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

TEAM ID : PNT2022TMID35361

Personal Expense Tracker Application

A PROJECT REPORT

BALAMURUGAN S (2019105005)
VIGNESH V (2019105065)
PRAVEEN RP (2019105557)
SUGUMAR NR (2019105587)

BACHELOR OF ENGINEERING IN
ELECTRONICS AND COMMUNICATION ENGINEERING

COLLEGE OF ENGINEERING GUINDY
ANNA UNIVERSITY
CHENNAI – 600054

INDEX

7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
    1. Feature 1
    2. Feature 2
    3. Database Schema (if Applicable)
8. **TESTING**
    1. Test Cases
    2. User Acceptance Testing
9. **RESULTS**
    1. Performance Metrics
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**

        Source Code GitHub Link & Project Demo Link

1. INTRODUCTION

1.1 Project Overview

Category: Cloud AppDevelopment
Team ID: PNT2022TMID35361

Skills Required:

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

Project Description:

Income and Expense Tracker will maintain data of daily, weekly, monthly, yearly expenses, Manages your expenses and earnings in a simple and intuitive way. The web application “Personal Expense Tracker” is developed to manage the daily expenses in a more efficient and manageable way. By using this application we can reduce the manual calculations of the daily expenses and keep track of the expenditure
This application will ask users to add their expenses and based on that 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.

1.2 Purpose

In this project we propose a web application known as “Personal Expense Tracker” which is helpful to manage out income and expense as a daily or periodically or else whenever we want to remind. It also acts as an indicator or reminder example in the fastest world which we can’t able to remember what are the things we have to do for the end of month and what are the payments we have to pay for the particular month. Due to some conflict or some other stress we forget some times that what are the income or where the money has to come from or what the payments we have to pay. This application will help you to make a note for what or the things we have to do for the end of month. For example, like how much it expenses for monthly and what are the expenses for a month. Some of the expense features like food expenses billing expenses like phone, electricity, taxation and some other personal expense. In this fast-moving world this web application will be very useful for a people who was a family and especially for a business people. Budgeting is an integral part of the society. Budget Tracking involvesrecording and analyzing the incomes and expenses of a person or an

organization over a particular period of time. Today, since we are living in a hurry up and get it done society, many people are looking forward to efficient ways to budget their time and money. During the recent years, some research has been carried out on household budget. It has been noted that in most cases, budget management is being done mentally and never being put on paper which makes Budget Tracking very difficult.

2. LITERATURE SURVEY

2.1 Existing problem

In a study conducted by Forrester in 2016 surveying small and medium businesses (SMBs) across the world, 56% companies reported expense management as being the biggest challenge for their finance departments. In another survey conducted by Level Research in 2018 in North America, respondents reported the following pain points in expense management before adopting automation:

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

2.2 References

S.NO	TITLE	AUTHOR	YEAR	ABOUT
1	D2D Expense tracker application	Anjali Kumara, Utkarsh Raj	2021	D2D App aims to assist everybody WHO square measure progressing to understand their expenses and save from it. Here user will outline their own classes for expense sort like food, clothing, rent and bills wherever they need to enter the cash that has been spent and can also add some info in further info to specify the expense.
2	Expense tracker	Atiya Kazi, Praphulla.S	2020	This project is an android app which is used to track the daily expenses of the user. It is like digital record keeping which keeps the records of expenses done by a user. The application keeps the track of the Income and Expenses both of user on a day-to-day basis.

3	Expenditure management system	Dr.V.Geetha, G. Nikhitha	2022	In this project a weekly, monthly, and yearly basis, details of expenses will be displayed in the form of a pie chart. It aids us in remembering and adding information about what money we receive from others and what costs or payments we must make on a given date or month.
4	Expense Tracker	Miriam Thomas, Lekshmi.P,Dr.T.Mahalekshmi	2020	By using this application we can reduce the manual calculations of the daily expenses and keep track of the expenditure. In this application, user can provide his income to calculate his total expenses per day and these results will be stored for each user.
5	Tracking Expenses by Commodity at Widget Farmers Cooperative	Ramasamy	2004	Widget Farmers Coop (WFC) is a large retail agricultural supply cooperative with 12 locations in two states. It has over 40 million dollars in annual sales each year since its creation in 2004. WFC management would like to track expenses and identify areas of the business that are profitable and capitalize on them, as well as identify areas that are not profitable and realign or eliminate them.

2.3 Problem Statement Definition

PROBLEM STATEMENT:

