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

TEAM ID : PNT2022TMID11437

Personal Expense Tracker Application

A PROJECT REPORT

KARTHIKEYAN R(910619104038)

LAURENCE A (910619104041)

MOHAMMED ARAFATH M (910619104047)

NANDHA KRISHNA V T (910619104051)

BACHELOR OF ENGINEERING IN COMPUTER SCIENCE AND ENGINEERING

K.L.N COLLEGE OF ENGINEERING

POTTAPALAYAM - 630612

1.INTRODUCTION

1.1 Project Overview:

Personal Expense Tracker 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 through computerized system which tends to eliminate manual paper works. It will also manage records in systematic way and user can access the stored data conveniently. We have tried to design the project in such way that user may not have any difficulty in using this application without much effort. This software can be really used by end user who has stable internet. The language that we use to develop this system is flask using python and IBMDb2 for database.

1.2 Purpose:

Expense Tracker is an Application which can help the user to keeptrack of their Expenses. Now a days, people can do various things by using a mobile and so, they can also use it for Budgeting and planning their expense in the mobile instead of doing it manually.For this purpose, an application can be developed to satisfy the needs of the customer. This application can help the user to keep track of their expenses in an organized way and to maintain a proper balance between expenditure and savings.

The idea of developing this project for user convenience. Because whenever they make expenses immediately, they add in theapplication. Some of the concerns maintaining a personal expense is aBIG problem, in daily expenses many times we don’t know where themoney goes. Some of the conventional methods used to tackle this problem in normal circumstances are like making use of as ticky notes by common users, Proficient people deals with this kind of problems by using spreadsheets to record expense and using a ledger to maintains the large amounts data by especially by expert people. We believe a handy design and a handy mobile application which handles thesetroubles.Such that appiscapable of recording the expenditure and giving broad view with easy to use the user interface and this application is intelligent enough to shows the history of expenses.

2.LITERATURESURVEY

2.1 Existing Problem

Shahed Anzarus Sabab, Sadman Saumik Islam, Md. Jewel Rana, Monir Hossain Department of Computer Science and Engineering (CSE) Northern University Bangladesh, Daffodil International University ,Dhaka, Bangladesh 4th International Conference of on Electrical Engineering and Information and Communication Technology, 2018[1]

eExpense is an application that supports Android smartphones. By using this application, users can save their expense by simply scanning the bills or receipt copies. This application extracts the textual information from the receipts and saves the amount and description for further processing. It also monitors user’s income by tracking the received SMS’s from the user’s saving accounts. By calculating income and expense it produces the user’s balance on a monthly and yearly basis. Overall, this is a smart computerized solution for tracking expenses.

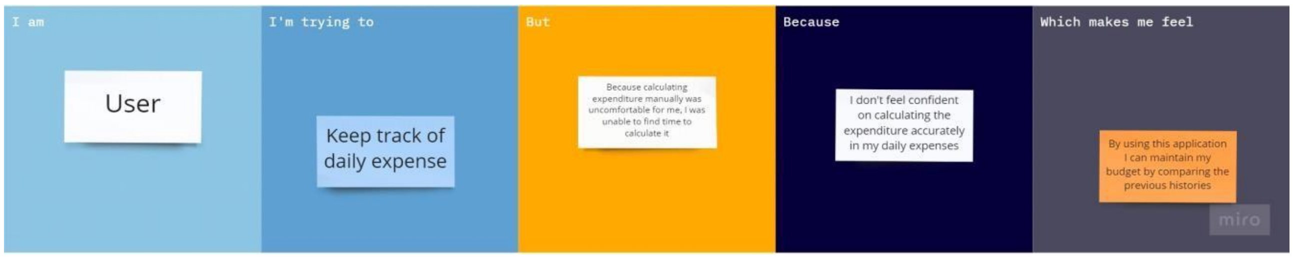
2.2 References:

Published by: P. Thanapal*, Mohammed Yaseen Patel, T.P.Lokesh Raj and J. Satheesh Kumar

Indian Journal of Science and Technology, Vol 8(S2), 118–122, January 2015[2]

Published by: Muskaan Sharma, Ayush Bansal , Dr. Raju Ranjan , Shivam Sethi June 2021, IJIRT , Volume 8 Issue 1 , ISSN: 2349-6002

2.3 Problem Statement


I am
User
I'm trying to
Keep track of daily expense
But
Because calculating expenditure manually was uncomfortable for me, I was unable to find time to calculate it
Because
I don't feel confident on calculating the expenditure accurately in my daily expenses
Which makes me feel
By using this application I can maintain my budget by comparing the previous histories
miro

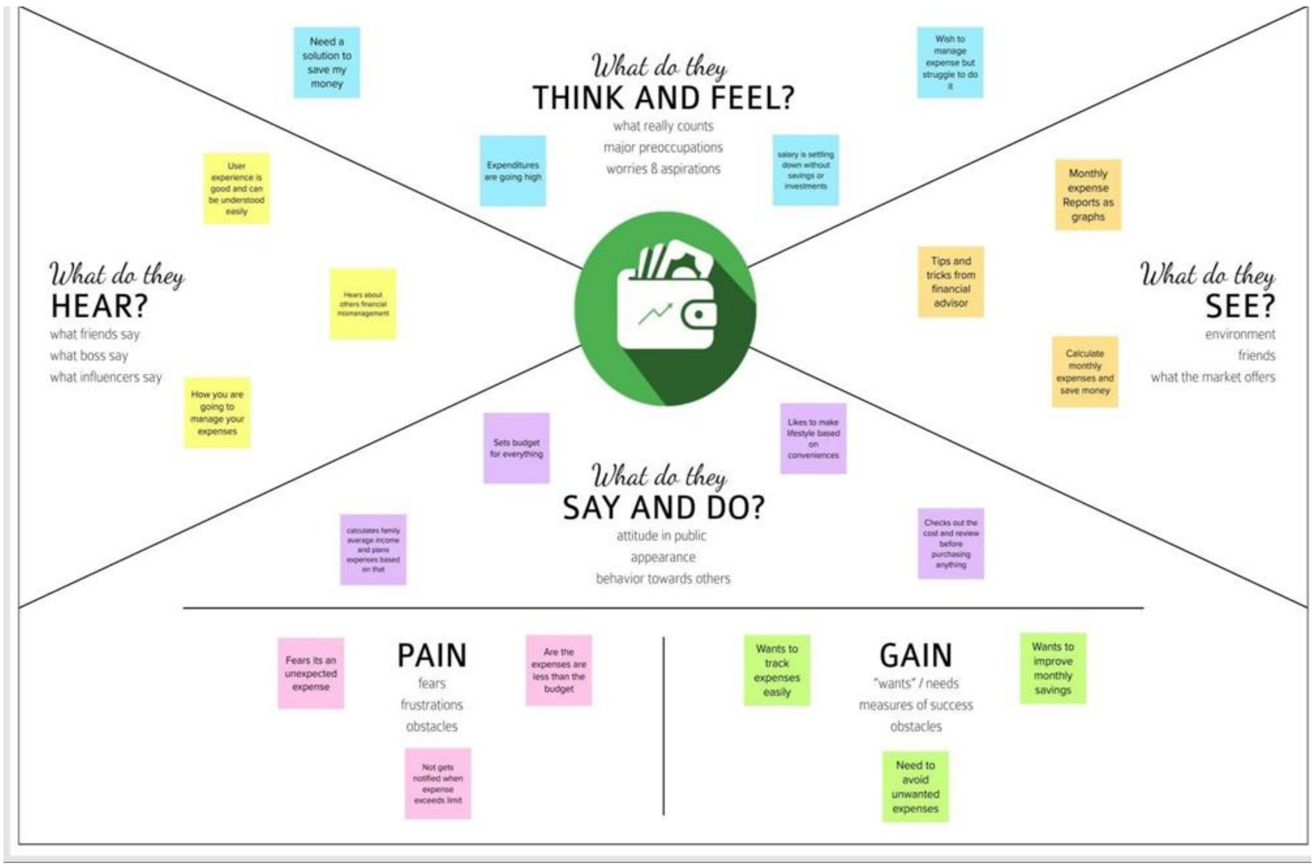
Problem Statement (PS)	**I am(Customer)**	**I'm tryingto**	**But**	**Because**	**Which makes me feel**
PS-1	User	Keep track of daily expense	Because calculating expenditure manually was uncomfortable for me, I was unable to find time to calculate it	I don't feel confident on calculating the expenditure accurately in my daily expenses	By using this application I can maintain my budget by comparing the previous histories
PS-2	user	limit my spending	can keep track of it	I have multiples mall spendings and I am forgetful	Irresponsible

