PERSONAL EXPENSE TRACKER APPLICATION

1. INTRODUCTION

a. Project Overview

Personal expense tracker entails all financial decisions and activities. It makes our life easier by helping us to manage our expenses efficiently.Using this application, We can know where our money goes and can ensure that our money is used wisely. It helps you track all transactions like bills, refunds, payrolls, receipts, taxes, etc., on a daily, weekly, and monthly basis.

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.

b. Purpose

This app helps to keep an accurate record of your money inflow and outflow. It not only helps with budgeting and accounting but also give helpful insights about money management.

- ➤ It helps achieve business goals
- ➤ Has an easy user interface for interaction
- ➤ Generated detailed reports to give insights about profits, loss, budgets, income, balance sheets, etc.
- ➤ Helps having a smart budget plan
- ➤ Has enhanced AI based bot for answering all queries and addressing user needs
- ➤ Avoiding data loss

2. LITERATURE SURVEY

a. Existing problem

- Lack of time for proper noting down of expenses
- Data entry errors
- Improper management of money
- Lack of visibility of expenses
- Lost receipts
- Internet issues(In case of online webapps)
- Security
- Unavailable balance due to less saving of money

b. References

https://www.researchgate.net/publication/347972162_Expense_Manager_Application

A mobile application that Keeps track of all of your daily transactions, keeps track of your money lent or borrowed ,suggests you with the most effective investment options, offers your discounts in popular categories , view exchange and to read latest authenticated financial news.

https://ijirt.org/master/publishedpaper/IJIRT150860_PAPER.pdf

A working prototype of an intelligent online budget tracker where security issues like web security or network security have also been treated in the design and development of the system, thus increasing the reliability of the system.

https://ijarsct.co.in/Paper391.pdf

“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.

https://github.com/simplyvinay/vue-expenses

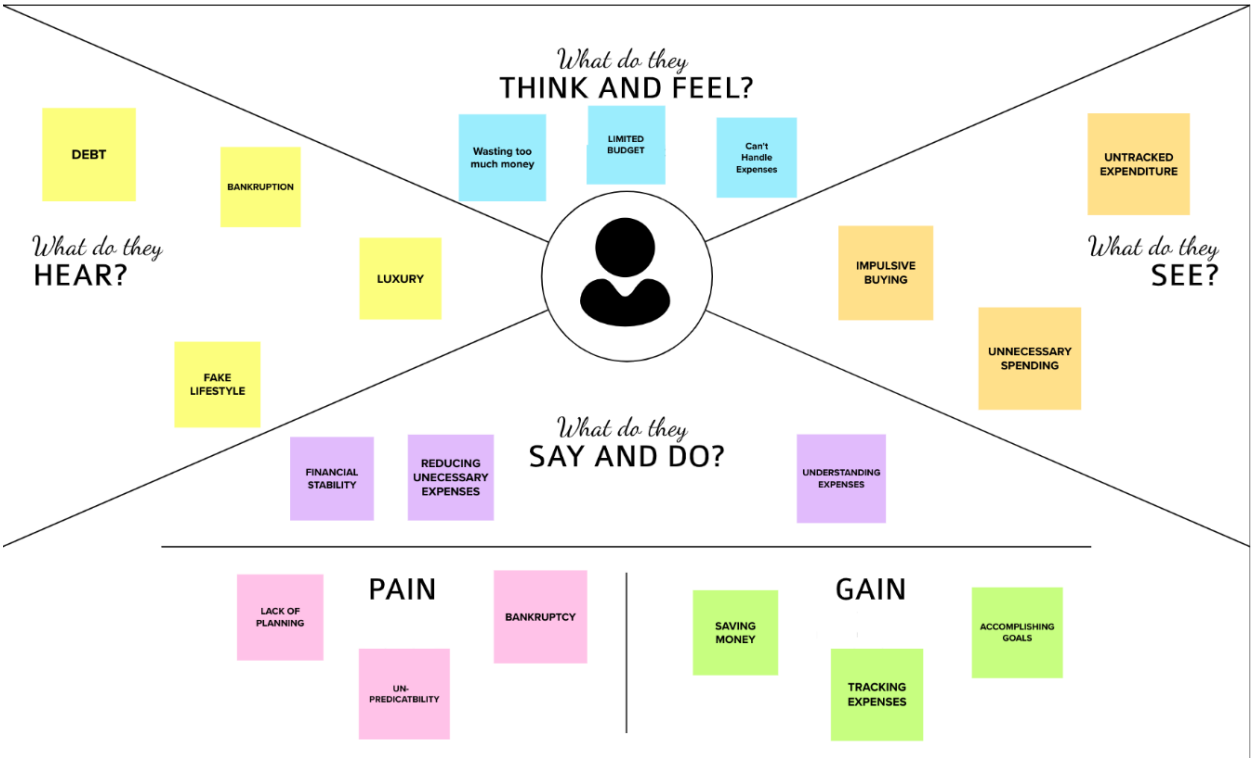
A simple expenses tracking application built with VueJs and .NET