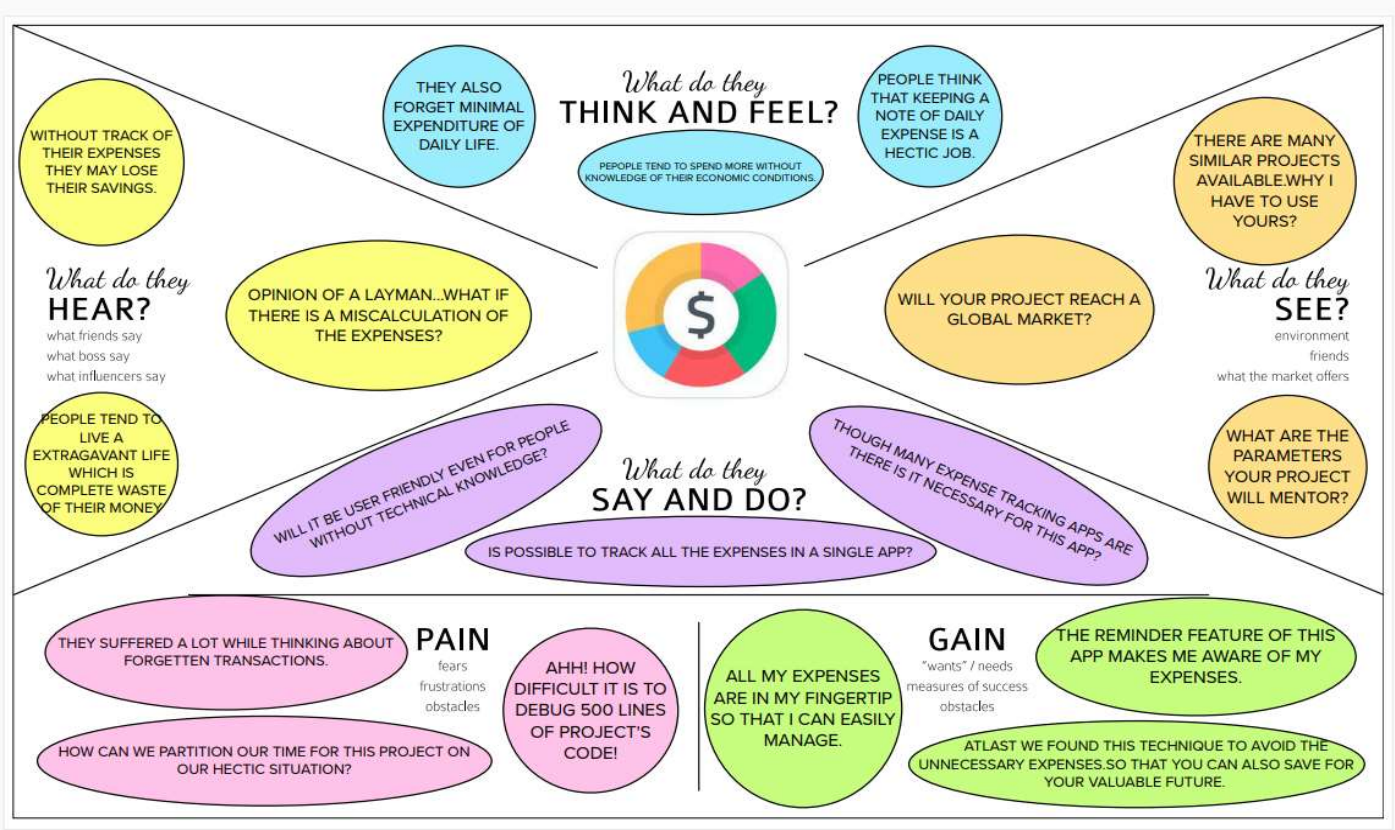
Many organizations have their own system to record their income and expenses, which they feel is the main key point of their business progress. It is good habit for a person to record daily expenses and earning but due to unawareness and lack of proper applications to suit their privacy, lacking decision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

Who does the problem affect?	**People getting regular wages.**
What is the issue?	The paper based expense tracker system does not provide the user portability , existing system only used on paper based records so unable to update anywhere expenses done and unable to update the location of the expense details disruptive that the proposed system.
When does the issue occurs?	When the digits could not be recognized correctly. When the transactions are not successful. When the elder people unable to understand the smaller handwritten digits. When the paper based expense tracker records are subjected to fire accident, flood, etc.
Where is the issue occurring?	The issue occurs when the person is unable to track his income and expenditure.

Why is it important that we fix the problem ?	By solving this issue those people getting regular wages can track their expenses and avoid unwanted expenses.