3. IDEATION&PROPOSEDSOLUTION

3.1Empathymapcanvas:

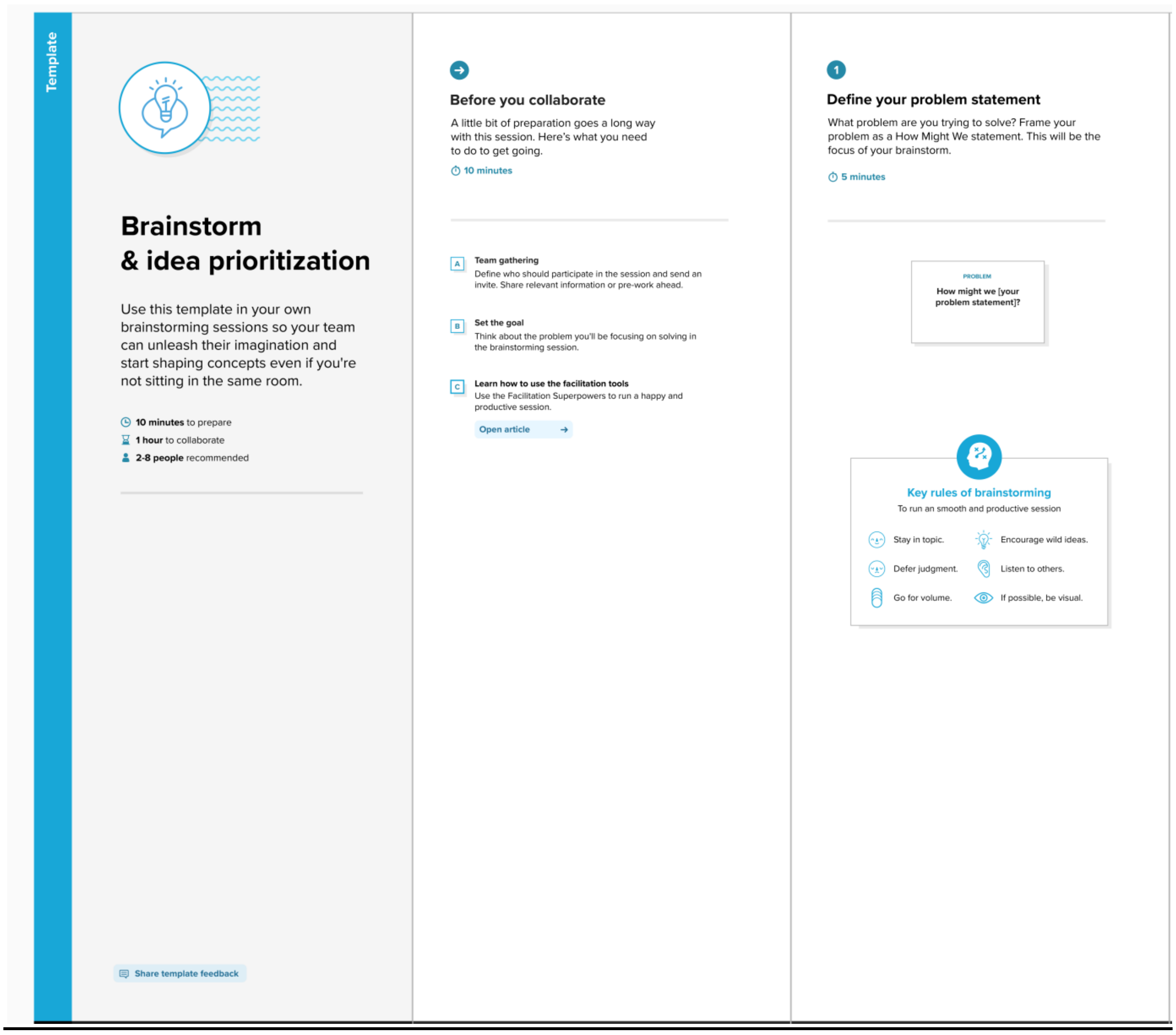
An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes.

It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges


What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
Need a solution to save my money
Expenditures are going high
Wish to manage expense but struggle to do it
salary is settling down without savings or investments
What do they HEAR?
what friends say
what boss say
what influencers say
User experience is good and can be understood easily
Hears about others financial mismanagement
How you are going to manage your expenses
What do they SEE?
environment
friends
what the market offers
Monthly expense Reports as graphs
Tips and tricks from financial advisor
Calculate monthly expenses and save money
What do they SAY AND DO?
attitude in public
appearance
behavior towards others
Sets budget for everything
calculates family average income and plans expenses based on that
Likes to make lifestyle based on conveniences
Checks out the cost and review before purchasing anything
PAIN
fears
frustrations
obstacles
Fears its an unexpected expense
Are the expenses are less than the budget
Not gets notified when expense exceeds limit
GAIN
"wants" / needs
measures of success
obstacles
Wants to track expenses easily
Wants to improve monthly savings
Need to avoid unwanted expenses

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.

