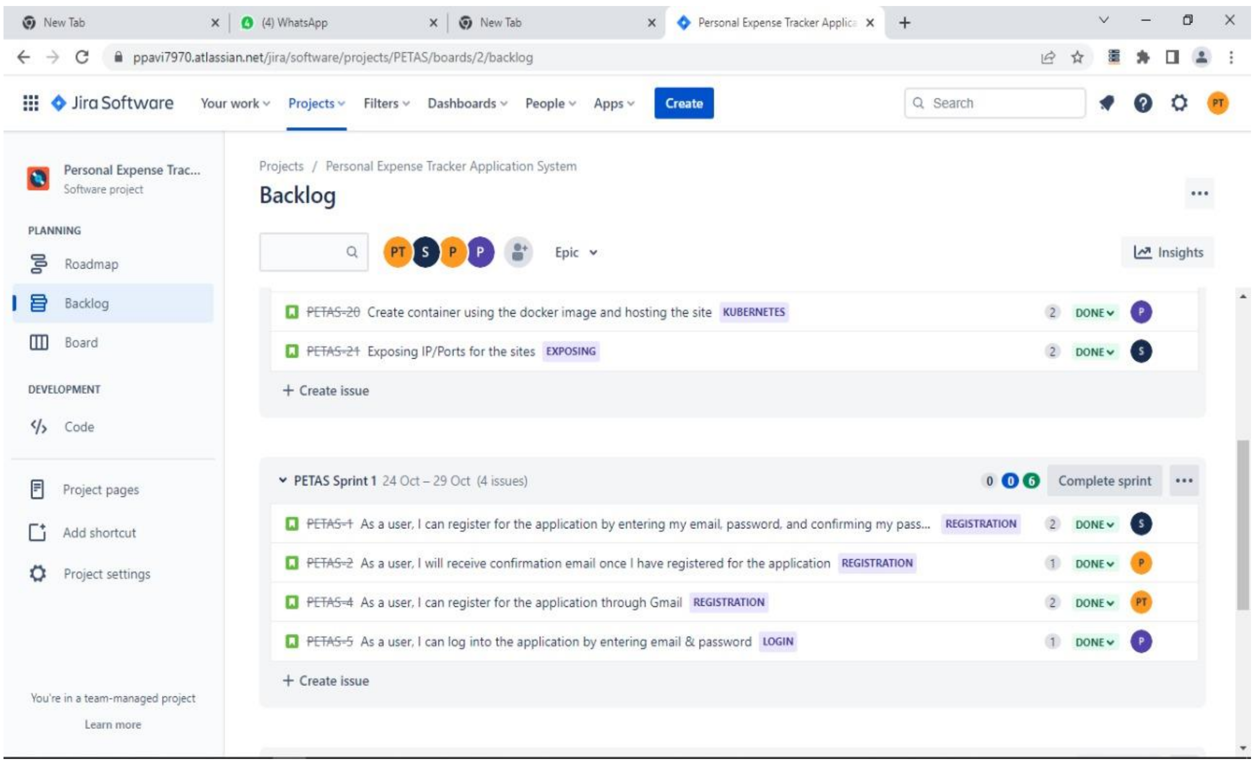
AV=Sprint duration/Velocity

=10/6=1.66

Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. Itis often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

Reports from JIRA



ppavi7970.atlassian.net/jira/software/projects/PETAS/boards/2/backlog

Projects / Personal Expense Tracker Application System

Backlog

~~PETAS-20~~ Create container using the docker image and hosting the site KUBERNETES 2 DONE

~~PETAS-21~~ Exposing IP/Ports for the sites EXPOSING 2 DONE

+ Create issue

PETAS Sprint 1 24 Oct – 29 Oct (4 issues) 0 0 6 Complete sprint

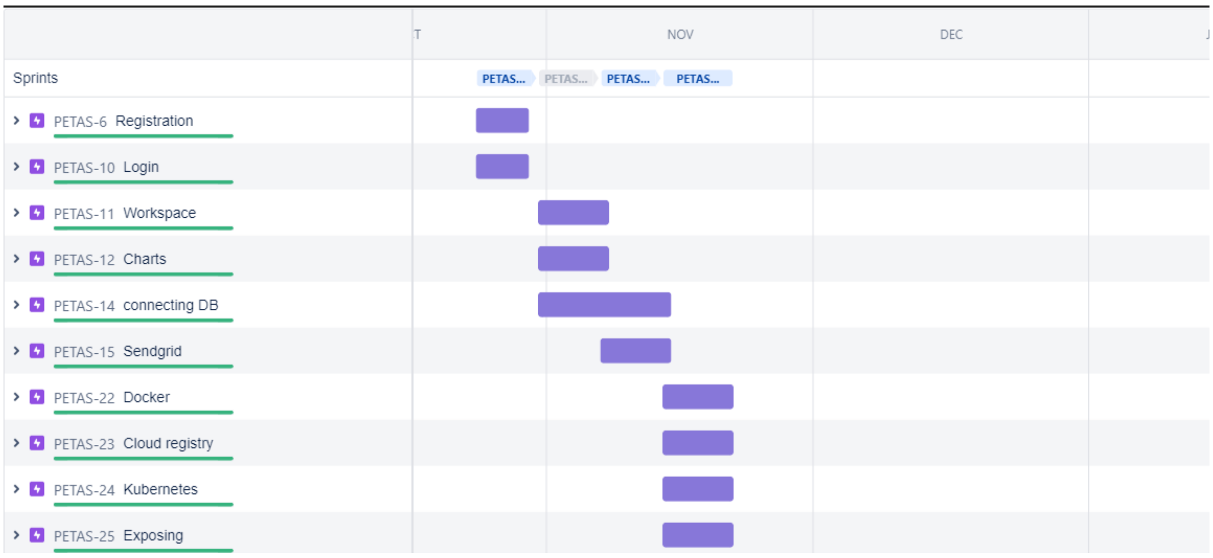
~~PETAS-1~~ As a user, I can register for the application by entering my email, password, and confirming my pass... REGISTRATION 2 DONE

~~PETAS-2~~ As a user, I will receive confirmation email once I have registered for the application REGISTRATION 1 DONE

~~PETAS-4~~ As a user, I can register for the application through Gmail REGISTRATION 2 DONE