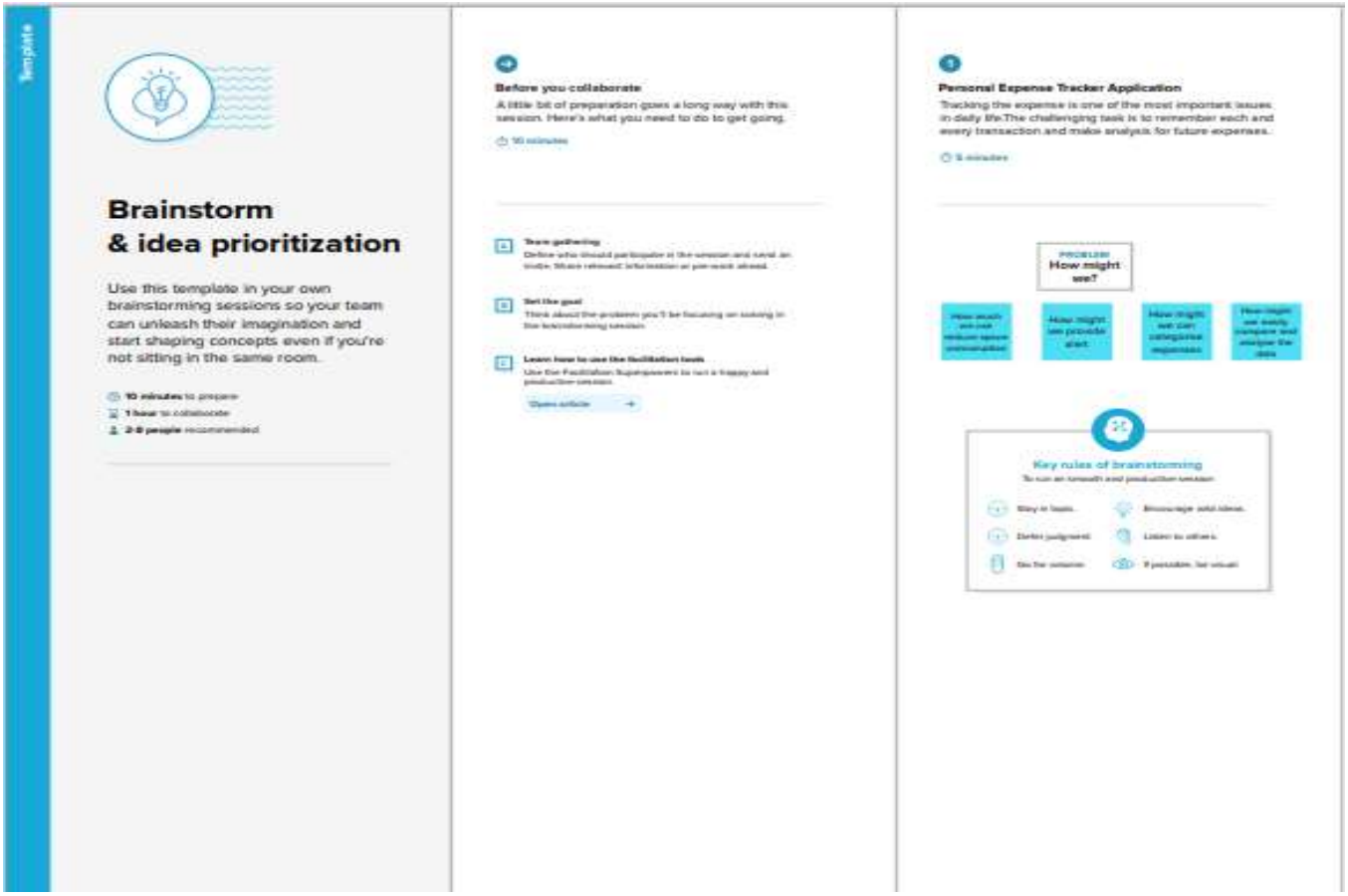
3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map