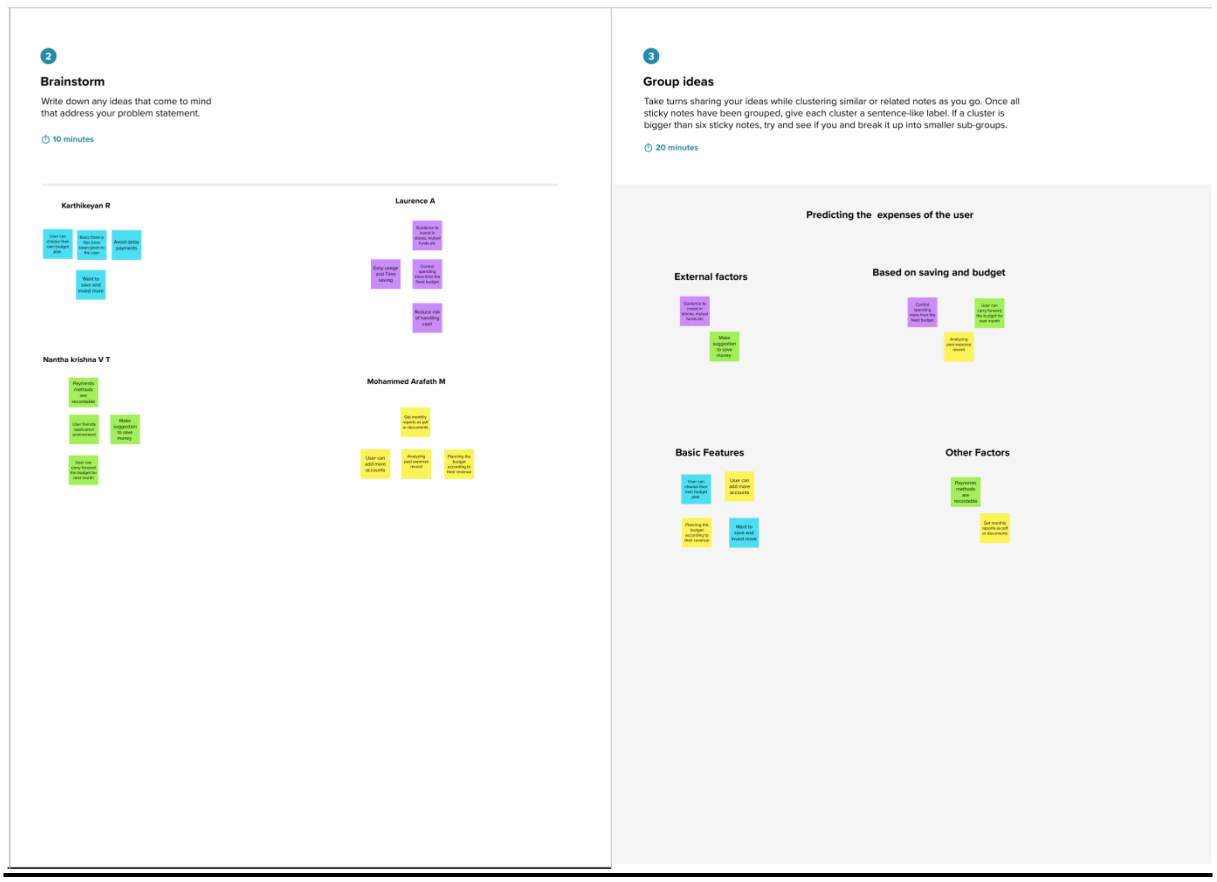
5 minutes

PROBLEM
How might we [your problem statement]?

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 and Grouping



2

Brainstorm

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

10 minutes

Karthikeyan R

User can choose their own budget plan

Basic finance tips have been given to the user

Avoid delay payments

Want to save and invest more

Laurence A

Guidance to invest in stocks, mutual funds etc

Easy usage and Time saving

Control spending more than the fixed budget

Reduce risk of handling cash

Nantha krishna V T

Payments methods are recordable

User friendly application environment

Make suggestion to save money

User can carry forward the budget for next month

Mohammed Arafath M

Get monthly reports as pdf or documents

User can add more accounts

Analyzing past expense record

Planning the budget according to their revenue

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

Predicting the expenses of the user

External factors

Guidance to invest in stocks, mutual funds etc

Make suggestion to save money

Based on saving and budget

Control spending more than the fixed budget

User can carry forward the budget for next month

Analyzing past expense record

Basic Features

User can choose their own budget plan

User can add more accounts

Planning the budget according to their revenue

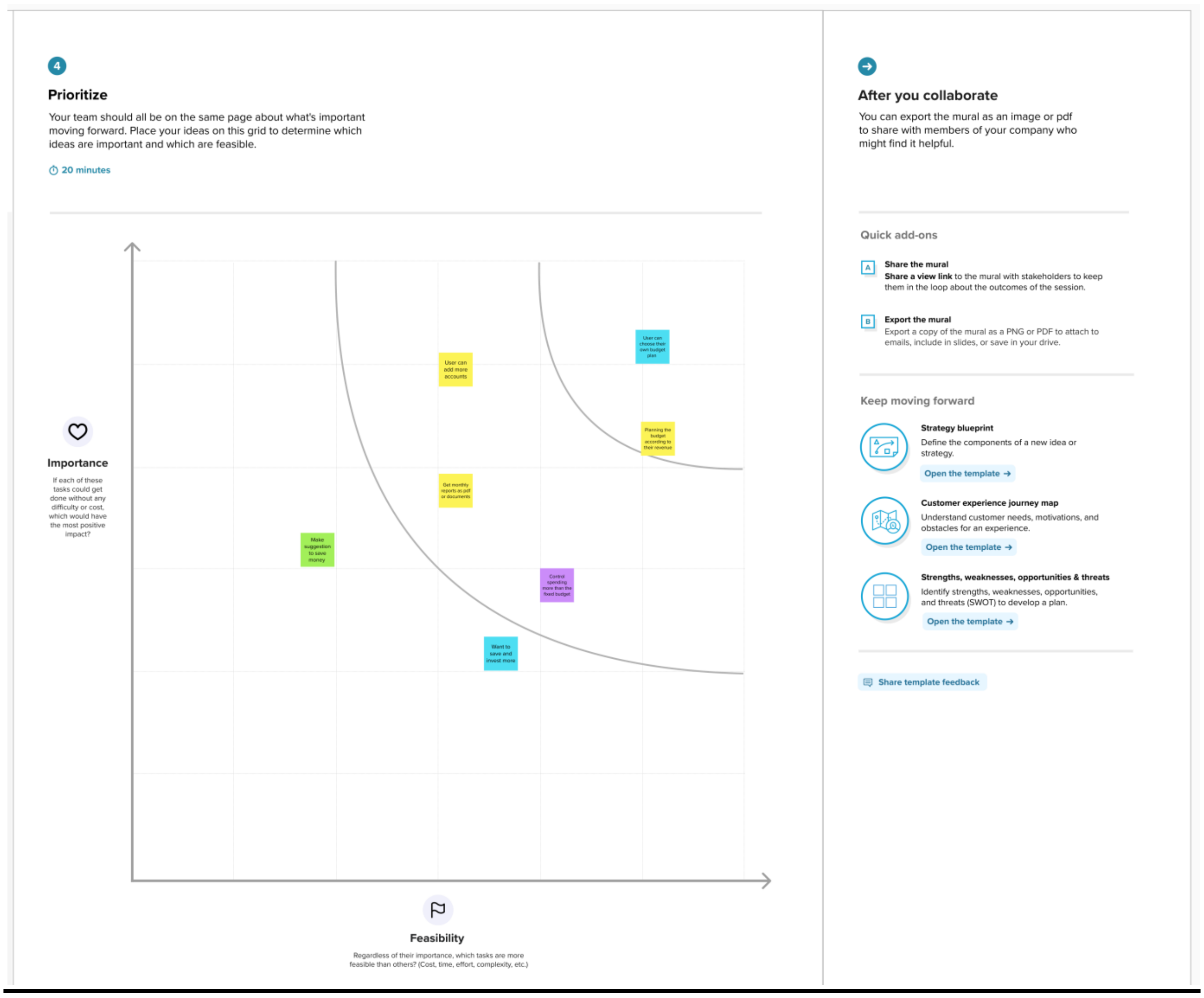
Want to save and invest more

Other Factors

Payments methods are recordable

Get monthly reports as pdf or documents

Step 3: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

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.)

User can choose their own budget plan
User can add more accounts
Planning the budget according to their revenue
Get monthly reports as pdf or documents
Make suggestion to save money
Control spending more than the fixed budget
Want to save and invest more

After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

A **Share the mural**
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

B **Export the mural**
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint
Define the components of a new idea or strategy.
Open the template →

Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
Open the template →

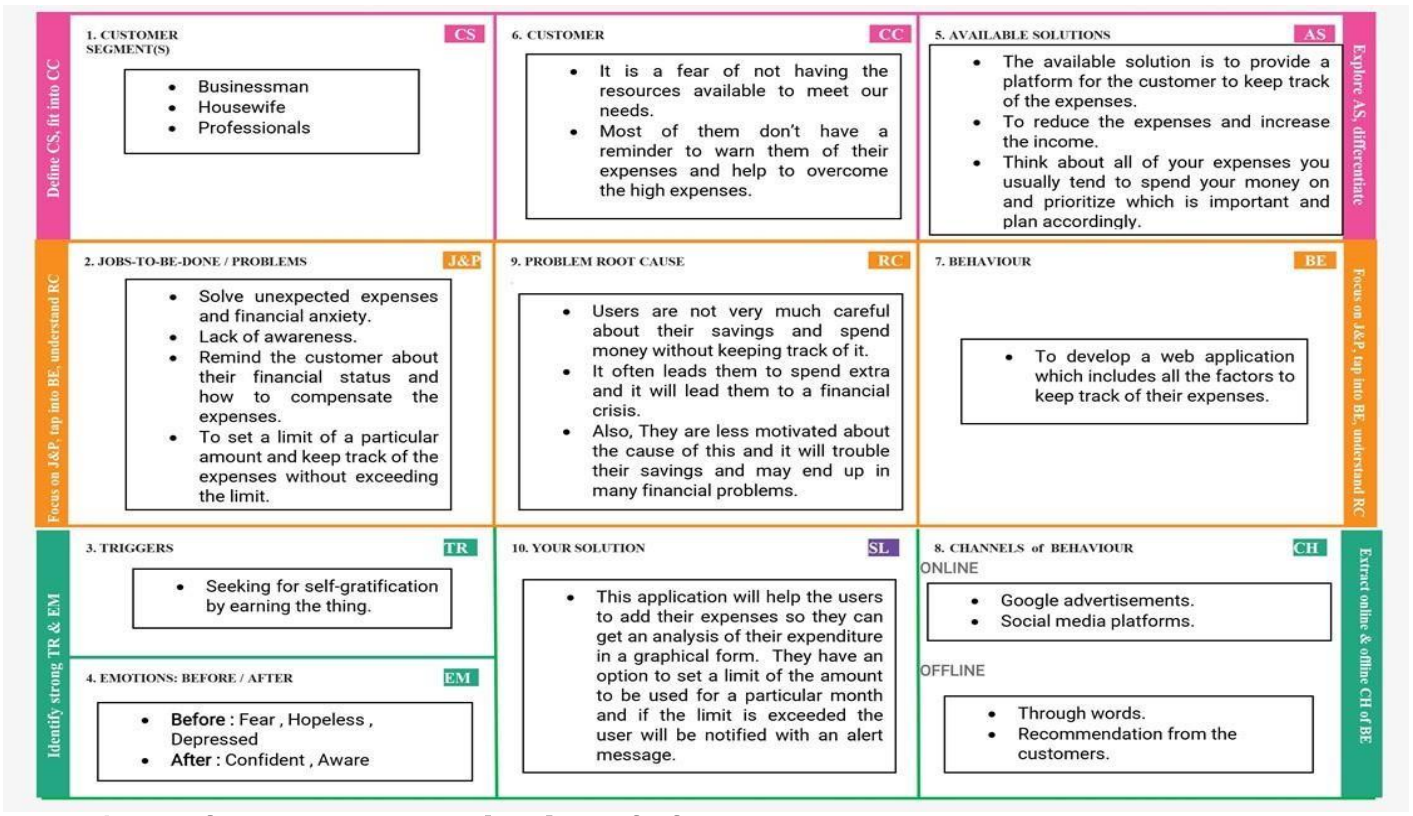
Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template →

Share template feedback

3.3 Proposed Solution:

S.No.	**Parameter**	**Description**
1.	Problem Statement (Problem to be solved)	○ To keep track of our expenses using Cloud Computing.
2.	Idea/Solution description	○ Most of the people are not aware of their expenses and often get stuck with a financia lcrisis. ○ To overcome the financial problems and make a budget according to the salary, Our project helps them to keep track of their daily expenses and provide a monthly record of their expenses in a graphical representation. ○ I twill help the user to know where it Went all wrong and how to overcome the financial problems.
3.	Novelty/Uniqueness	○ We help the customers to keep track ofthe expenses and also we alert themwhich expense to be reduced and chartthe expense in a monthly basis so theycanknoweither theyare benefitedor Not from ourapplication.
4.	Social Impact/ Customer Satisfaction	○ By using our application, Customer canknow where their money is going andthey can save the money by creating abudgetfortheamountthey have and Use accordingly.
5.	Business Model(Revenue Model)	○ Saving money with the help of an application makes our idea realistic.A sit is useful who cares about their money,it can attract the customers as well.
6.	Scalability of the Solution	○ What ever the expense, The application provides a clear chart of their expense and help them create a budget. ○ Even a large scale businessman can also use our application and keep track of his expenditure.

3.4 Proposed Solutionfit:



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

- Businessman
- Housewife
- Professionals

6. CUSTOMER CC

- It is a fear of not having the resources available to meet our needs.
- Most of them don't have a reminder to warn them of their expenses and help to overcome the high expenses.

5. AVAILABLE SOLUTIONS AS

- The available solution is to provide a platform for the customer to keep track of the expenses.
- To reduce the expenses and increase the income.
- Think about all of your expenses you usually tend to spend your money on and prioritize which is important and plan accordingly.

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

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

- Solve unexpected expenses and financial anxiety.
- Lack of awareness.
- Remind the customer about their financial status and how to compensate the expenses.
- To set a limit of a particular amount and keep track of the expenses without exceeding the limit.

9. PROBLEM ROOT CAUSE RC

- Users are not very much careful about their savings and spend money without keeping track of it.
- It often leads them to spend extra and it will lead them to a financial crisis.
- Also, They are less motivated about the cause of this and it will trouble their savings and may end up in many financial problems.

7. BEHAVIOUR BE

- To develop a web application which includes all the factors to keep track of their expenses.

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TR

- Seeking for self-gratification by earning the thing.

4. EMOTIONS: BEFORE / AFTER EM

- **Before** : Fear , Hopeless , Depressed
- **After** : Confident , Aware

10. YOUR SOLUTION SL

- This application will help the users to add their expenses so they can get an analysis of their expenditure in a graphical form. They have an option to set a limit of the amount to be used for a particular month and if the limit is exceeded the user will be notified with an alert message.

8. CHANNELS of BEHAVIOUR CH

ONLINE

- Google advertisements.
- Social media platforms.

OFFLINE

- Through words.
- Recommendation from the customers.

Extract online & offline CH of BE

4. REQUIREMENT ANALYSIS

Requirements analysis, also called requirements engineering, is the process of determining user expectations for a new or modified product. These features ,called requirements, must be quantifiable, relevant and detailed. In software engineering, such requirements are often called functional specifications. Requirements analysis is an important aspect of project management.

4.1 Functional requirement

4.2 Non-Functional requirements

4.1 Functional Requirements:

Following are the functional requirements of the proposed solution.

SINo.	Functional Requirement(Epic)	Sub Requirement(Story/Sub-Task)
1	User Registration	Registration is done through email.
2	User Confirmation	Confirmation via Email with OTP.
3	User Login	By entering username and password.
4	Enter your expenses page.	Save User's expenses with date and time

5	Expenses Report is generated.	Represent all user's data in graphical form for easy understanding of report.
6	Option to add categories and Type of expense to the data	The app can organize expenses based on different categories.
7	Export the Report generated	Print or Save Report as Pdf or Word Document.

4.2 Non-functional Requirements:

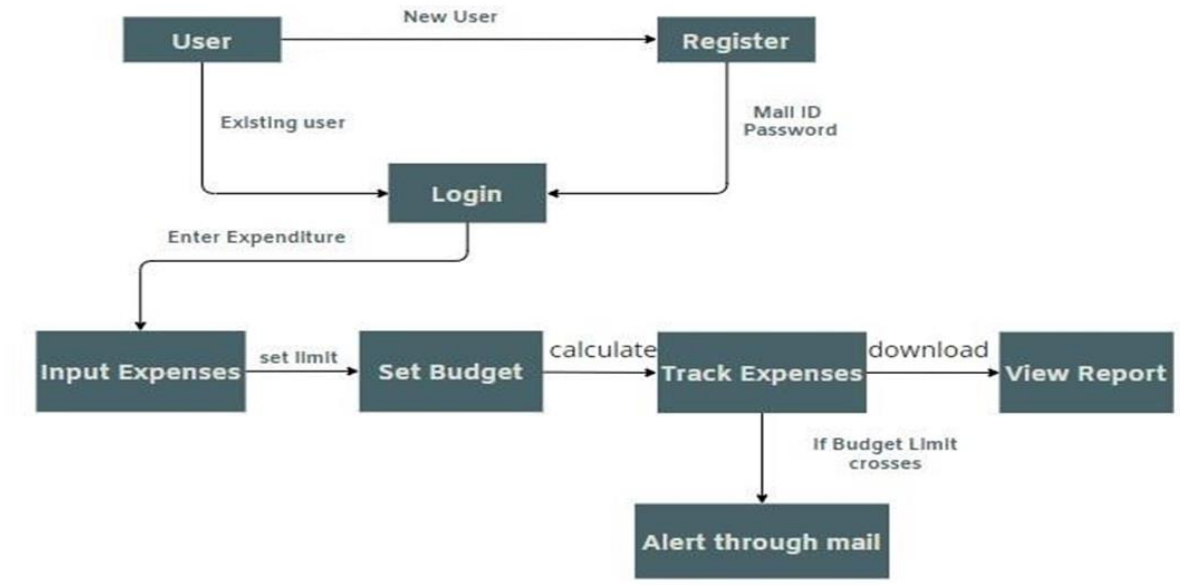
Following are the non-functional requirements of the proposed solution.