c. Problem Statement Definition

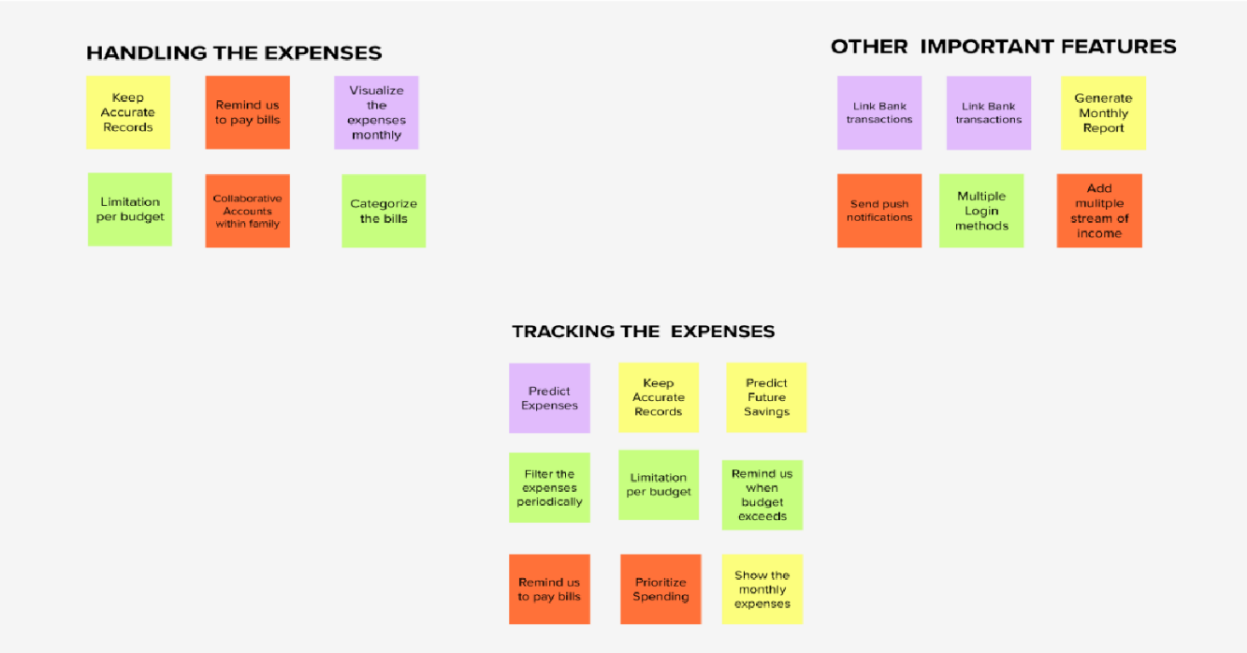
I) Expense Tracker is a day-to-day expense management system designed to easily and efficiently track the daily expenses of an individual. In simple words, personal finance entails all the financial decisions and activities that a Finance app makes your life easier by helping manage your finances efficiently.
ii)A personal finance app will not only help you with budgeting and accounting but also give you helpful insights about money management.
iii)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.
iv)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.