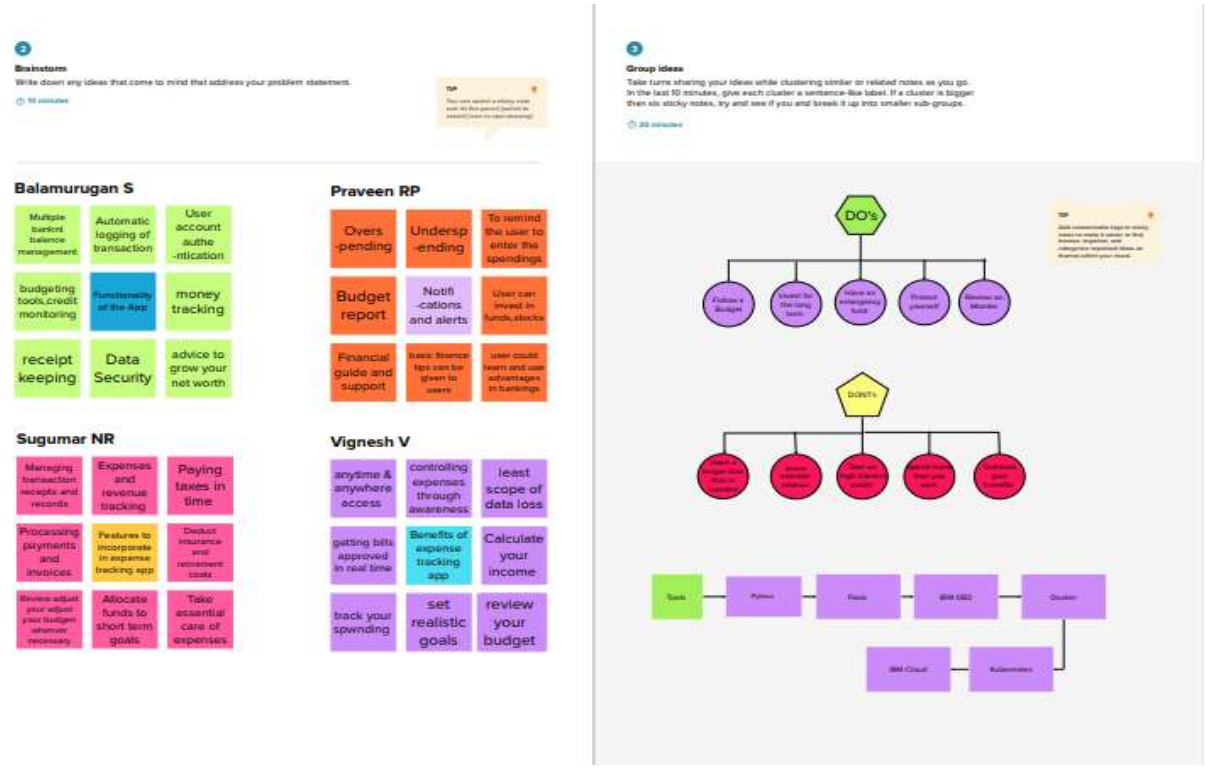


3.2 Idea on & Brainstorming

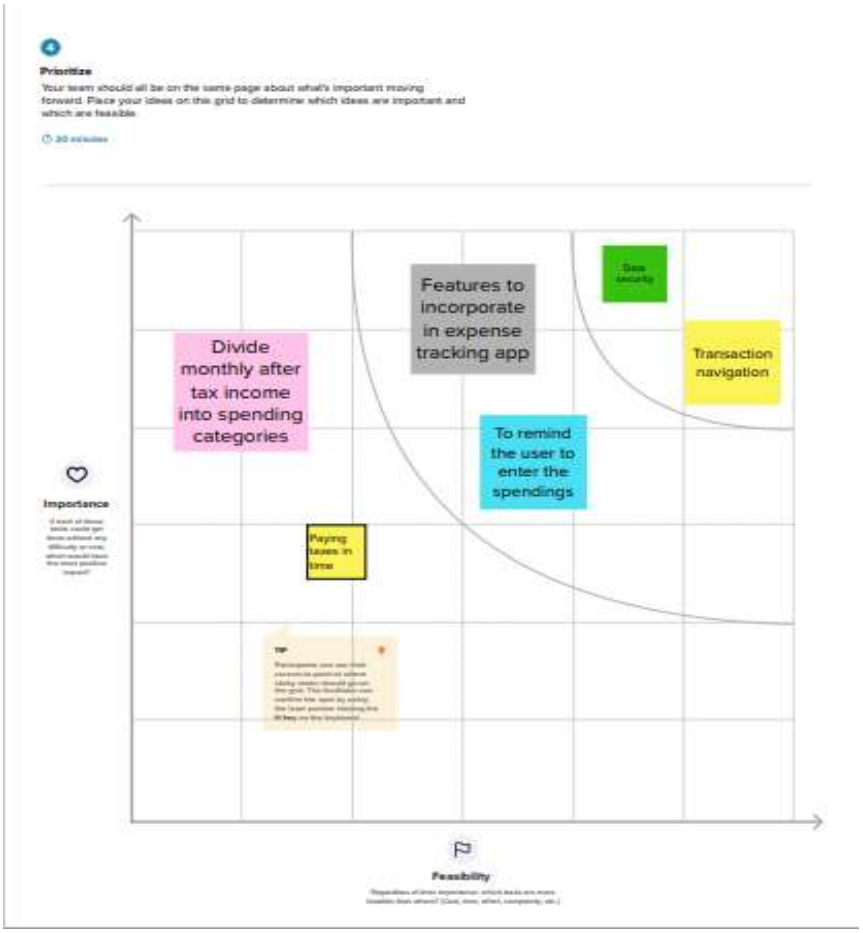
Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea listening and Grouping