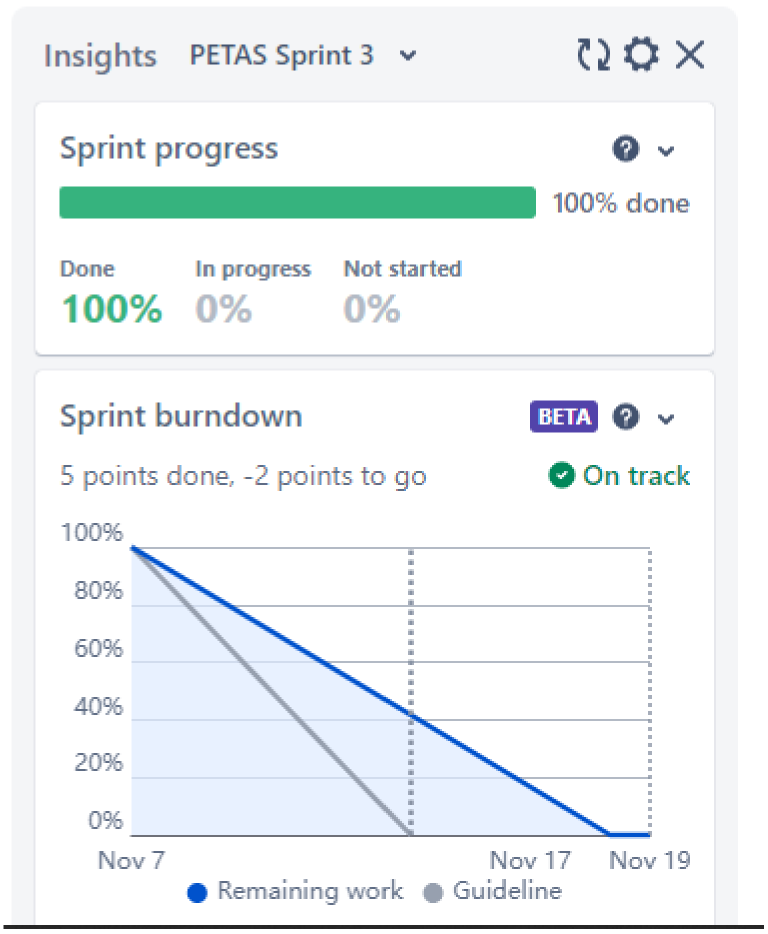
~~PETAS-5~~ As a user, I can log into the application by entering email & password LOGIN 1 DONE

+ Create issue



Sprints PETAS... PETAS... PETAS... PETAS...

- PETAS-6 Registration
- PETAS-10 Login
- PETAS-11 Workspace
- PETAS-12 Charts
- PETAS-14 connecting DB
- PETAS-15 Sendgrid
- PETAS-22 Docker
- PETAS-23 Cloud registry
- PETAS-24 Kubernetes
- PETAS-25 Exposing



Insights PETAS Sprint 3

Sprint progress

100% done

Done	In progress	Not started
100%	0%	0%

Sprint burndown BETA

5 points done, -2 points to go

On track

6. Coding and Solutioning

(Explain the features added in the project along with code)

6.1 Feature 1

Python

- Python is a widely-used, interpreted, object-oriented, and high-level programming language with dynamic semantics, used for general-purpose programming. It's everywhere, and people use numerous Python-powered devices on a daily basis, whether they realize it or not.
- Python was created by Guido van Rossum, and first released on February 20, 1991.
- Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Smalltalk, and Unix shell and other scripting languages.
- Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL) .
- It is easy to learn – the time needed to learn Python is shorter than for many other languages; this means that it's possible to start the actual programming fast
- It is easy to use for writing new software – it's often possible to write code faster when using Python.
- It is easy to obtain, install and deploy – Python is free, open and multiplatform; not all languages can boast that. Programming skills prepare you for careers in almost any industry and are required if you want to continue to more advanced and higher-paying software development and engineering roles.
- Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Flask

- Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries.
- It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself.
- Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.
- Applications that use the Flask framework include Pinterest and LinkedIn.

6.2 Feature 2

IBM DB2

- DB2 is a database product from IBM.

- It is a Relational Database Management System (RDBMS). DB2 is designed to store, analyze and retrieve the data efficiently.
- DB2 product is extended with the support of Object-Oriented features and non-relational structures with XML.
- Provide a massively parallel processing (MPP) architecture Exploits Hive, HBase and Apache Spark concurrently for best-in-class analytic capabilities.
- Provides low latency support for ad-hoc and complex queries, high performance, and federation capabilities Understands dialects from other vendors and various products 21 from Oracle, IBM® Db2® and IBM Netezza® Enables advanced row and column security.

KUBERNETES

- Kubernetes is also known as 'k8s'.
- Kubernetes is an extensible, portable, and open-source platform designed by Google in 2014.
- It is mainly used to automate the deployment, scaling, and operations of the containerbased applications across the cluster of nodes.
- Kubernetes helps to manage containerised applications in various types of physical, virtual, and cloud environments.
- Google Kubernetes is a highly flexible container tool to consistently deliver complex applications running on clusters of hundreds to thousands of individual servers .
- Kubernetes is the Linux kernel which is used for distributed systems.
- It helps you to be abstract the underlying hardware of the nodes(servers) and offers a consistent interface for applications that consume the shared pool of resources

8.Testing

8.1 Test cases

1. Login button click with wrong credentials entered.
2. Signup with already registered mail ID.
3. Signup with wrong form data entered.
4. Entering home page with logged out session.
5. delete expense triggers change in graph.
6. Add expense without choosing category.

8.2 User Acceptance Testing

S. No	Test Case id	Feature Type	Test description	Input test Data	Actual output	Expected output	Remarks
1	TC – RG 01	Functional	Register for application by enteringmy name,email, password, monthly limit	User1 User1@gmail.com ***** 5000	Registration successful	Registration successful	pass
2	TC – SI 01	Functional	Log into the application by entering Email & password	User1@gmail.com *****	Login successful	Login sucessfull	pass
3	TC – ST 01	UI	View myentire expenses throughout a particular period oftime		Expensesare displayed for particular time	Expensesare displayed for particular time	pass
4	TC – DB 01	UI	Display graph in dashboard		Graph is displayed	Graph is displayed	pass
5	TC – ST 02	Functional	Generate reports based on my previous expenditures		Reports generated in graphical form	Reports generated in graphical form	pass
6	TC – SI 02	Functional	Can logout		Go to sign page	Sign in page displayed	pass

7	TC – ST 03	Functional	Create expense	14-11-2022 100 Food	Expenses created	Expenses created	pass
8	TC – ST 04	Functional	Can edit ,delete, update expense		Expenses updated	Updated of expenses	pass
9	TC – ST 05	UI	Can view Credit and debit expenses separately.		Expenses are listed separately	Expenses are listed separately	pass
10	TC – ST 06	UI	Aware of the expense that I spend the most on		Expenses are listed for particular category	Expenses are listed for particular category	pass
11	TC – PG 01	Functional	Able to update my set monthly limit		Monthly limit updated	Monthly limit updated	pass
12	TC – PG 01	UI	Able to View my profile		Profile details displayed	Profile details displayed	pass