SI No.	Non-Functional Requirement	Description
1	**Usability**	Provides an effective and user-friendly way to keep track of all the users expenses.
2	**Security**	Data is protected by giving a unique login ID and password.
3	**Reliability**	Since the app is hosted on the web it can be Accessed anytime and from anywhere on all devices if you have a device with internet connectivity.
4	**Performance**	User data is stored in a very data efficient way using cloud which reduces load time of the application.
5	**Availability**	Application is hosted on the web and should be Available 24/7 for the user.
6	**Scalability**	Can be scaled by increasing database size and better UI design to suit a larger audience as we are using cloud.

5 PROJECT DESIGN

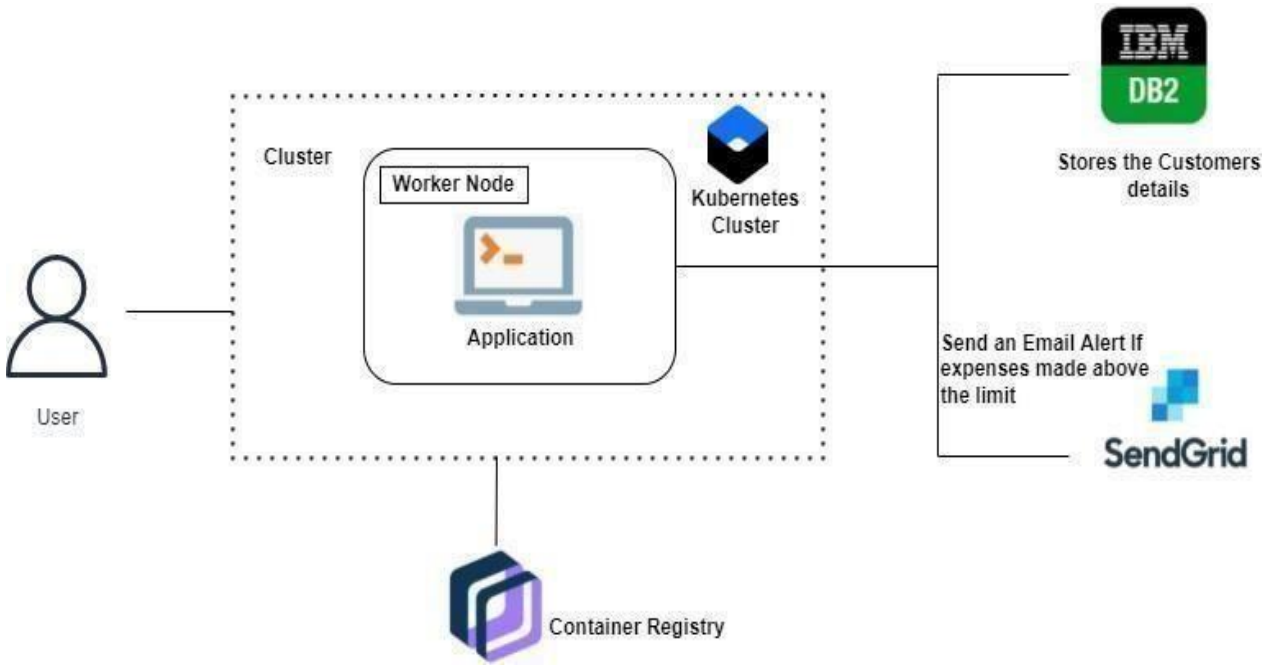
5.1 Data Flow Diagram:

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



5.2 Solution & Technical Architecture

The Deliverable shall include the architectural diagram as below and the information as per the table1&table2



S.No.	**Component**	**Description**	**Technology**
1.	User Interface	The user can Interact with the application with use of Chat bot	HTML, CSS, JavaScript /AngularJs/ReactJ setc.
2.	Application Logic-1	The application contains the sign in/sign up where the user will login into the main dashboard	Java/Python
3.	Application Logic-2	Dashboard contains the fields like Add income, Add Expenses, Save Money	IBM Watson STT service
4.	Application Logic-3	The user will get the expense report in the graph form and also get alerts if the expense limit exceeds	IBM Watson Assistant ,Send Grid
5.	Database	The Income and Expense data are stored in the My SQL database	My SQL ,No SQL, etc.
6.	Cloud Database	With use of Database Service on Cloud, the User data are stored in a well secured Manner	IBMDB2, IBMCloudantetc .

7.	File Storage	IBM Block Storage used to store the Financial data of the user	IBM Block Storage or Other Storage Service or Local File system

Table-2: Application Characteristics:

S. No.	**Characteristics**	**Description**	**Technology**
1.	Open-Source Frameworks	Flask Framework in Python is used to implement this Application	Python-Flask
2.	Security Implementations	This Application Provides high security to the user Financial data. It can be done by using the Container Registry in IBM cloud	Container Registry, Kuber netes Cluster
3.	ScalableArchitecture	Expense Tracker is a life time accesss application.It's demand will increase when the user's income are high	ContainerRegistry,KubernetesCluster
4.	Availability	This application will be available to the user at any part of time	Container Registry,KubernetesCluster
5.	Performance	The performance will be high because there will be none work traffics in the application	Kubernetes Cluster

5.3.**User Stories**

User Type	**Functional Requirement (Epic)**	**User Story Number**	**User Story/Task**	**Acceptance criteria**	**Priority**	**Release**
Customer(Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account/dashboard.	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm.	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook.	I can register & access the dashboard with Facebook Login.	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can register for the app through Gmail login.	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can register &access the dashboard with Gmail Login.	High	Sprint-1
	Dashboard	USN-6	As a user, I can add my day-to-day expenses regularly.	I can track my expenses perfectly.	High	Sprint-2
Customer (Webuser)	Dashboard	USN-7	As a user, I can see login page and registration Page for which the user logins and input expenses.	I can login through Gmail And register for expense tracking.	Medium	Sprint-2
Customer CareExecutive	Dashboard	USN-8	As a customer care executive , I can solve the queries of users.	I can reply to their queries and solve their problems.	High	Sprint-3
Administrator	Registration	USN-9	As an Administrator , I can view the basic details of user.	I can provide the login details.	Medium	Sprint-4
	Dashboard	USN-10	As an administrator, I can able to view the over all progress of a user.	I can give rewards based on their progress.	Low	Sprint-4

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	A literature review is a comprehensive summary of previous research on a topic. The collection/gathering of relevant information based on our project use case and by referring the existing solutions etc.	19 SEPTEMBER 2022
Prepare Empathy Map	Empathy map canvas is prepared to identify the customer or user's Pains & Gains and their feelings and thinking and list of problem statement is prepared.	22 SEPTEMBER 2022
Ideation	In ideation ,Ideas are listed in 3 steps like the first step is problem statement , second step is brainstorming, idea listing and grouping and third step is Idea prioritization based on the feasibility & importance are prepared and submitted for review	1 OCTOBER 2022
Proposed Solution	Proposed solution includes the problem statement, idea, novelty, Customer satisfaction, business model, social impact, scalability of solutions are prepared	5 OCTOBER 2022