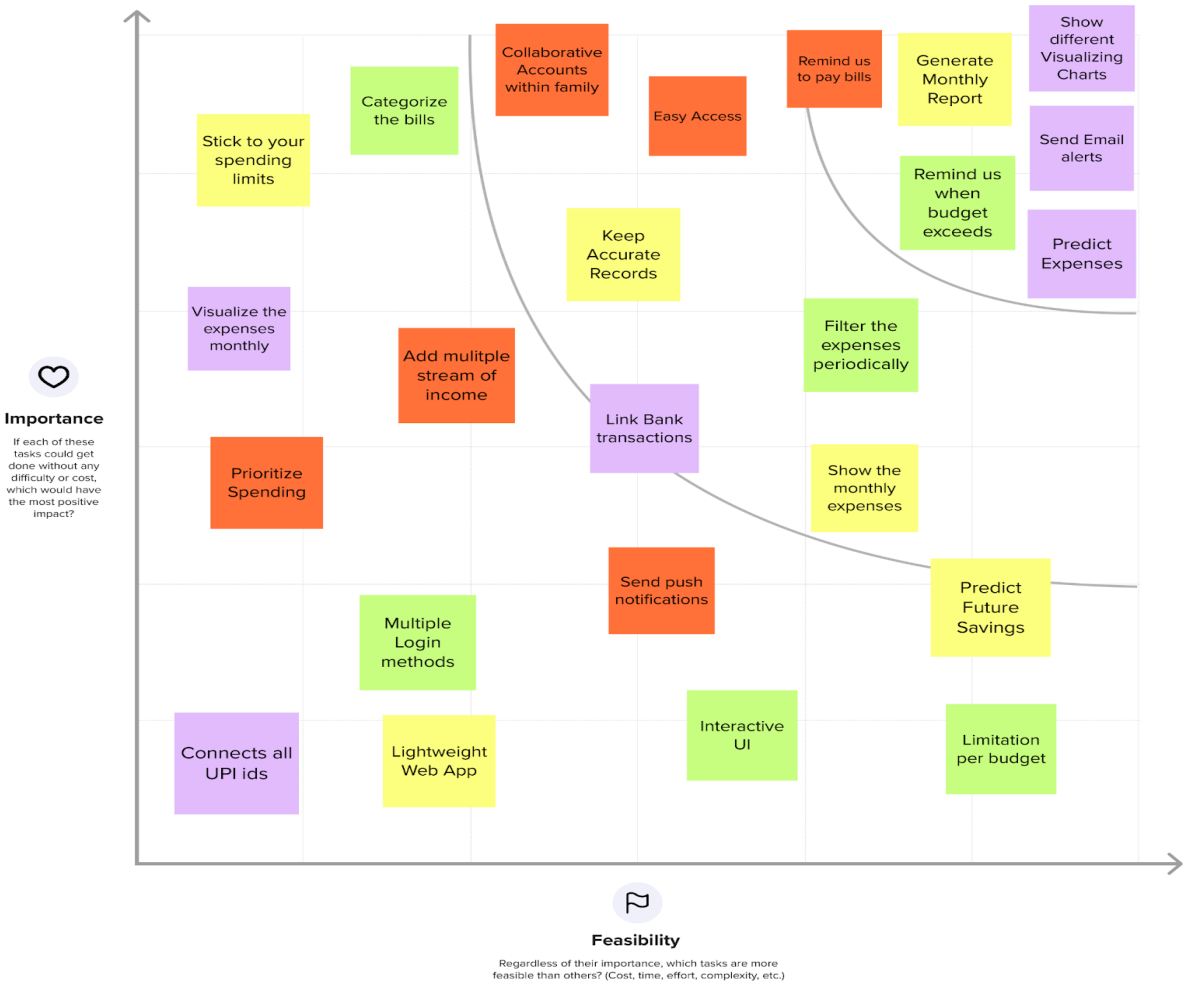
3. IDEATION & PROPOSED SOLUTION

a. Empathy Map Canvas



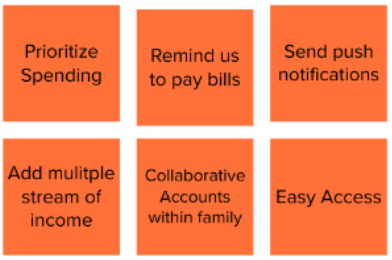
b. Ideation & Brainstorming




Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?
Collaborative Accounts within family
Remind us to pay bills
Generate Monthly Report
Show different Visualizing Charts
Categorize the bills
Easy Access
Send Email alerts
Stick to your spending limits
Remind us when budget exceeds
Keep Accurate Records
Predict Expenses
Visualize the expenses monthly
Filter the expenses periodically
Add mulitple stream of income
Link Bank transactions
Prioritize Spending
Show the monthly expenses
Send push notifications
Predict Future Savings
Multiple Login methods
Interactive UI
Connects all UPI ids
Lightweight Web App
Limitation per budget
Feasibility
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