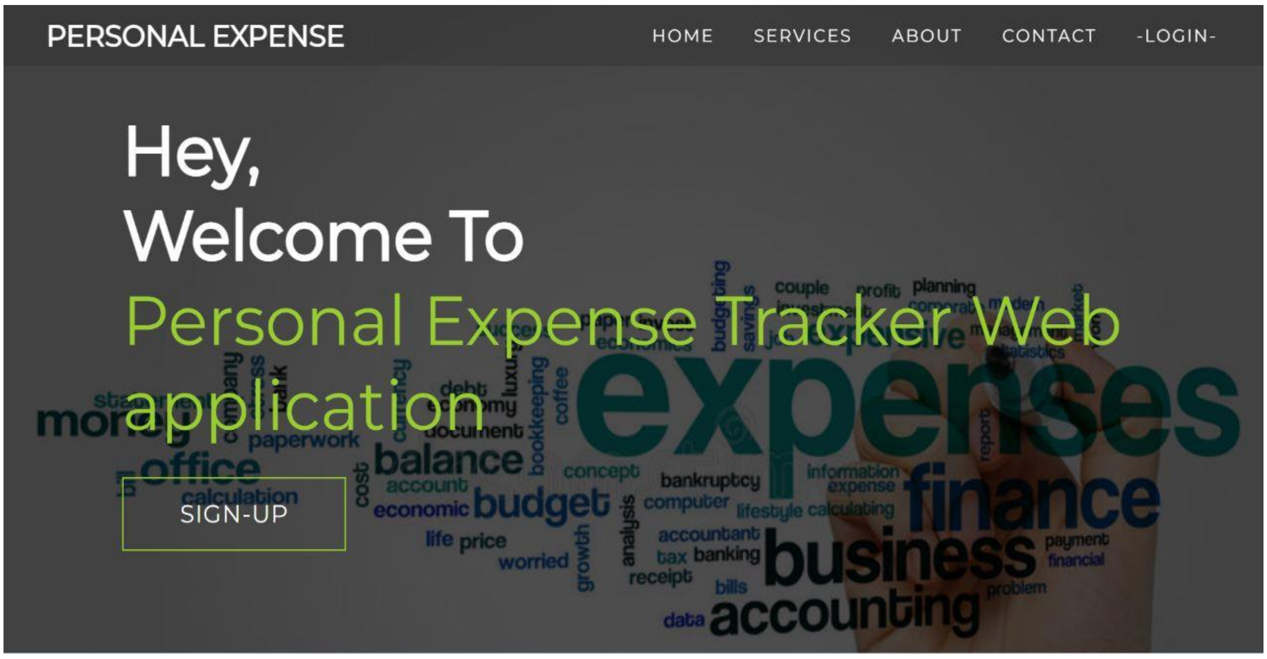
7. Results

7.1 Performance Metrics

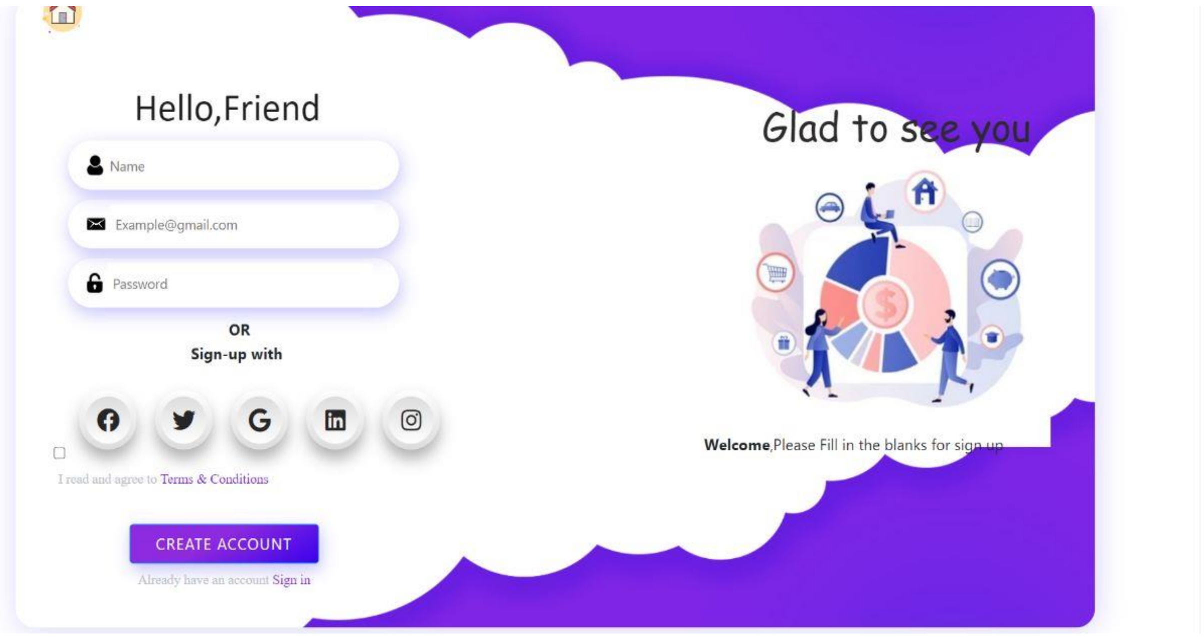
1. Hours worked : 50 hours
2. Stick to Timelines : 100%
3. Consistency of the product : 75%
4. Efficiency of the product : 80%
5. Quality of the product : 85%