Problem Solution Fit	In problem-solution fit document include is customer segment, problems,triggers, Emotions before and after, available solutions, Customer constraint,Behavior, problem root cause, your Solution these things are prepared.	7 OCTOBER 2022
Solution Architecture	Prepare solution architecture document is prepared	13 OCTOBER 2022

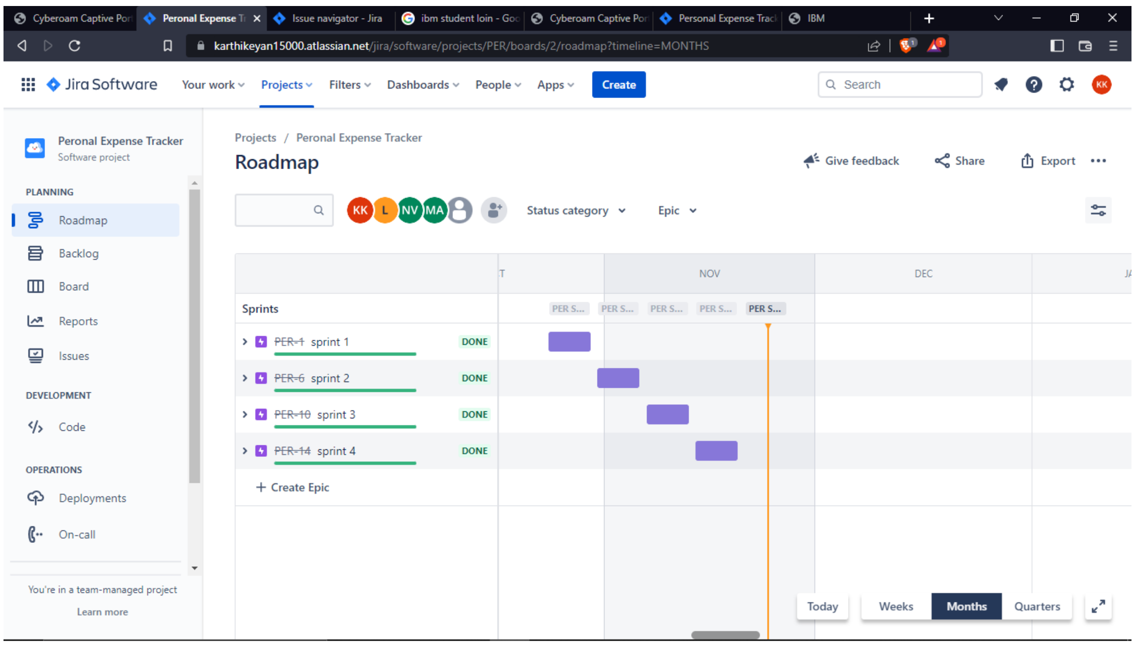
Customer Journey	Customer journey helpful to identify and understand the user interactions, goals and opportunities, positive and negative moments & experiences with the Application (entry to exit).	15 OCTOBER 2022
Functional Requirement	Functional requirement document having the functionalities and non-functionalities of our project	16 OCTOBER 2022
Data Flow Diagrams	Data flow diagrams show the flow of our project and it has been done and submitted for review.	20 OCTOBER 2022
Technology Architecture	Technology Architecture has been done and submitted for review.	27 OCTOBER 2022
Prepare Milestone & Activity List	Prepared the milestones& activity list of the project.	3 NOVEMBER 2022
Project Development - Delivery of Sprint-1, 2, 3& 4	Develop & submit the developed code by testing it.	IN PROGRESS..

6.2 Sprint Delivery Schedule

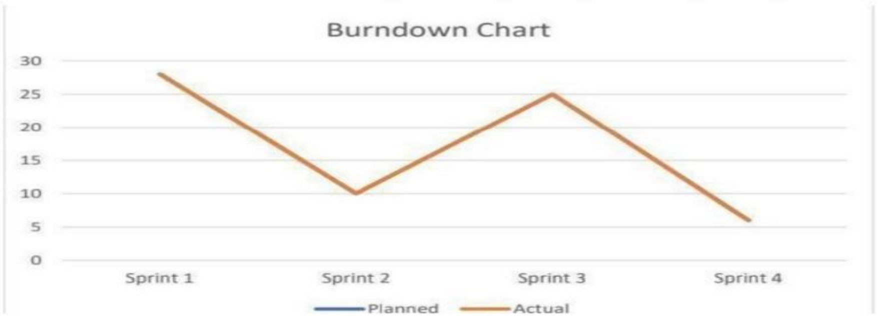
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points
Sprint - 1	Registration	USN -1	As a user, I can register for the application by entering my email, new password and confirming the same password.	2
		USN-2	As a user, I will receive confirmation email onceI have registered for the application.	1
	Login	USN -3	As a user, I can log into the application by entering email and password / Google OAuth.	2
	Dashboard	USN -4	Logging in takes the user to their dashboard.	1
Sprint - 2		USN -5	As a user, I will update my salary at the start of each month	1
		USN -6	As a user, I will set a target/limit to keep track ofmy expenditure.	1
	Workspace	USN -7	Workplace for personal expense tracking	1
	Charts	USN -8	Graphs to show weekly and everyday expenditure	2
		USN -9	As a user, I can export raw data as excel file.	1

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Pri
Sprint - 3	IBM DB2	USN -10	Linking database with dashboard	2	Hig
		USN -11	Making dashboard interactive with JS	2	Hig
	Watson Assistant	USN -12	Embedding Chatbot to clarify user's queries.	1	Lo
	SendGrid	USN -14	Using SendGrid to send mail to the user. (To	1	Me
Sprint - 4	Integration	USN -15	Integrating frontend and backend.	2	Hig
	Docker	USN -16	Creating Docker image of web app.	2	Hig
	Cloud Registry	USN -17	Uploading docker image to IBM cloud registry.	2	Hig
	Kubernetes	USN -18	Creating container using docker and hosting the webapp	2	Hig
	Exposing Deployment	USN -19	Exposing IP/Ports for the site.	1	Me