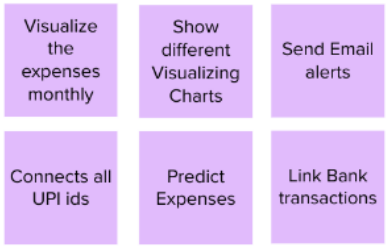
Akshay Ajit

Santhosh Kumar S



Sivakumar TV

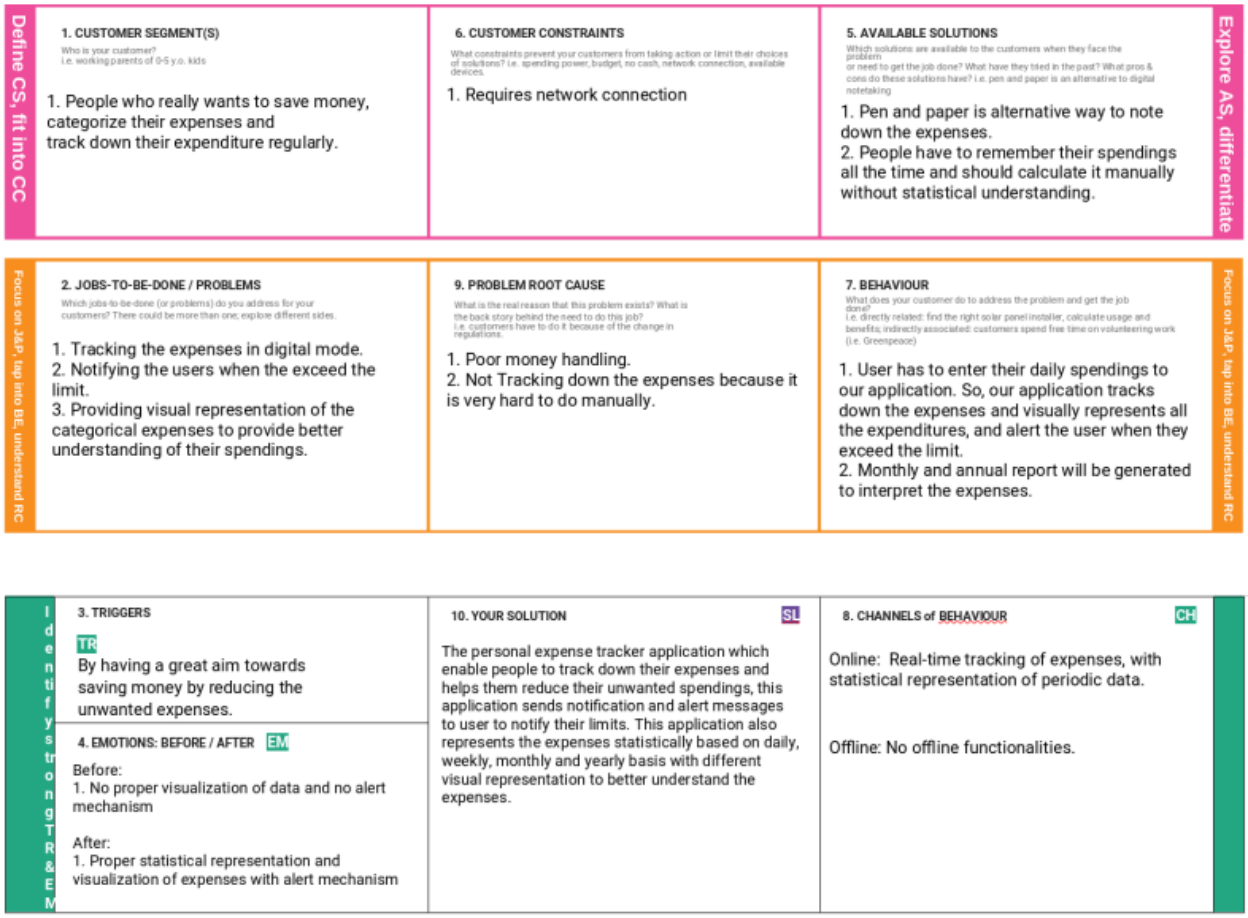
Vishnutheep B



c. Proposed Solution

S.no	Parameter	Description
1.	Problem Statement (Problem to be solved)	Normally people find it hard to keep track of their daily expenses using pen-and-paper methods, record books or in their mind. This leads to them forgetting their entire expense history and make it difficult to draw inferences from previous expenses.
2.	Idea / Solution description	To create a personal expense tracker application which manages their expense and provide insights to reduce the expenses.
3.	Novelty / Uniqueness	It will alert the user when they exceed their budget limit and user can analyze the details and a report is alsogenerated.
4.	Social Impact / Customer Satisfaction	Our application will help users to reduce their daily expenses by effectively tracking down and categorizing the expenses.
5.	Business Model (Revenue Model)	Advertisements and some premium features like money management and investing tips (only for subscribed) can be introduced to generate revenue.
6.	Scalability of the Solution	This application handles multiple users at a time as newly emerging technologies like cloud and dockers are used.