Step-3: Idea Prioritization

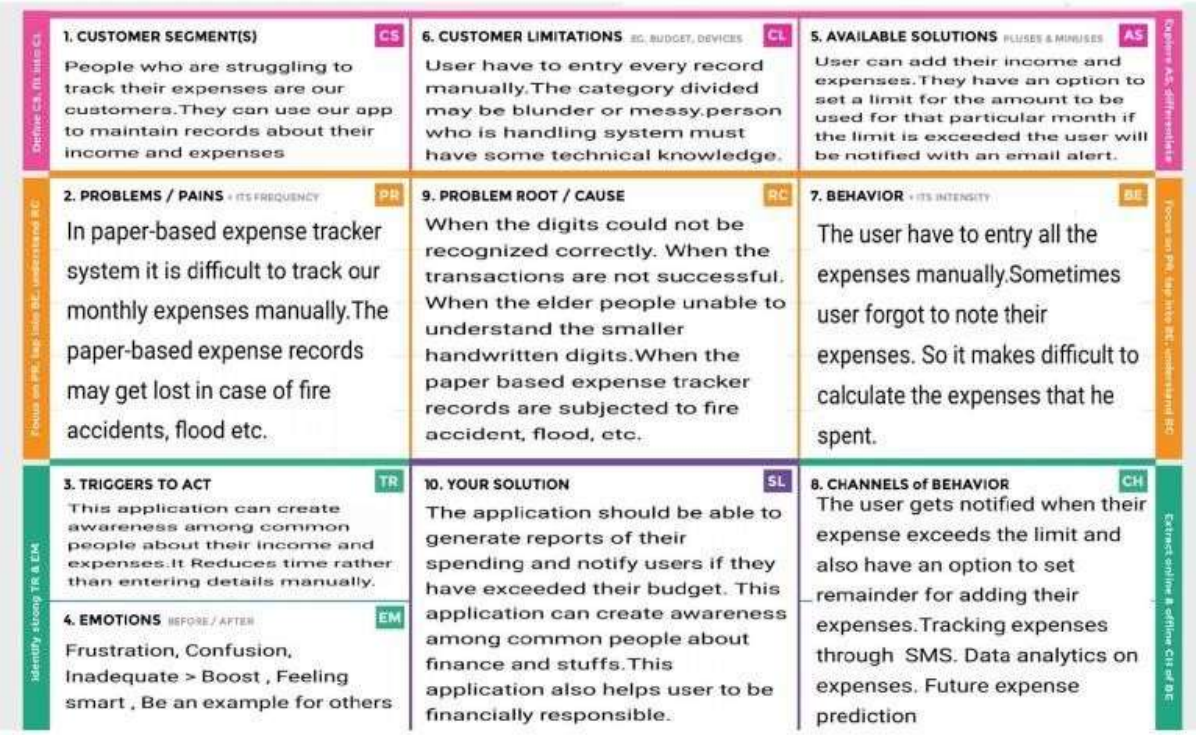


3. Proposed Solution

S.NO.	Parameter	Description
1.	Problem Statement	In paper-based expense tracker system it is difficult to track our monthly expenses manually. In paper-based expense tracker system it is difficult to track our monthly expenses manually. The paper-based expense records may get lost in case of fire accidents, flood etc.
2.	Scalability of the Solution	This application can handle large number of users and data with high performance and security. This application can adapt for both large-scale and small- scale purposes. Easily available in all kinds of devices.
3.	Idea / Solution description	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. 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. 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.
4.	Novelty / Uniqueness	The user gets notified when their expense exceeds the limit and also it reminds the user when they forgot to make entry. Tracking expenses through SMS. Data analytics on expenses. Future expense prediction

5.	Social Impact / Customer Satisfaction	The application should be able to generate reports of their spending and notify users if they have exceeded their budget. It is designed to be dynamic to produce the prediction. It also provides users' personal information, their income as well as their expenses. This application can create awareness among common people about finance and stuffs. This application also helps user to be financially responsible. It Reduces time rather than entering details manually.
6.	BusinessModel (Revenue Model)	This Application is provided for free of cost. But It will have some advertisement. In premium version there is no advertisement and contains some additional features.

4. Problem Solution fit



4. REQUIREMENT ANALYSIS

4.1 Functional requirement

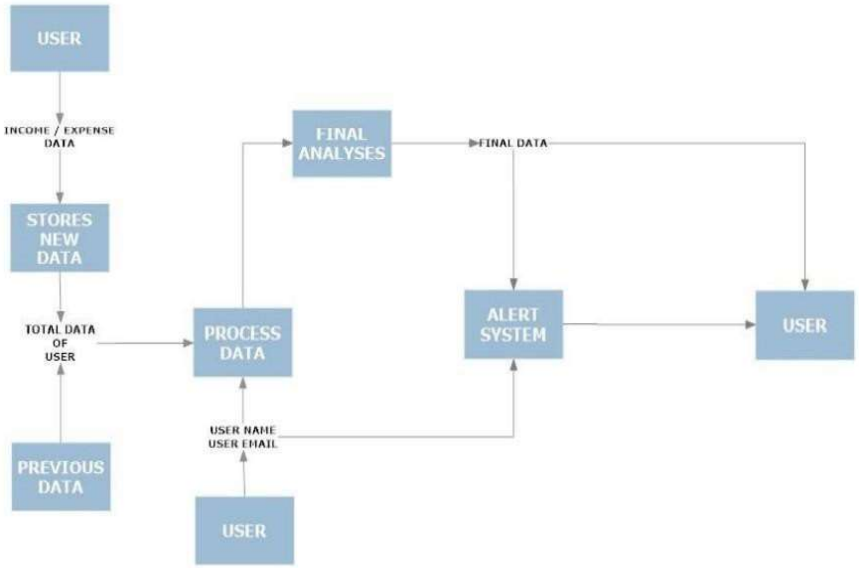
FR No.	Functional Requirement (Epic)	Sub Requirement(Story / Sub- Task)
FR- 1	User Registration	Form for collecting details
FR- 2	Login	Enter username and password
FR- 3	Calendar	Personal expense tracker application must allo w user to add the data to their expenses.
FR- 4	Expense Tracker	This application should graphically represent the expense in the form of report.
FR- 5	Report generation	Graphical representation of report must be generated.
FR- 6	Category	This application shall allow users to add ca tegories of their expenses.

4.2 Non-Functional requirements

FR No.	NonFunctional Require ment	Description
NFR- 1	Usability	Helps to keep an accurate record of your income and expenses.
NFR- 2	Security	Budget tracking apps are considered very sa fe from those who commit cyber crimes.
NFR- 3	Reliability	Each data record is stored on a well built efficient database schema. There is no risk of data loss.
NFR- 4	Performance	The types of expense are categories along with an option. Throughput of the system is increase d due to light weight database support.
NFR- 5	Availability	The application must have a 100% up-time.
NFR- 6	Scalability	The ability to appropriately handle increasin g demands.