Output

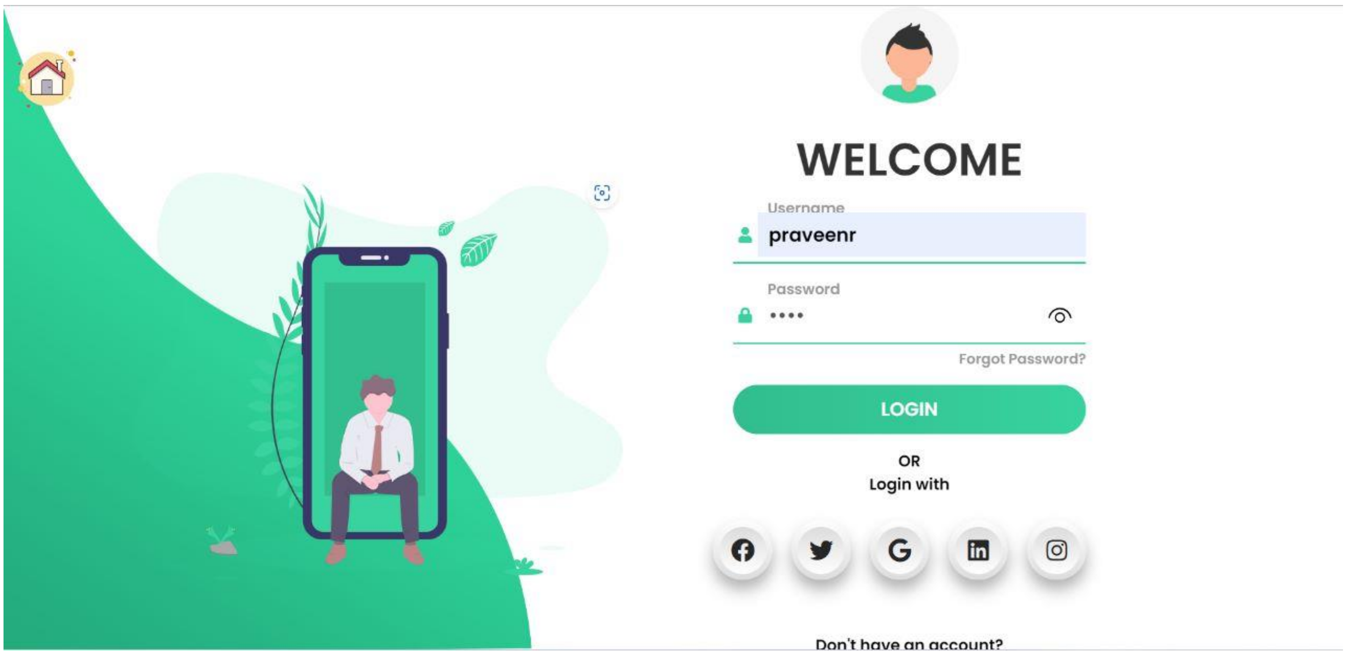
DASHBOARD PAGE



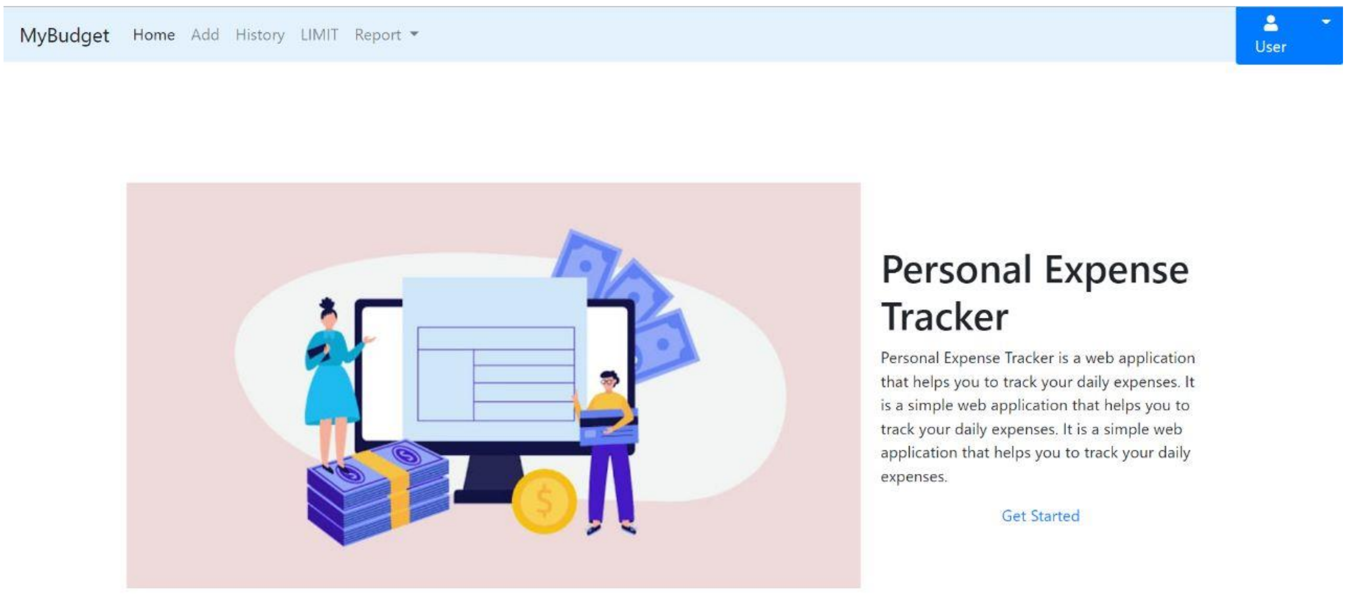
REGISTER PAGE


Hello,Friend
Name
Example@gmail.com
Password
OR
Sign-up with
I read and agree to Terms & Conditions
CREATE ACCOUNT
Already have an account Sign in
Glad to see you
Welcome,Please Fill in the blanks for sign up

LOGIN PAGE



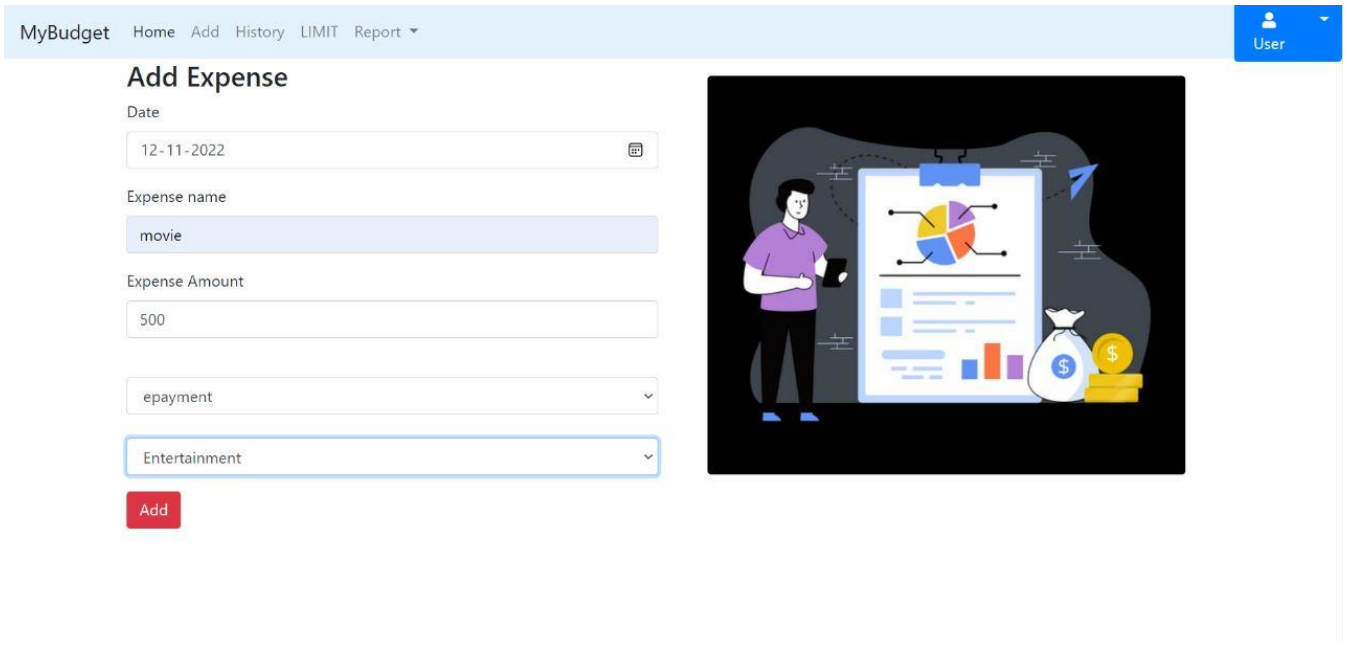
HOME PAGE



Personal Expense Tracker

Personal Expense Tracker is a web application that helps you to track your daily expenses. It is a simple web application that helps you to track your daily expenses. It is a simple web application that helps you to track your daily expenses.