d. Problem Solution fit



4. REQUIREMENT ANALYSIS

a. Functional requirement

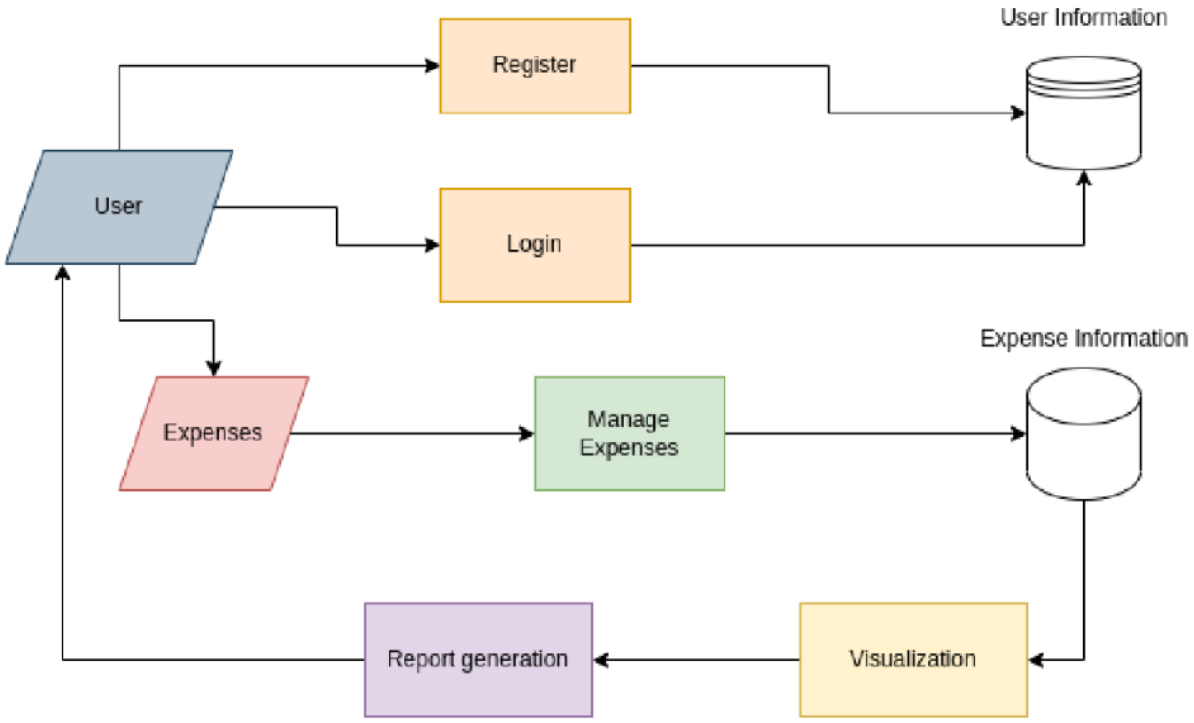
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form
FR-2	User Confirmation	Confirmation via Email
FR-3	Tracking Expense	Helpful insights about money management
FR-4	Alert Message	• Give alert mail if the amount exceeds the budget limit. • Alert through Email. • Alert through SMS.
FR-5	Report Generation	Generate monthly and yearly based report.