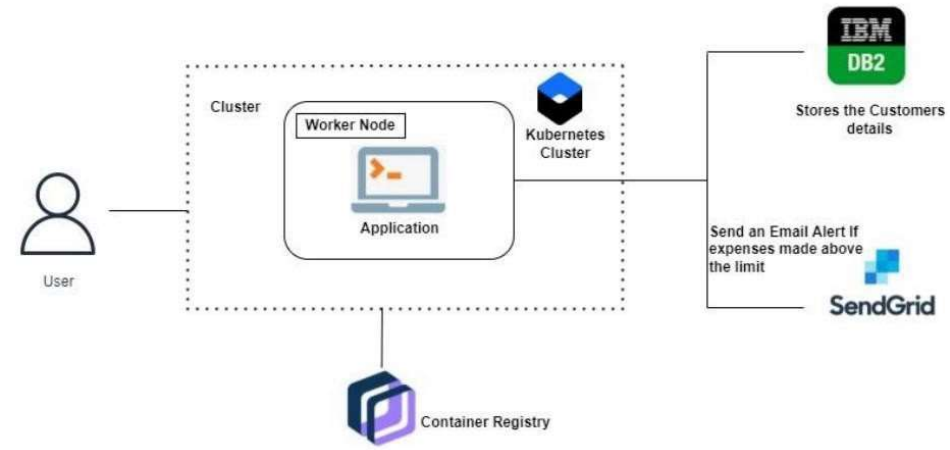
5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture

Technical Architecture:



Solution Architecture

5.3 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 confirming my password.	I can access my account / dashboard	High
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High
		USN- 3	As a user, I can register for the application through Facebook	I can register & access the Dashboard with Facebook Login	Low
	Login	USN - 4	As a user, I can log into the application by entering email & password	I can access the application	High
	Dashboard	USN - 5	As a user I can enter my income and expenditure details.	I can view my daily expenses	High
Customer Care Executive		USN – 6	As a customer care executive I can solve the log in issues and other issues of the application.	I can provide support or solution at any time 24*7	Medium
Administrator	Application	USN - 7	As a administrator I can upgrade or update the application.	I can fix the bug which arises for the customers and users of the application	Medium

6.PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by ente ring my email, password, and confirming my password.	8	High	VIGNESH , PRAVEEN
Sprint-1	Login	USN-2	As a user, I can log into the application by entering email & password	8	High	BALAMURUGAN, SUGUMAR
Sprint-2	Add Expense	USN-3	As a user, I can add the day-to- day expense to the application	5	Medium	VIGNESH , PRAVEEN
Sprint-2	Edit and Delete Expense	USN-4	As a user, I can edit and delete the previously created expense	5	Medium	BALAMURUGAN, Panner
Sprint-3	Creating time based filters in history.	USN-5	As a user, I can see the time based history of expenses.	8	High	VIGNESH , PRAVEEN
Sprint-3	Integrating with pie-charts for analysis	USN-6	As a user, I can view diagrammatic representation of expenses	5	Medium	BALAMURUGAN, SUGUMAR
Sprint-4	Enabling limit feature	USN-7	As a user, I can set monthly limit to expenses	5	Medium	VIGNESH , PRAVEEN
Sprint-4	Sending Email Alerts	USN-8	As a user, I will receive a mail if I cross a limit	8	High	BALAMURUGAN, SUGUMAR