6.3 Reports from JIRA



Burndown Chart

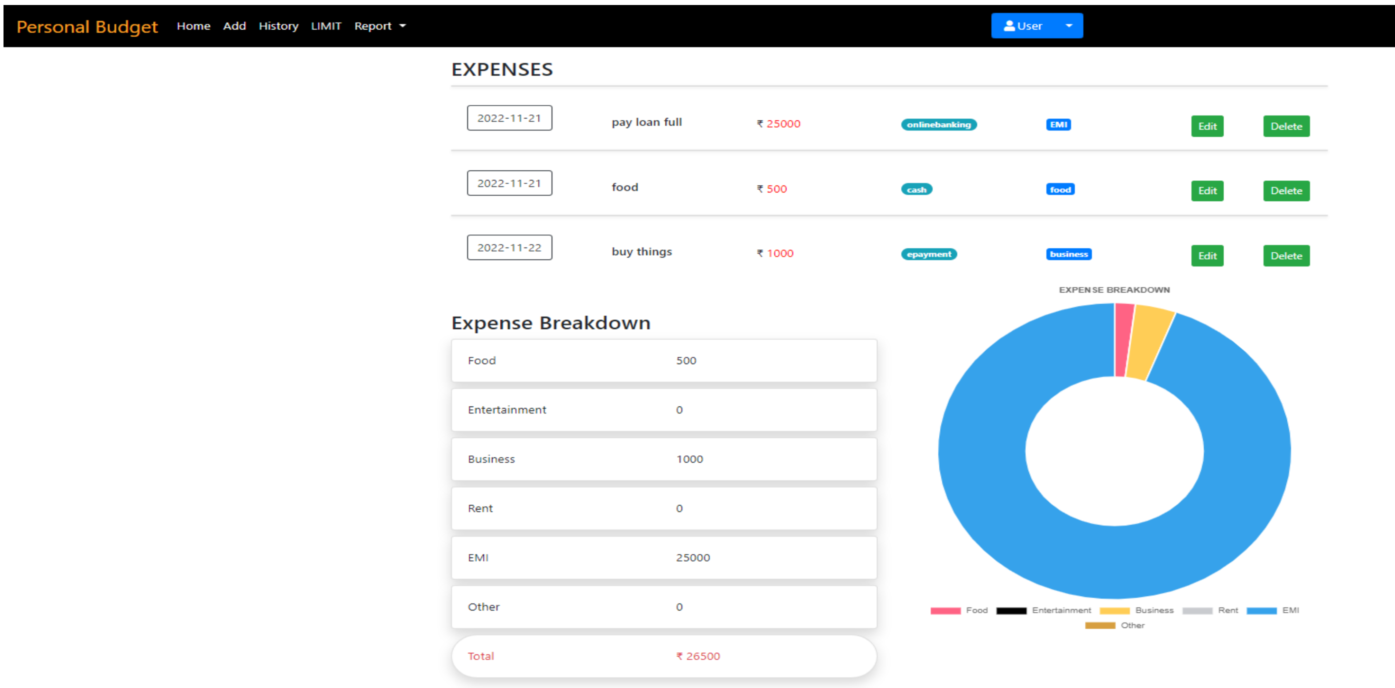


7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1

Display your Expenses According to your Data's Entered

Display Graphs for your Expenses By Analyzing your stored Data's from DataBase



EXPENSES

2022-11-21	pay loan full	₹ 25000	onlinebanking	EMI	Edit	Delete
2022-11-21	food	₹ 500	cash	food	Edit	Delete
2022-11-22	buy things	₹ 1000	epayment	business	Edit	Delete

Expense Breakdown

Food	500
Entertainment	0
Business	1000
Rent	0
EMI	25000
Other	0
Total	₹ 26500

Code

```
@app.route("/display")
def display():
    query = "SELECT * FROM expenses where id = ? ;"
    stmt = ibm_db.prepare(connection, query)
    ibm_db.bind_param(stmt, 1, session['email'])
    ibm_db.execute(stmt)
    dictionary=ibm_db.fetch_assoc(stmt)
    rexpense=[]
    while dictionary != False:
```

```
exp=(dictionary["ID"],dictionary["DATES"],dictionary["EXPENSENAME"],dictionary["A
MOUNT"],dictionary["PAYMODE"],dictionary["CATEGORY"],dictionary["IDX"])
        rexpense.append(exp)
        dictionary = ibm_db.fetch_assoc(stmt)
    que = "SELECT MONTH(dates) as DATES, SUM(amount) as AMOUNT FROM
expenses WHERE id=? AND YEAR(dates)= YEAR(now()) GROUP BY MONTH(dates);"
    stm = ibm_db.prepare(connection, que)
    ibm_db.bind_param(stm, 1,session['email'])
    ibm_db.execute(stm)
    dictionary=ibm_db.fetch_assoc(stm)
    texpense=[]
    while dictionary != False:
        exp=(dictionary["DATES"],dictionary["AMOUNT"])
        texpense.append(exp)
        dictionary = ibm_db.fetch_assoc(stm)
    print(texpense)

    quer = "SELECT * FROM expenses WHERE id = ? AND YEAR(dates)= YEAR(now());"
    st = ibm_db.prepare(connection, quer)
    ibm_db.bind_param(st, 1,session['email']
  ibm_db.execute(st)
    dictionary=ibm_db.fetch_assoc(st)
    expense=[]
    while dictionary != False:

exp=(dictionary["ID"],dictionary["DATES"],dictionary["EXPENSENAME"],dictionary["A
MOUNT"],dictionary["PAYMODE"],dictionary["CATEGORY"],dictionary["IDX"])
        expense.append(exp)
        dictionary = ibm_db.fetch_assoc(st)
    total=0
    t_food=0
```

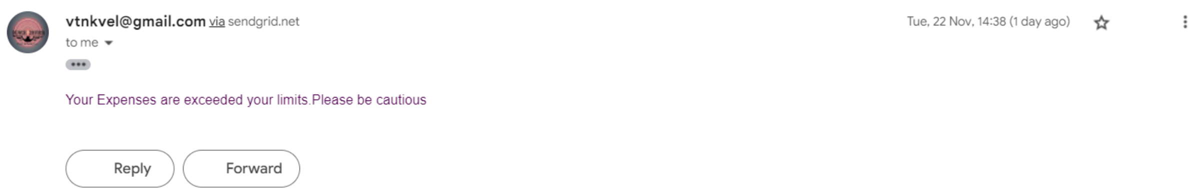
```
t_entertainment=0
t_business=0
t_rent=0
t_EMI=0
t_other
for x in expense:
      total += x[3]
      if x[5] == "food":
          t_food += x[3]

      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)
```

```
for x in lexpense:
      ttotal += x[3]
      if x[5] == "food":
          to_food += x[3]


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


      elif x[5] == "business":
          to_business  += x[3]
      elif x[5] == "rent":
          to_rent  += x[3]


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


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

print(ttotal)



qy = "SELECT max(IDX) as IDX FROM limits where id=?;"
smt = ibm_db.prepare(connection, qy)
ibm_db.bind_param(smt, 1, session['email'])
ibm_db.execute(smt)
dictionary = ibm_db.fetch_assoc(smt)
uexpense=[]
while dictionary != False:
    exp=(dictionary["IDX"])
```

```
        uexpense.append(exp)
        dictionary = ibm_db.fetch_assoc(smt)
    k=uexpense[0]
    qu = "SELECT NUMBER FROM limits where id=? and idx=?"
    sm = ibm_db.prepare(connection, qu)
    ibm_db.bind_param(sm, 1, session['email'])
    ibm_db.bind_param(sm, 2, k)
    ibm_db.execute(sm)
    dictionary = ibm_db.fetch_assoc(sm)
    fexpense=[]
    while dictionary != False:
        exp=(dictionary["NUMBER"])
        fexpense.append(exp)
        dictionary = ibm_db.fetch_assoc(stmt)

    if len(fexpense) <= 0:
        print("Enter the limit First")
    else:
        if ttotal > fexpense[0]:
            m=sendemail.sendgridmail(session["email"])
            print(m)
        else: print("Error")
    return render_template("display.html",rexpense=rexpense, texpense = texpense, expense =
expense,  total = total ,
                           t_food = t_food,t_entertainment =  t_entertainment,
                           t_business = t_business,  t_rent =  t_rent,
                           t_EMI =  t_EMI,  t_other =  t_other )
```

7.2 Features 2

This Feature Enable us to Alert Users By sending emails as Notification when the expense data Exceeds its limits


vtnkvel@gmail.com via sendgrid.net
to me

Tue, 22 Nov, 14:38 (1 day ago)

Your Expenses are exceeded your limits.Please be cautious

Code

```
"""
import requests
def sendgridmail(user):
        url = "https://rapidprod-sendgrid-v1.p.rapidapi.com/mail/send"
        payload = {
                "personalizations": [
                        {
                                "to": [{"email": user}],
                                "subject": "Your Monthly expense is exceeded"
                        }
                ],
                "from": {"email": "vtnkvel@gmail.com"},
                "content": [
                        {
                                "type": "text/plain",
                                "value": "Avoid spending money, your monthly expense is
exceeded..."
                        }
                ]
        }
        headers = {
```

```
                    "content-type": "application/json",

                    "X-RapidAPI-Key":
"8dbcdbb4e0msh2ca0fb8c6cfb3b9p13a154jsn1b29565d9fd5",

                    "X-RapidAPI-Host": "rapidprod-sendgrid-v1.p.rapidapi.com"

          }

          response = requests.request("POST", url, json=payload, headers=headers)

          print(response.text)

          """


from sendgrid import SendGridAPIClient

import os

from sendgrid.helpers.mail import *

def sendgridmail(user):


          sg = SendGridAPIClient('SG.gs4orzGhR82E7l5ICcJQAQ.-
Xb66DzubZ1hBrwJ31a5HSjosFfnYPOyx-eqopz0ccw')

          from_email = Email("vtnkvel@gmail.com")

          to_email = To(user)

          subject = "EXPENSE TRACKER NOTIFICATION"

          content = Content("text/plain", "Your Expenses are exceeded your limits.Please be
cautious")

          mail = Mail(from_email, to_email, subject, content)

          response = sg.client.mail.send.post(request_body=mail.get())

          print(response.status_code)

          print(response.body)

          print(response.headers)
```