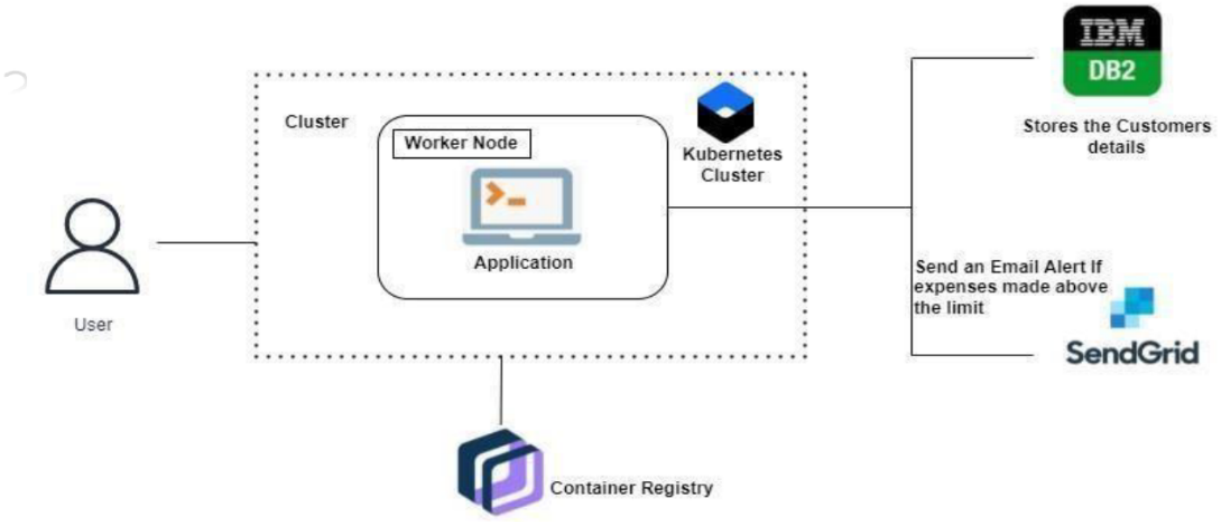
b. Non-Functional requirements

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Categorizing the expenses and to cut down the unnecessary spending
NFR-2	**Security**	It employs the latest security and technology measures to keep customers personal and financial information safe.
NFR-3	**Reliability**	Error – free application which helps the user to track and manage their expenses. Application should serve with minimal failures.
NFR-4	**Performance**	Smooth and responsiveness of the application.
NFR-5	**Availability**	People from all over the world should be able to access the application any time without any restrictions.
NFR-6	**Scalability**	To scale the capacity of the application to serve large community.

5. PROJECT DESIGN

a. Data Flow Diagram


User Information
Register
User
Login
Expense Information
Expenses
Manage Expenses
Report generation
Visualization


IBM
DB2
Stores the Customers details
Cluster
Worker Node
Kubernetes Cluster
Application
User
Send an Email Alert If expenses made above the limit
SendGrid
Container Registry

c. User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	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
Customer	Account Activation	USN-2	As a user, I will receive confirmation emailonce I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
Customer	Login	USN-3	As a user, I can log into the application by entering email & password	I can login on correct credentials	High	Sprint-1
Customer	Dashboard	USN-4	As a user, I can add expenses and income	I can enter the amount and category to save	High	Sprint-2
Customer	Dashboard	USN-5	As a user, I can change my password	I can change password if i'm logged in	Low	Sprint-4
Customer	Dashboard	USN-6	As a user, I can view my past expenses andincome over a period of time, and visualize them	I can view past records provided if records available	High	Sprint-2
Customer	Notification	USN-7	As a user, When my monthly limit exceeds, i get a email notification	I can get alert notification, if i have set a limit	Low	Sprint-4
Customer	Dashboard	USN-8	As a user, I can set a monthly expense limit	I can set a valid limit	Low	Sprint-4
Customer	Forgot password	USN-9	As a user, I can get a reset password link if iforget it through mail	I need to have access tomy email	High	Sprint-3
Customer	Forgot password	USN-10	As a user, I can change my password if I forget it	The link should be valid	High	Sprint-3

6. PROJECT PLANNING & SCHEDULING

a. Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint - 1	Registration	USN -1	User should register for the application by entering their username, email and password	4	High	Sivakumar TV Vishnutheep B
		USN -2	User, will receive confirmation email once they registered for the application.	3	Low	
	Login	USN -3	User should be able to log into the application by entering the provided email and password	4	High	Akshay Ajit Santhosh Kumar S
	Dashboard	USN -4	On successful login, user should land on the dashboard page.	2	Low	
Sprint - 2		USN -5	User should able to add expense, they should give information regarding expense_name, amount, payment mode, payment category	2	Medium	Vishnutheep B Santhosh Kumar S
		USN -6	As a user , I will set a target/limit to keep track of my expenditure.	2	Medium	Sivakumar T V Akshay Ajit
	Workspace	USN -7	Workplace for personal expense tracking	6	Medium	Sivakumar T V Santhosh Kumar S
	Charts	USN -8	Graphs to show weekly and everyday expenditure	6	High	Vishnutheep B Akshay Ajit

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint - 3	IBM DB2	USN - 9	Linking database with dashboard	3	High	Akshay Ajit Sivakumar T V
		USN -10	Making dashboard interactive with JS	3	High	Akshay Ajit Santhosh Kumar S
	SendGrid	USN -11	Sending mail to the user once they have crossed their limits, and their reports can also be sent to their mail.	5	Medium	Vishnutheep B Santhosh Kumar S
	PDF	USN - 12	Daily, Weekly, Yearly based report can be downloaded as PDF	3		Vishnutheep B Sivakumar T V
Sprint - 4	Integration	USN -13	Integrating frontend and backend.	3	High	Akshay Ajit Santhosh Kumar S
	Docker	USN -14	Creating Docker image of web app.	2	High	Vishnutheep B Akshay Ajit
	Cloud Registry	USN -15	Uploading docker image to IBM cloud registry.	3	High	Vishnutheep B Sivakumar T V
	Kubernetes	USN -16	Creating container using docker and hosting the webapp.	5	High	Akshay Ajit Vishnutheep B
	Exposing Deployment	USN -17	Exposing IP/Ports for the site.	1	Medium	Sivakumar T V Santhosh Kumar S

b. Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	28 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	6 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