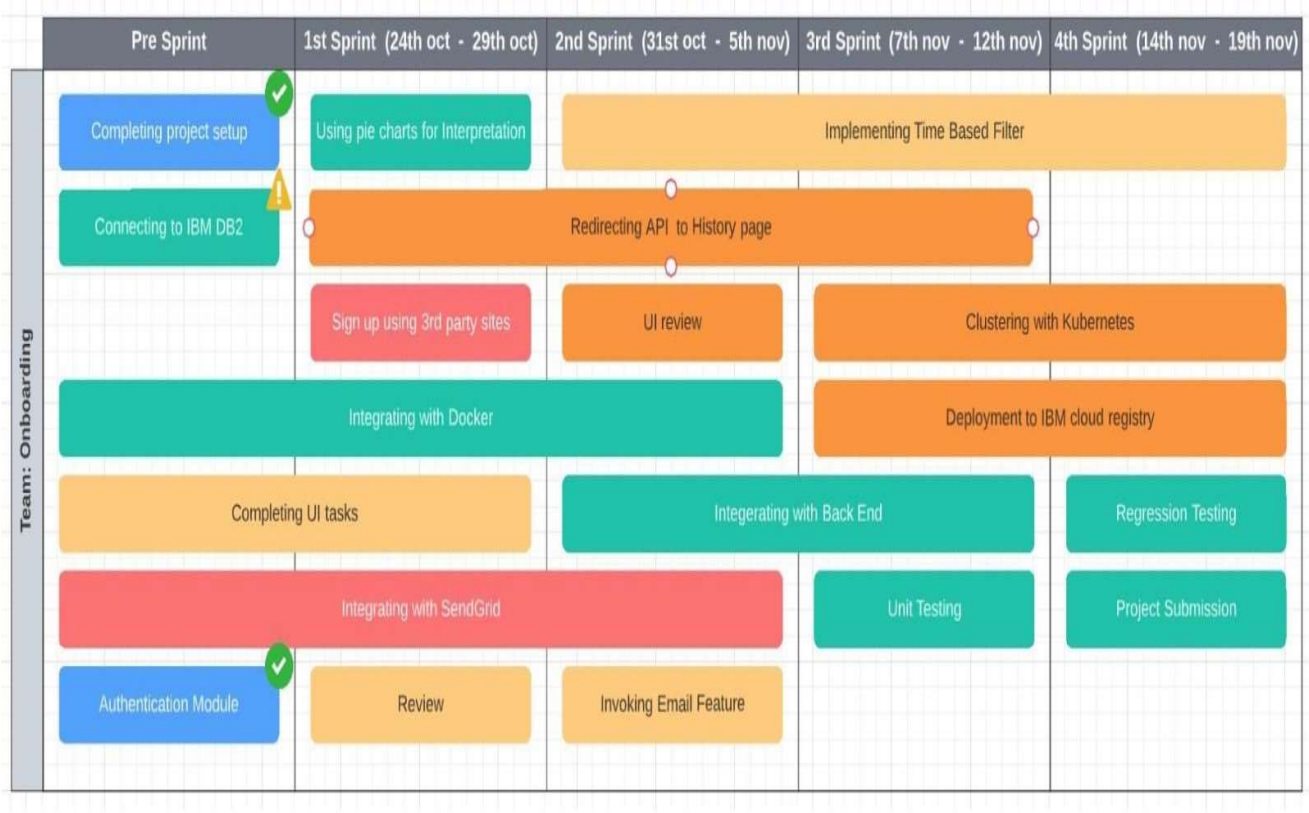
6.2 Sprint Delivery Schedule

Sprint	Total Story P oints	Dur at ion	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as o n Planned End Date)	Sprint Release D ate (Actual)
Sprint-1	20	6 Days	23 Oct 2022	28 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	30 Oct 2022	04 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	06 Nov 2022	11 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	13 Nov 2022	18 Nov 2022	20	19 Nov 2022

Velocity

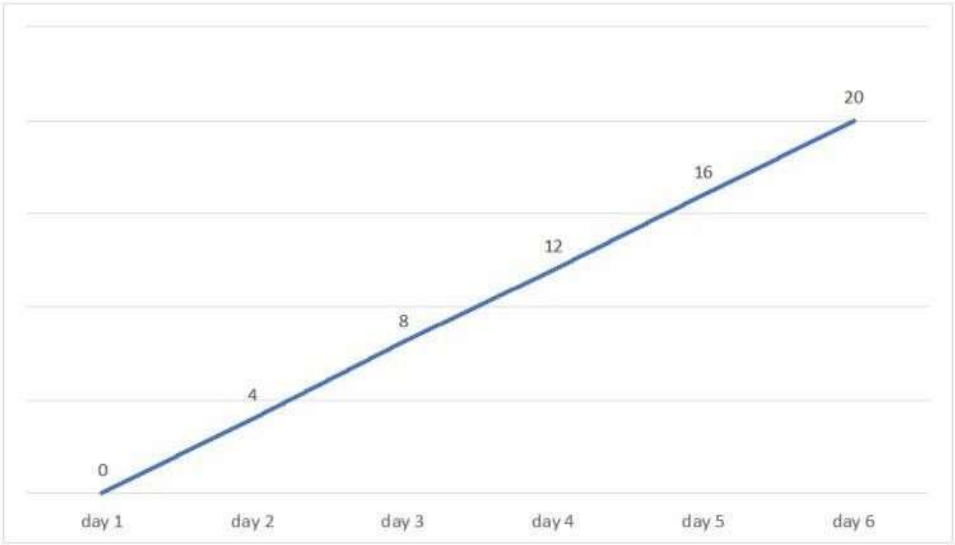
We have a 6-day sprint duration, and the velocity of the team is 20 (points per sprint). Calculating the team's average velocity (AV) per iterationunit (story points per day)

$$AV = \textbf{sprint duration / velocity} = 20/6 = 3.33$$



Burn down Chart:

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



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

7.1 FEATURE 1

We have added the data visualization methods for expenditure. The pie chart have been used to represent the monthly expenses. The pie chart is a pictorial representation of data that makes it possible to visualize the relationships between the parts and the whole of a variable. For example, it is possible to understand the industry count or percentage of a variable level from the division by areas or sectors. The recommended use for pie charts is two-dimensional, as three-dimensional use can be confusing.

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

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

CODE:

Signup.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/dAiS6
JXm" 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</h
                     1>

 <!--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 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="trems">
<input class="check" type="checkbox" required/>
 <p class="conditions">I read and agree to <a href="#">Trems &amp;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


from flask_db2 import DB2
import ibm_db
import ibm_db_dbi
from sendemail import sendgridmail,sendmail
from flask_mail import Mail, Message

import os


app.config['MAIL_SERVER']='smtp.gmail.com'
app.config['MAIL_PORT'] = 465
app.config['MAIL_USERNAME'] = 'balamurugan15042001@gmail.com'
app.config['MAIL_PASSWORD'] = '*********'
app.config['MAIL_USE_TLS'] = False
app.config['MAIL_USE_SSL'] = True
mail = Mail(app)

app.secret_key = 'a'

   mysql = DB2(app)

   conn_str='database=bludb;hostname=2d46b6b4-cbf6-40eb-bbce-
6251e6ba0300.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;port=32328;protocol=tcpip;\
        uid=lsc91268;pwd=dlWyz6qJK3v27xP6;security=SSL'
   ibm_db_conn = ibm_db.connect(conn_str,"")
```

```
    print("Database connected without any error !!")
except:

    print("IBM DB Connection error   :     " + DB2.conn_errormsg())