ADD EXPENSE



Add Expense

Date: 12-11-2022

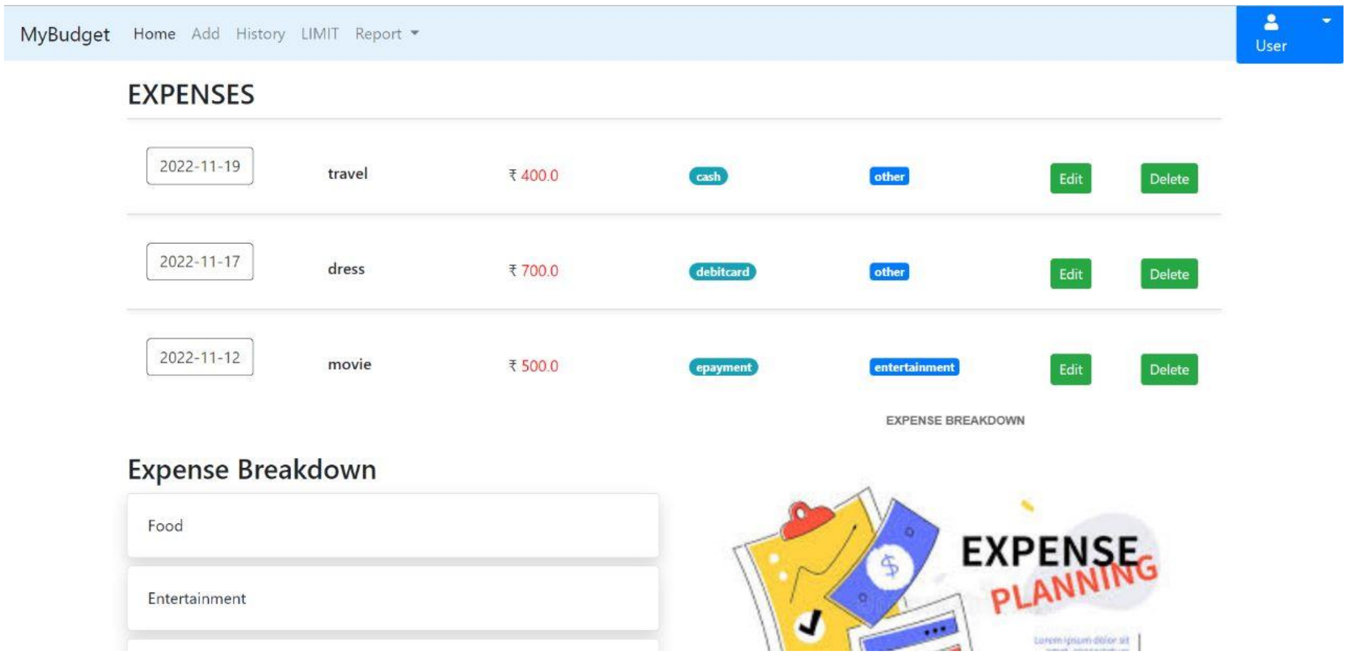
Expense name: movie

Expense Amount: 500

epayment

Entertainment

HISTORY PAGE



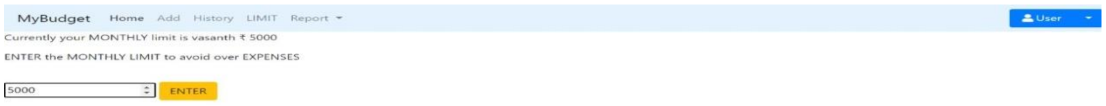
EXPENSES

2022-11-19	travel	₹ 400.0	cash	other	Edit	Delete
2022-11-17	dress	₹ 700.0	debitcard	other	Edit	Delete
2022-11-12	movie	₹ 500.0	epayment	entertainment	Edit	Delete