c. Reports from JIRA

CODING & SOLUTIONING

a. Feature 1 - Sending report via mail

#GENERATE REPORT DAILY/YEARLY/MONTHLY

```
def generateReport(report_type):
    if(session.get('id') == None):
        return

    id = str(session['id'])

    total=0
    t_food=0
    t_entertainment=0
```

```
    t_business=0
    t_rent=0
    t_EMI=0
    t_other=0
    expense = []
    texpense = []

    if(report_type == 'Today'):
        sql = 'SELECT TIME(date), AMOUNT FROM EXPENSES WHERE USER_ID = {} AND
DATE(date) = DATE(NOW())'.format(id)
        df = pd.read_sql(sql,pd_conn)
        texpense = df.values.tolist()

        sql = 'SELECT * FROM EXPENSES WHERE USER_ID={} AND DATE(DATE) =
(CURRENT_DATE) ORDER BY AMOUNT DESC, DATE DESC'.format(id)
        df = pd.read_sql(sql,pd_conn)
        expense = df.values.tolist()

    elif(report_type == 'Monthly'):
        sql = 'SELECT DATE(date), SUM(AMOUNT) FROM EXPENSES WHERE USER_ID = {} AND
MONTH(DATE(date)) = MONTH(CURRENT_DATE) GROUP BY DATE(date)'.format(id)
        df = pd.read_sql(sql,pd_conn)
        texpense = df.values.tolist()

        sql = 'SELECT * FROM EXPENSES WHERE USER_ID={} AND MONTH(DATE(DATE)) =
MONTH(CURRENT_DATE) ORDER BY AMOUNT DESC, DATE DESC'.format(id)
        df = pd.read_sql(sql,pd_conn)
        expense = df.values.tolist()

    elif(report_type == 'Yearly'):
        sql = 'SELECT YEAR(date), SUM(AMOUNT) FROM EXPENSES WHERE USER_ID = {} AND
YEAR(DATE(date)) = YEAR(CURRENT_DATE) GROUP BY YEAR(date)'.format(id)
        df = pd.read_sql(sql,pd_conn)
        texpense = df.values.tolist()

        sql = 'SELECT * FROM EXPENSES WHERE USER_ID={} AND YEAR(DATE(DATE)) =
YEAR(CURRENT_DATE) ORDER BY AMOUNT DESC, DATE DESC'.format(id)
        df = pd.read_sql(sql,pd_conn)
        expense = df.values.tolist()
```

```
    for x in expense:
        total += x[4]
        if x[6] == "food":
            t_food += x[4]

        elif x[6] == "entertainment":
            t_entertainment  += x[4]

        elif x[6] == "business":
            t_business  += x[4]
        elif x[6] == "rent":
            t_rent  += x[4]

        elif x[6] == "EMI":
            t_EMI  += x[4]

        elif x[6] == "other":
            t_other  += x[4]

    return {'texpense':texpense,'t_food':[t_food],'t_entertainment':[t_entertainment],
                    't_business':[t_business],'t_rent':[t_rent],
                    't_EMI':[t_EMI],'t_other':[t_other],'total':[total] }
```

#MAIL THE REPORT

```
@app.route('/mail/<report_type>')
def sendReportMail(report_type):
    mailConfig()
    mail = Mail(app)

    sql = 'SELECT EMAIL FROM REGISTER WHERE USER_ID={}'.format(session['id'])
    df = pd.read_sql(sql,pd_conn)
    row = df.values.tolist()
    msg = Message('Personal Expense Tracker', recipients=[row[0][0]])
    msg.body = report_type + ' Report'
```

```
    sql = 'SELECT * FROM EXPENSES WHERE USER_ID={} AND MONTH(DATE(DATE)) =
MONTH(CURRENT_DATE) ORDER BY AMOUNT DESC, DATE DESC'.format(session['id'])
    df = pd.read_sql(sql,pd_conn)
    df.drop(columns=['EXPENSE_ID','USER_ID'],inplace=True)

    rep = generateReport(report_type)
    del rep['texpense']
    df1 = pd.DataFrame(rep)
    df1.rename(columns={"t_food": "Food", "t_entertainment": "Entertainment","t_business":
"Business","t_rent": "Rent","t_EMI": "EMI","t_other":"Other","total":"TOTAL"}, inplace=True)



    html = """\
    <html>
      <head></head>
      <body>
        <p>Hi!<br>
           <b>Monthly Expenses:<b><br>
           {0}
           <br><b>Total Categorized Expenses:<b><br>
           {1}

           Regards,
        </p>
      </body>
    </html>

    """.format(df.to_html(), df1.to_html())

    msg.html = html
    mail.send(msg)

    print("mail sent successfully")
    flash("Mail sent successfully!")
    return redirect(report_type)
```