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

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

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



@app.route('/register', methods =['GET', 'POST'])
def register():
    msg = ''
    print("Break point1")
    if request.method == 'POST' :
        name = request.form['username']
        email = request.form['email']
        password = request.form['password']

        print("Break point2" + "name: " + name + "------" + email + "------" + password)

        try:
            print("Break point3")
            connectionID = ibm_db_dbi.connect(conn_str, '', '')
            cursor = connectionID.cursor()
            print("Break point4")
        except:
            print("No connection Established")


        print("Break point5")
        sql = "SELECT * FROM account WHERE name = ?"
        stmt = ibm_db.prepare(ibm_db_conn, sql)
        ibm_db.bind_param(stmt, 1, name)
        ibm_db.execute(stmt)
        result = ibm_db.execute(stmt)
        print(result)
        account = ibm_db.fetch_row(stmt)
        print(account)

        param = "SELECT * FROM account WHERE name = '" + "\'" + name + "\'"
        res = ibm_db.exec_immediate(ibm_db_conn, param)
        print("---- ")
        dictionary = ibm_db.fetch_assoc(res)
        while dictionary != False:
            print("The ID is : ", dictionary["USERNAME"])
```

```
        dictionary = ibm_db.fetch_assoc(res)

    print("break point 6")
    if account:
        msg = 'Username already exists !'
    elif not re.match(r'[^@]+@[^@]+\.[^@]+', email):
        msg = 'Invalid email address !'
    elif not re.match(r'[A-Za-z0-9]+', name):
        msg = 'name must contain only characters and numbers !'
    else:
        sql2 = "INSERT INTO account (name, email,password) VALUES (?, ?, ?)"
        stmt2 = ibm_db.prepare(ibm_db_conn, sql2)
        ibm_db.bind_param(stmt2, 1, name)
        ibm_db.bind_param(stmt2, 2, email)
        ibm_db.bind_param(stmt2, 3, password)
        ibm_db.execute(stmt2)
        msg = 'You have successfully registered !'
    return render_template('signup.html', msg = msg)



@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' :
        name = request.form['username']
        password = request.form['password']

        sql = "SELECT * FROM account WHERE name = ? and password = ?"
        stmt = ibm_db.prepare(ibm_db_conn, sql)
        ibm_db.bind_param(stmt, 1, name)
        ibm_db.bind_param(stmt, 2, password)
        result = ibm_db.execute(stmt)
        print(result)
        account = ibm_db.fetch_row(stmt)
        print(account)

        param = "SELECT * FROM account WHERE name = '" + "\'" + name + "\'" + " and password = " + "\'" + password + "\'"
        res = ibm_db.exec_immediate(ibm_db_conn, param)
        dictionary = ibm_db.fetch_assoc(res)


        if account:

            id = dictionary.get("ID")
            password_db = dictionary.get("PASSWORD")
            if password == password_db:
                check = 1
                print("Success")
            return redirect('/home')
```

```
        else:
            msg = 'Incorrect username / password !'

    return render_template('login.html', msg = msg)

@app.route('/logout')

def logout():
   session.pop('loggedin', None)
   session.pop('id', None)
   session.pop('username', None)
   session.pop('email', None)
   return render_template('home.html')


port = os.getenv('VCAP_APP_PORT', '8080')
if __name__ == "__main__":
    app.secret_key = os.urandom(12)
    app.run(debug=True, host='0.0.0.0', port=port)
```

7.2 FEATURE 2

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

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

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

Sendemail.py

```
import smtplib
import sendgrid as sg
import os
SUBJECT = "Expense Tracker"
s = smtplib.SMTP('smtp.gmail.com', 587)

def sendmail(TEXT,email):
   s = smtplib.SMTP('smtp.gmail.com', 587)
   s.starttls()
   s.login("balamurugan15122001@gmail.com", "******")
   message  = 'Subject: {}\n\n{}'.format(SUBJECT, TEXT)
   s.sendmail("balamurugan15122001@gmail.com", email, message)
   s.quit()
```

7.3 Database

Schema Tables

REGISTER

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

EXPENSES

id INT NOT NULL GENERATED ALWAYS AS IDENTITY,
userid INT NOT NULL,date TIMESTAMP NOT NULL,

expensename VARCHAR(255) NOT
NULL, amount INT NOT NULL,
paymode VARCHAR(255) NOT NULL,
category 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

1. TEST CASES

TEST CASE ID	PURPOSE	**TEST CASES**	**RESULT**
TC1	Authentication	Password with length less than 4 characters	Password cannot be less than 4 characters
TC2	Authentication	User name with length less than 2 characters	User name cannot be less than 2 characters
TC3	Authentication	Valid user name with minimum2 characters	User name accepted
TC4	Authentication	User name left blank	User name cannot be less than 2 characters
TC5	Authentication	Password field left blank	Password cannot be empty
TC6	Authentication	Minimum 4 characters valid password	Password accepted
TC7	Authentication	Password and Confirm Password did not match	Pleaseenter same password
TC8	Authentication	Confirm Password field left blank	Pleaseenter same password

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

1. Performance Metrics

- Tracking income and expenses: Monitoring the income and tracking all expenditures (through bank accounts, 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 bank transfers, and track the status of your invoices and bills in the mobile app itself. Also, the tracking app sends reminders for payments and automatically matches the payments with invoices.
- 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 compiles employee 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 are maintained 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 the expense 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

Github Source code link:

https://github.com/IBM-EPBL/IBM-Project-5897-1658819614

Project Demo Link:

https://drive.google.com/file/d/1SY_9RI4d5tW8FbrDSBTTK60lJHJd4u_2/view?usp=sharing