Expense Breakdown

- Food
- Entertainment

LIMIT PAGE

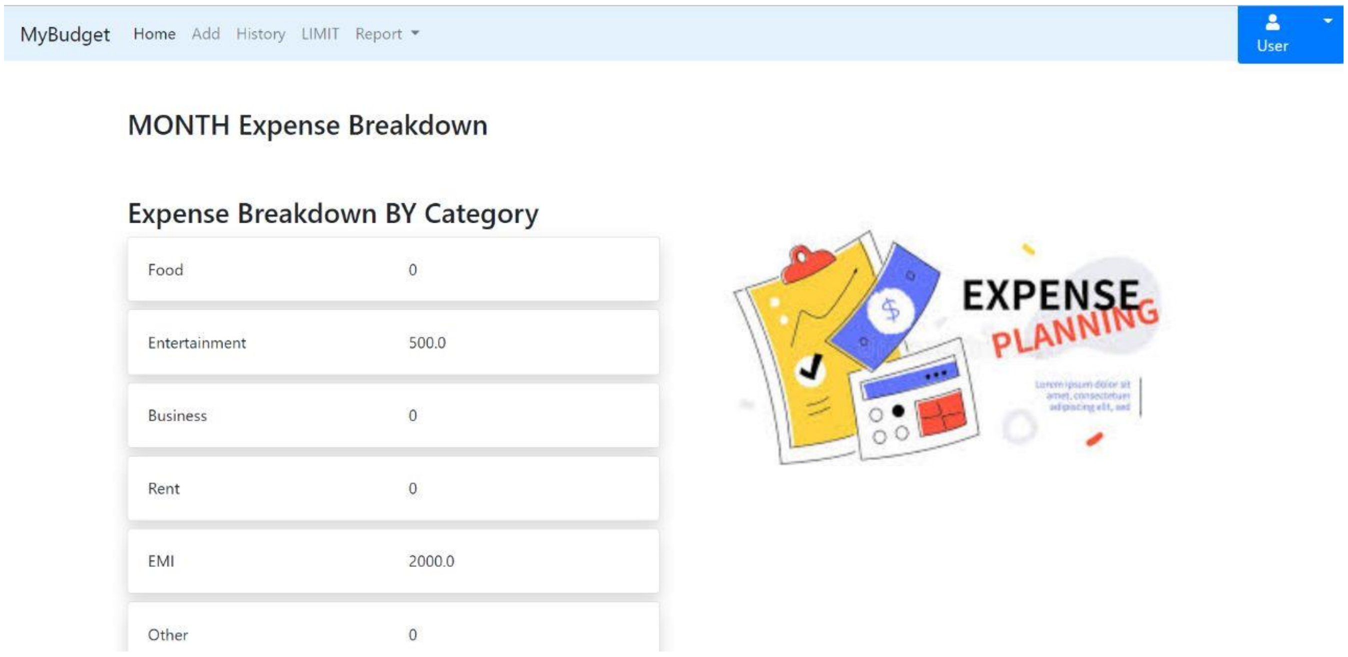


Currently your MONTHLY limit is vasanth ₹ 5000

ENTER the MONTHLY LIMIT to avoid over EXPENSES

5000

MONTHLY REPORT

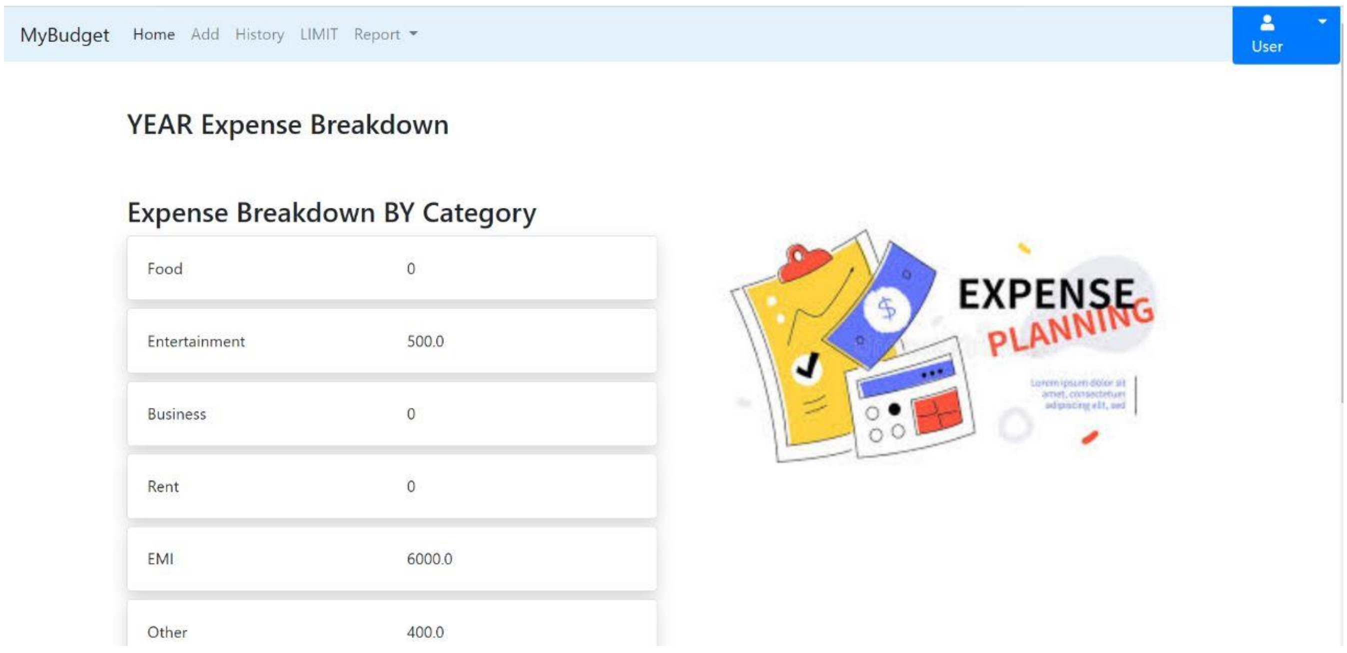


MONTH Expense Breakdown

Expense Breakdown BY Category

Food	0
Entertainment	500.0
Business	0
Rent	0
EMI	2000.0
Other	0

YEARLY REPORT



YEAR Expense Breakdown

Expense Breakdown BY Category

Food	0
Entertainment	500.0
Business	0
Rent	0
EMI	6000.0
Other	400.0

10.Advantages & Disadvantages

Advantages

- Which allows users to track their expenses daily, weekly, monthly, and yearlyinterms of summary, bar graphs, and pie-charts.
- Separate view for credit and debit transactions
- no burden of manual calculations
- generate and save reports.
- You can insert, delete records
- You can track expenses by categories like food, automobile, entertainment,education etc..
- You can track expenses by time, weekly, month, year etc..
- Setting monthly limits and we can update it later Customized email alerts when limit exceeds

Disadvantages

- User have entry every records manually
- The category divided may be blunder or messy
- Can't able to customized user defined categories

11. Conclusion

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

The project personal expense tracker has been successfully implemented by using python, flask, html/css/java script and the database created by using ibm db2 and also successfully executed and implemented.

12. Future Scope

In further days, there will be mails and payment embedded with the app. Also, backup details will be recorded on cloud.

- 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 .
- Alerts for paying dues and remainders to record input at particular userdefined time.
- 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.

13. Appendix

Source code

Home.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <link rel="stylesheet" href="..\static\css\home.css" />
    <title>Personal Expense Tracker</title>
  </head>

  <body>
    <!-- Header -->
    <section id="header">
      <div class="header container">
        <div class="nav-bar">
          <div class="brand">
            <a href="#hero">
              <h1>Personal Expense</h1>
            </a>
          </div>
          <div class="nav-list">
            <div class="hamburger">
              <div class="bar"></div>
            </div>
            <ul>
              <li><a href="#hero" data-after="Home">Home</a></li>
              <li><a href="#services" data-after="Service">Services</a></li>
              <li><a href="#about" data-after="About">About</a></li>
              <li><a href="#contact" data-after="Contact">Contact</a></li>
              <li><a href="/signin" data-after="Login">-Login-</a></li>
            </ul>
          </div>
        </div>
      </div>
    </section>
    <!-- End Header -->
```

```
<!-- Hero Section  -->
<section id="hero">
  <div class="hero container">
    <div>
      <h1>Hey, <span></span></h1>
      <h1>Welcome To <span></span></h1>
      <h1>Personal Expense Tracker Web application <span></span></h1>
      <a href="/signup" type="button" class="cta">Sign-up</a>
    </div>
  </div>
</section>
<!-- End Hero Section  -->
<!-- Service Section -->
<section id="services">
  <div class="services container">
    <div class="service-top">
      <h1 class="section-title">Serv<span>i</span>ces</h1>
      <p>
        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.
      </p>
    </div>
    <div class="service-bottom">
      <div class="service-item">
        <div class="icon">
          <img
            src="https://img.icons8.com/bubbles/100/000000/services.png"
          />
        </div>
        <h2>Efficience</h2>
        <p>
          personal finance entails all the financial decisions and
          activities that a Finance app makes your life easier by helping
          you to manage your finances efficiently.
        </p>
      </div>
      <div class="service-item">
        <div class="icon">
          <img
            src="https://img.icons8.com/bubbles/100/000000/services.png"
          />
        </div>
        <h2>Feature</h2>
        <p>
          A Flask app that users may use on a website to update their daily
          expense and keep track of their spending.And to know personal
          activity.
        </p>
      </div>
      <div class="service-item">
```