7.3 Database Schema

REGISTER

```
id INT NOT NULL GENERATED
ALWAYS AS IDENTITY,username
VARCHAR(255) NOT NULL,
email
```

```
VARCHAR(255
) NOT NULL,
password
VARCHAR(255
) NOT NULL
```

LIMITS

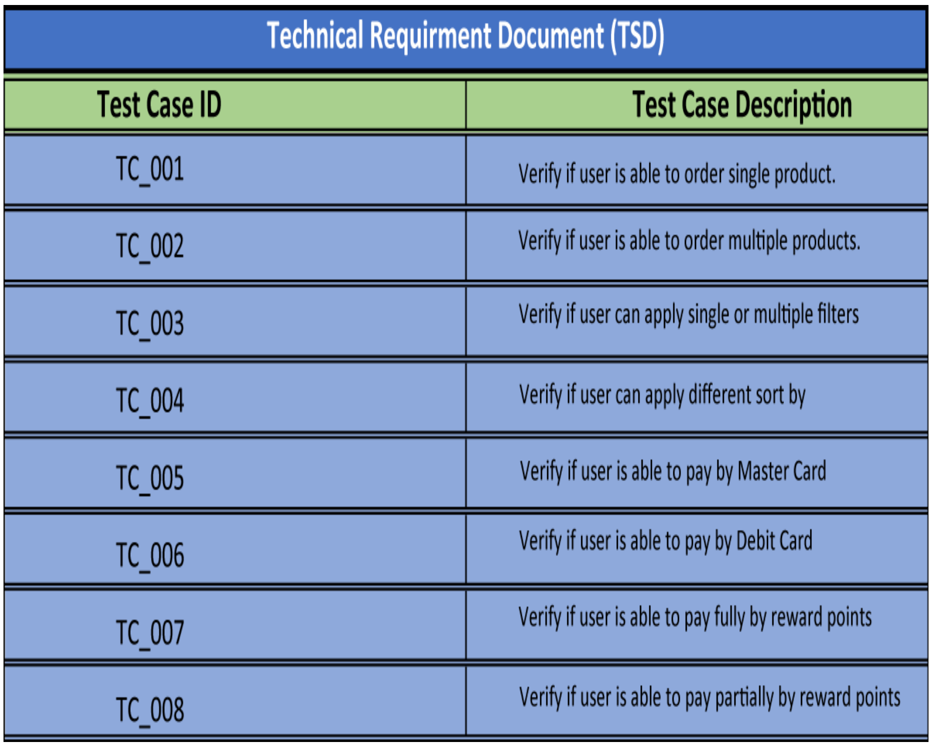
```
id INT NOT NULL GENERATED
ALWAYS AS IDENTITY,userid
VARCHAR(255) NOT NULL,
limitss VARCHAR(255) NOT NULL
```

8.TESTING:

8.1 TEST CASES:

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

8.2 User Acceptance 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

- Tracking income and expenses: Monitoring the income and tracking all expenditures (through bankaccounts, mobile wallets, and credit & debit cards).
- Transaction Receipts: Capture and organize your payment receipts to keep track of your expenditure.
- Organizing Taxes: Import your documents to the expense tracking app, and it will streamline your income and expenses under the appropriate tax categories.
- Payments & Invoices: Accept and pay from credit cards, debit cards, net banking, mobile wallets, and banktransfers, and track the status of your invoices and bills in the mobile app itself. Also, the tracking app sends reminders for payments and automatically matches the payments with invoices.
- Reports: The expense tracking app generates and sends reports to give a detailed insight about profits, losses, budgets, income, balance sheets, etc.,
- E-commerce integration: Integrate your expense tracking app with your eCommerce store and track your sales through payments received via multiple payment methods.
- Vendors and Contractors: Manage and track all the payments to the vendors and contractors added to the mobile app.
- Access control: Increase your team productivity by providing access control to particular users through custom permissions.
- Track Projects: Determine project profitability by tracking labor costs, payroll, expenses, etc., of your ongoing project.
- Inventory tracking: An expense tracking app can do it all. Right from tracking products or the cost of goods, sending alert notifications when the product is running out of stock or the product is not selling, to purchase orders.
- In-depth insights and analytics: Provides in-built tools to generate reports with easy-to- understand visuals and graphics to gain insights about the performance of your business.
- 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 And Disadvantages

Advantages:

1. **Improved visibility:**

   Most expense management software includes a dashboard that compilesemployee expense data and presents it in an easy-to-understand visual format using charts and other graphics.

2. **Security:**

   All the Data's are stored in ibm cloud and db2 so all the data aremaintained safely.

3. **Month wise Comparison:**

   Using the Expense Manager, you can easily make month on month comparisons of earning, expenses and spending in a more organized manner.

4. **Alert Mail:**

   User Receives the alert mail when they exceed the expense limit.

5. **Automation:**

   All the calculations are automated. Graph are generated based on theexpense made.

6. **User Friendly:**

   Expenses can be added easily.

Disadvantage:

1. **Requires Internet Connection:**

   This web application requires an active internet connection to access.

2. **Cost:**

   Using cloud service need some investments. Every time we can't access the cloud freely.

11. Conclusion

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

12.FUTURE SCOPE

1. User can able to upload the receipt of their expenses made.
2. Application will make suggestion to reduce unnecessary expense.
3. User get remainder in email to add their daily expense.
4. User can able to link bank accounts with our application

13.Appendix

Source code

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

from datetime import datetime


import ibm_db

import re

import sendemail


app = Flask(__name__)


hostname = '55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;'

uid = 'xzh21841'

pwd = 'nHocC72lDavJaypx'

driver = "{IBM DB2 ODBC DRIVER}"

db_name = 'Bludb'

port = '31929'
```

```
protocol = 'TCPIP'
cert = "DigiCertGlobalRootCA.crt"
dsn = (
    "DATABASE ={0};"
    "HOSTNAME ={1};"
    "PORT ={2};"
    "UID ={3};"
    "SECURITY=SSL;"
    "PROTOCOL={4};"
    "PWD ={6};"
).format(db_name, hostname, port, uid, protocol, cert, pwd)
connection = ibm_db.connect(dsn, "", "")
app.secret_key = 'a'


#HOME--PAGE
@app.route("/home")
def home():
    return render_template("homepage.html")
@app.route('/register', methods =['GET', 'POST'])
def register():
    global user_email
    msg = ''
    if request.method == 'POST' :
        username = request.form['username']
        email = request.form['email']
        password = request.form['password']
        query = "SELECT * FROM register WHERE email=?;"
        stmt = ibm_db.prepare(connection, query)
        ibm_db.bind_param(stmt, 1, email)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        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:
            query = "INSERT INTO register values(?,?,?);"
            stmt = ibm_db.prepare(connection, query)
            ibm_db.bind_param(stmt, 1, username)
            ibm_db.bind_param(stmt, 2, email)
            ibm_db.bind_param(stmt, 3, password)
            ibm_db.execute(stmt)
            session['loggedin'] = True
            session['id'] = email
            user_email = email
            session['email'] = email
            session['username'] = username

            msg = 'You have successfully registered ! Proceed Login Process'
            return render_template('login.html', msg = msg)
    else:
        msg = 'PLEASE FILL OUT OF THE FORM'
@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)

        return render_template('register.html', msg=msg)
```

Github Source Code link:

https://github.com/IBM-EPBL/IBM-Project-30464-1660147011