b. Feature 2 -Download Report as pdf

```
@app.route('/<report_type>')
```

```
def downloadPDF(report_type):
    rep = generateReport(report_type)
    html = render_template("MailPDF.html", type=report_type, texpense = rep['texpense'],  total =
rep['total'][0] ,
                        t_food = rep['t_food'][0],t_entertainment =  rep['t_entertainment'][0],
                        t_business = rep['t_business'][0],  t_rent =  rep['t_rent'][0],
                        t_EMI = rep['t_EMI'][0],  t_other =  rep['t_other'][0] )

    pdf = pdfkit.from_string(html, False)
    response = make_response(pdf)
    response.headers["Content-Type"] = "application/pdf"
    response.headers["Content-Disposition"] = "inline; filename=output.pdf"

    return response


def mailConfig():
    SENDGRID_API_KEY=
"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"
    MAIL_DEFAULT_SENDER= "sparklingvishnu@gmail.com"

    app.config['SECRET_KEY'] = 'top-secret!'
    app.config['MAIL_SERVER'] = 'smtp.sendgrid.net'
    app.config['MAIL_PORT'] = 587
    app.config['MAIL_USE_TLS'] = True
    app.config['MAIL_USERNAME'] = 'apikey'
    app.config['MAIL_PASSWORD'] = SENDGRID_API_KEY
    app.config['MAIL_DEFAULT_SENDER'] = MAIL_DEFAULT_SENDER
```

c. Feature-3 Sending alert mail if the limit exceeds

CODE:

```
def sendLimitAlert(exceded_amt):
    mailConfig()
    mail = Mail(app)
```

```
    sql = 'SELECT EMAIL FROM REGISTER WHERE USER_ID={}'.format(session['id'])
    df = pd.read_sql(sql,pd_conn)
    row = df.values.tolist()
    msg = Message('Personal Expense Tracker', recipients=[row[0][0]])
    msg.body = 'Exceeded the Limit!!! You have exceeded by ' + str(exceded_amt)


    sql = 'SELECT * FROM EXPENSES WHERE USER_ID={} AND MONTH(DATE(DATE)) =
MONTH(CURRENT_DATE) ORDER BY AMOUNT DESC, DATE DESC'.format(session['id'])
    df = pd.read_sql(sql,pd_conn)
    df.drop(columns=['EXPENSE_ID','USER_ID'],inplace=True)

    rep = generateReport('Monthly')
    del rep['texpense']
    df1 = pd.DataFrame(rep)
    df1.rename(columns={"t_food": "Food", "t_entertainment": "Entertainment","t_business":
"Business","t_rent": "Rent","t_EMI": "EMI","t_other":"Other","total":"TOTAL"}, inplace=True)



    html = """\
    <html>
      <head></head>
      <body>
        <p>Hi!<br>
           <b>Total Categorized Expenses<b><br>
           {0}
           <br><b>Monthly Expenses:<b><br>
           {1}

           Regards,
        </p>
      </body>
    </html>

    """.format(df1.to_html(), df.to_html())

    msg.html = html
    mail.send(msg)

    print("mail sent successfully")
```

flash("Mail sent successfully!")

d.Database Schema

8. TESTING
a. Test Cases
b. User Acceptance Testing

9. RESULTS
a. Performance Metrics

10.ADVANTAGES & DISADVANTAGES

a.Advantages

i) Personal expense tracker helps the user to maintain their expenses and sends email if the user exceeds their budget limit that they have setted.

ii) This web application helps the users' visualize the expenses in terms of graphs which will provide a clear understanding of how much they spent weekly monthly yearly

iii) The users can set the budget limit for their expenses if they exceed their limit user will get email alert about it.

b.Disadvantages

i) The users have to manually enter their expenses

ii) Users can't search for specific expenses on a specific day

11.CONCLUSION:

The personal expense tracker web application has the functionalities of adding expenses,visualizing the expenses,viewing the expenses history, setting the budget limit and sending an email alert when the budget exceeds the limit. We can download the report or mail the report generated through this web application therefore by using the web application the users can optimise their spending habits and save money This web application developed through the ibm created the oppurtunity to learn dockers,flask, html,and css.

12.FUTURE SCOPE:

- The web application can be modified where users can search for their expenses based on the date or category

- Their expenses can be taken from bank transaction which would automate the process of manually entering the expenses

13.APPENDIX

SOURCE CODE:

- **https://github.com/IBM-EPBL/IBM-Project-6575-1658831974/tree/main/Final%20Deliverables/Code**

PROJECT DEMO LINK : http://169.51.207.139:31405/
GITHUB LINK: https://github.com/IBM-EPBL/IBM-Project-6575-1658831974