```
        <div class="icon">
          <img
            src="https://img.icons8.com/bubbles/100/000000/services.png"
          />
        </div>
        <h2>Personal Expenses</h2>
        <p>
          Budgeting is more than paying bills and setting aside savings.it's
          about creating a money plan for the life you want
        </p>
      </div>
      <div class="service-item">
        <div class="icon">
          <img
            src="https://img.icons8.com/bubbles/100/000000/services.png"
          />
        </div>
        <h2>Financial Life</h2>
        <p>
          Get your Complete financial picture at a glance. With MyBudget
          application you can view your all the financial activities
        </p>
      </div>
    </div>
  </div>
</section>
<!-- End Service Section -->

<!-- About Section -->
<section id="about">
  <div class="about container">
    <div class="col-left">
      <div class="">
        <img
          src="https://tpgit.edu.in/wp-content/uploads/2019/03/tpgit_logo.png"
          alt="img"
        />
        <div><h2></h2></div>
      </div>
    </div>

    <div class="col-right">
      <h1 class="section-title">About <span>Us</span></h1>
      <h2>Category: Cloud App Development</h2>
      <h2>Contributors:</h2>
      <h2>Praveen R</h2>
      <h2>Praveena S</h2>
      <h2>Pavithra T</h2>
      <h2>Rabooni S</h2>
      <p></p>
      <a href="#footer" class="cta">Follow Us</a>
    </div>
  </div>
```

```html
</section>
<!-- End About Section -->

<!-- Contact Section -->
<section id="contact">
  <div class="contact container">
    <div>
      <h1 class="section-title">Contact <span>info</span></h1>
    </div>
    <div class="contact-items">
      <div class="contact-item">
        <div class="icon">
          <img src="https://img.icons8.com/bubbles/100/000000/phone.png" />
        </div>
        <div class="contact-info">
          <h1>Phone</h1>
          <h2>666666</h2>
        </div>
      </div>
      <div class="contact-item">
        <div class="icon">
          <img
            src="https://img.icons8.com/bubbles/100/000000/new-post.png"
          />
        </div>
        <div class="contact-info">
          <h1>Email</h1>
          <h2>expensestracker@gmail.com</h2>
        </div>
      </div>
      <div class="contact-item">
        <div class="icon">
          <img
            src="https://img.icons8.com/bubbles/100/000000/map-marker.png"
          />
        </div>
        <div class="contact-info">
          <h1>Address</h1>
          <h2>Tamil Nadu, India</h2>
        </div>
      </div>
    </div>
  </div>
</section>
<!-- End Contact Section -->

<!-- Footer -->
<section id="footer">
  <div class="footer container">
    <div class="brand">
      <h1>
        <span>P</span>ersonal <span>E</span>xpense <span>T</span>racker
      </h1>
    </div>
```

```
        <div class="social-icon">
          <div class="social-item">
            <a href="#"
              ><img
                src="https://img.icons8.com/bubbles/100/000000/facebook-new.png"
            /></a>
          </div>
          <div class="social-item">
            <a href="#"
              ><img
                src="https://img.icons8.com/bubbles/100/000000/instagram-new.png"
            /></a>
          </div>
          <!-- <div class="social-item">
          <a href="#"><img src="https://img.icons8.com/bubbles/100/000000/twitter.png" /></a>
        </div> -->
          <div class="social-item">
            <a href="#"
              ><img src="https://img.icons8.com/bubbles/100/000000/behance.png"
            /></a>
          </div>
        </div>
        <p>Copyright © PE . All rights reserved</p>
      </div>
    </section>
    <!-- End Footer -->
    <script src="..\static\js\home.js"></script>
  </body>
</html>
```

Login.html

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

        <div class="img">
            <div id="png"><a href="/" title="HOME"><img style="width:75px; height:75px ; "
src="..\static\images\home-page.png"></a></div>
            <img src="..\static\images\bg.svg">
        </div>
        <div class="login-content">

            <form action='/login' method="POST">
                <div class="msg">{{ msg }}</div>
                <img src="..\static\images\avatar.svg">
                <h2 class="title">Welcome</h2>
                <div class="input-div one">
                   <div class="i">
                        <i class="fas fa-user"></i>
                   </div>
                   <div class="div">
                        <h5>Username</h5>
                        <input type="text" name="username" class="input" required>
                   </div>
                </div>
                <div class="input-div pass">
                   <div class="i">
                        <i class="fas fa-lock"></i>
                   </div>
                   <div class="div">
                        <h5>Password</h5>
                        <input type="password"  name="password" class="input" required>
                   </div>
                </div>
                <a href="#">Forgot Password?</a>
                <input type="submit" class="btn" value="Login">
                <span>OR</span>

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

                </div>
                <div class="app" ><b>Don't have an account?</b><a id="app1"
href="\signup">REGISTER.here</a></div>
            </form>

        </div>

    </div>

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

Sign up.html

```
<html>
```

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

   <div id="png"><a href="/" title="HOME"><img style="width:55px; height:55px ; "
src="..\static\images\home-page.png"></a></div>
 <!--heading-->
 <form action="/register" method="post">
   <div class="msg">{{ msg }}</div>
 <h1 class="heading">Hello,Friend</h1>
 <!--name-box-->
 <div class="text">
 <img height="20px" src="..\static\images\user.png" />
 <input placeholder="Name" type="text" name="username"/>
 </div>
 <!--Email-box-->
 <div class="text">
 <img height="12px" src="..\static\images/email.png" />
 <input placeholder=" Example@gmail.com" type="email" name="email"" />
 </div>
 <!--Password-box-->
 <div class="text">
 <img height="20px" src="..\static\images\password.png" />
 <input placeholder=" Password" type="password" name="password"/>
 </div>
<div class="or"><b>OR</b></div>
<div class="s1"><p><b>Sign-up with</b></p></div>

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

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

<div class="terms">
   <input class="check" type="checkbox" required/>
 <p class="conditions">I read and agree to <a href="#">Terms &amp; Conditions</a></p>
 </div>
<!--button-->
<div class="toop">
<button type="submit" class="btn btn-primary" >CREATE ACCOUNT</button> </div>

</form>
<!--sign-in-->
<div class="t"><p class="conditions" id="p3">Already have an account <a href="/signin">Sign
in</a></p> </div></div>
 </div>
<!--text-container-->
<div class="text-container">

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

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

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

App.py

```
from flask import Flask, render_template, request, redirect, session
import ibm_db
import ibm_db_dbi
import re

app = Flask(__name__)


app.secret_key = 'a'

# conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=19af6446-6171-4641-8aba-
9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=30699;SECURITY=SSL;SSLS
erverCertificate=DigiCertGlobalRootCA.crt;UID=mbs46040;PWD=MIEpZ1DoqwMRpGvs",",")
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=6667d8e9-9d4d-4ccb-ba32-
21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=30376;SECURITY=SSL;SSL
ServerCertificate=DigiCertGlobalRootCA.crt;UID=nxp94278;PWD=XdzphucSHe8uQRe1", '', '')
connection = ibm_db_dbi.Connection(conn)
cursor = connection.cursor()

cursor.execute('''CREATE TABLE IF NOT EXISTS expenses (
    id VARCHAR(50) NOT NULL,
    date DATE NOT NULL,
    expensename VARCHAR(50) NOT NULL,
    amount FLOAT NOT NULL,
    paymode VARCHAR(50) NOT NULL,
    category VARCHAR(50) NOT NULL
    )''')
# HOME--PAGE
```

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


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


# SIGN--UP--OR--REGISTER


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


@app.route('/register', methods=['GET', 'POST'])
def register():
    msg = ''
    if request.method == 'POST':
        username = request.form['username']
        email = request.form['email']
        password = request.form['password']

        # cursor = mysql.connection.cursor()
        cursor.execute(
            "SELECT * FROM register WHERE username = ?", (username, ))
        account = cursor.fetchone()
        print(account)
        if account:
            msg = 'Account already exists !'
        elif not re.match(r'[^@]+@[^@]+\.[^@]+', email):
            msg = 'Invalid email address !'
        elif not re.match(r'[A-Za-z0-9]+', username):
            msg = 'name must contain only characters and numbers !'
        else:
            cursor.execute("INSERT INTO register VALUES ( ?, ?, ?)",
                           (username, email, password))
            connection.commit()
            msg = 'You have successfully registered !'
            return render_template('signup.html', msg=msg)

 # LOGIN--PAGE


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


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

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

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

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


# ADDING----DATA


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


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

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

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

    return redirect("/display")


# DISPLAY---graph
@app.route("/display")
def display():
    print(session["username"], session['id'])
    # cursor = mysql.connection.cursor()
    cursor.execute(
        'SELECT * FROM expenses WHERE id = ? ORDER BY date DESC', (session['id'],))
    expense = cursor.fetchall()

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


# delete---the--data

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


# UPDATE---DATA

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

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


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

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

    #   cursor = mysql.connection.cursor()
    cursor.execute("UPDATE 'expenses' SET 'date' = ? , 'expensename' = ? , 'amount' = ?, 'paymode' = ?,
'category' = ? WHERE 'expenses'.'id' = ? ",
                       (date, expensename, amount, str(paymode), str(category), session['id']))
        connection.commit()
        print('successfully updated')
        return redirect("/display")

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

```
@app.route("/limitnum", methods=['POST'])
def limitnum():
    if request.method == "POST":
        number = request.form['number']
        #  cursor = mysql.connection.cursor()
        cursor.execute("INSERT INTO limits VALUES (?, ?) ",
                       (session['id'], number))
        connection.commit()
        return redirect('/limitn')


@app.route("/limitn")
def limitn():
    # cursor = mysql.connection.cursor()
    # cursor.execute(
    #     "SELECT * FROM limits WHERE ID = ? AND ORDER BY id  DESC", (session['id']))

    cursor.execute(
        "SELECT * FROM limits where id=?", (session['id'],))

    x = cursor.fetchone()
    n = x[0]
    s = x[1]
    print(s)

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

# REPORT


@app.route("/today")
def today():
    #   cursor = mysql.connection.cursor()
    print("HI")

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

    expense = cursor.fetchall()

    total = 0
    t_food = 0
    t_entertainment = 0
    t_business = 0
    t_rent = 0
    t_EMI = 0
    t_other = 0
for x in expense:
        print(x[3])
        total += x[3]
        if x[5] == "food":
            t_food += x[3]

        elif x[5] == "entertainment":
            t_entertainment += x[3]

        elif x[5] == "business":
            t_business += x[3]
        elif x[5] == "rent":
            t_rent += x[3]

        elif x[5] == "EMI":
            t_EMI += x[3]

        elif x[5] == "other":
            t_other += x[3]

    print(total)

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

    return render_template("today.html", expense=expense,  total=total,
                           t_food=t_food, t_entertainment=t_entertainment,
                           t_business=t_business,  t_rent=t_rent,
                           t_EMI=t_EMI,  t_other=t_other)
@app.route("/month")
def month():
    #   cursor = mysql.connection.cursor()
    #   cursor.execute("SELECT DATE(date), SUM(amount) FROM expenses WHERE userid= ? AND
MONTH(DATE(date))= MONTH(now()) GROUP BY DATE(date) ORDER BY DATE(date)
",(str(session['id'])))
    #   texpense = cursor.fetchall()
    #   print(texpense)
    #   cursor = mysql.connection.cursor()
    #   cursor.execute("SELECT * FROM expenses WHERE userid = ? AND MONTH(DATE(date))=
MONTH(now()) AND date ORDER BY `expenses`.`date` DESC",(str(session['id'])))
    cursor.execute(
        "SELECT * FROM EXPENSES WHERE ID = ? AND DATE <=
THIS_MONTH(CURRENT_DATE + 1 MONTH) AND DATE > THIS_MONTH(CURRENT_DATE)
", (session['id'],))
    expense = cursor.fetchall()
    print(expense)
    total = 0



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

    for x in expense:
        total += x[3]
        if x[5] == "food":
            t_food += x[3]

        elif x[5] == "entertainment":
            t_entertainment += x[3]

        elif x[5] == "business":
            t_business += x[3]
        elif x[5] == "rent":
            t_rent += x[3]

        elif x[5] == "EMI":
            t_EMI += x[3]

        elif x[5] == "other":
            t_other += x[3]

    print(total)

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

    return render_template("month.html", expense=expense,  total=total,
                           t_food=t_food, t_entertainment=t_entertainment,
                           t_business=t_business,  t_rent=t_rent,
                           t_EMI=t_EMI,  t_other=t_other)
@app.route("/year")
def year():
    #   cursor = mysql.connection.cursor()
    #   cursor.execute("SELECT MONTH(date), SUM(amount) FROM expenses WHERE userid= ?
AND YEAR(DATE(date))= YEAR(now()) GROUP BY MONTH(date) ORDER BY MONTH(date)
",(str(session['id'])))
    #   texpense = cursor.fetchall()
    #   print(texpense)
  #   cursor = mysql.connection.cursor()
    cursor.execute(
        "SELECT * FROM EXPENSES WHERE ID = ? AND DATE <=
THIS_YEAR(CURRENT_DATE + 1 YEAR) AND DATE > THIS_YEAR(CURRENT_DATE) ",
(session['id'],))
    expense = cursor.fetchall()


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

    for x in expense:
        total += x[3]
        if x[5] == "food":
            t_food += x[3]

        elif x[5] == "entertainment":
            t_entertainment += x[3]

        elif x[5] == "business":
            t_business += x[3]
        elif x[5] == "rent":
            t_rent += x[3]

        elif x[5] == "EMI":
            t_EMI += x[3]

        elif x[5] == "other":
            t_other += x[3]

    print(total)

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

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

# log-out
@app.route('/logout')
def logout():
    session.pop('loggedin', None)
    session.pop('id', None)
    session.pop('username', None)
    return render_template('home.html')
if __name__ == "__main__":
    app.run(debug=True)
```

Gitup & Project demo link:

Github Link : https://github.com/IBM-EPBL/IBM-Project-21065-1659771679

Project Demonstration Link: DemoVideo