

## **IBM – NALAIYA THIRAN PROJECT**

### **PERSONAL EXPENSE TRACKER APPLICATION**

**INDUSTRY MENTOR :** KUSHBOO

**FACULTY MENTOR :** VENGATESHWARAN M

**TEAM ID :** PNT2022TMID02796

**TEAM LEAD :** SATHISH KUMAR P

**TEAM MEMBER :** SANTHOSH P

**TEAM MEMBER :** SATHIYANARAYANA V

**TEAM MEMBER :** SATISH KUMAR V

## TABLE OF CONTENT

CHAPTER	CONTENTS	PAGE NO
1	<b>INTRODUCTION</b> 1.1 PROJECT OVERVIEW 1.2 PURPOSE	04
2	<b>LITERATURE SURVEY</b> 2.1 EXISTING PROBLEM 2.2 REFERENCES 2.3 PROBLEM STATEMENT DEFINITION	06
3	<b>IDEATION &amp; PROPOSED SOLUTION</b> 3.1 EMPATHY MAP CANVAS 3.2 IDEATION & BRAINSTROMING 3.3 PROPOSED SOLUTION 3.4 PROBLEM SOLUTION FIT	15
4	<b>REQUIREMENT ANALYSIS</b> 4.1 FUNCTIONAL REQUIREMENT 4.2 NON-FUNCTIONAL REQUIREMENTS	18
5	<b>PROJECT DESIGN</b> 5.1 DATA FLOW DIAGRAMS 5.2 SOLUTION & TECHNICAL ARCHITECTURE 5.3 USER STORIES	20
6	<b>PROJECT PLANNING &amp; SCHEDULING</b> 6.1 SPRINT PLANNING & ESTIMATION 6.2 REPORTS FROM JIRA	22

<b>7</b>	<b>CODING &amp; SOLUTIONING</b> 7.1 FEATURE 1 7.2 FEATURE 2	<b>24</b>
<b>8</b>	<b>TESTING</b> 8.1 TEST CASES	<b>35</b>
<b>9</b>	<b>RESULTS</b> 9.1 PERFORMANCE METRICS	<b>36</b>
<b>10</b>	<b>ADVANTAGES &amp; DISADVANTAGES</b>	<b>41</b>
<b>11</b>	<b>CONCLUSION &amp; FUTURE SCOPE</b>	<b>43</b>
<b>12</b>	<b>APPENDIX</b> GITHUB & PROJECT DEMO LINK	<b>44</b>

# **1.INTRODUCTION**

## **1.1 Project overview**

With the launch and increase in sales of smartphones over the last few years, people are using mobile applications to get their work done, which makes their lives easier. Mobile applications comprise various different categories such as Entertainment, Sports, Lifestyle, Education, Games, Food and Drink, Health and Fitness, Finance, etc.

This Expense Tracker application falls in the Finance Category and serves the important purpose of managing finances which is a very important part of one's life.

The software product went through the design, development, and the testing phase as a part of the Software Development Lifecycle.

The application's interface is designed using custom art elements, the functionality is implemented using iOS SDK, and the phase of testing the product was accomplished successfully. The application is not much user intensive but just comprises of having them enter the expense amount, date, category, merchant and other optional attributes (taking picture of the receipts, entering notes about the expense, adding subcategories to the categories). With this entered information, the user is able to see the expense details daily, weekly, monthly, and yearly in figures, graphs, PDF format, and can print

them as well if a printer is detected or scanned nearby. All these topics have been explained in detail in their respective chapters.

The aim of this project is to provide a solution for users on how to manage finances in any circumstance by keeping track of their expenses everyday. Ultimately, this contributes to societal well-being.

## 1.2 Purpose

The motivation to work in this project is actually our real-life experience. As a user We face many difficulties in our daily file. In our daily life money is the most important portion and without it we cannot last one day on earth but if we keep on track all financial data then we can overcome this problem.

Most of the people cannot track their expenses and income one way they face the money crisis and depression. This situation motivates us to make an android app to track all financial activities. Using the Daily Expense Tracker user can be tracking expenses day to day and making life tension free.

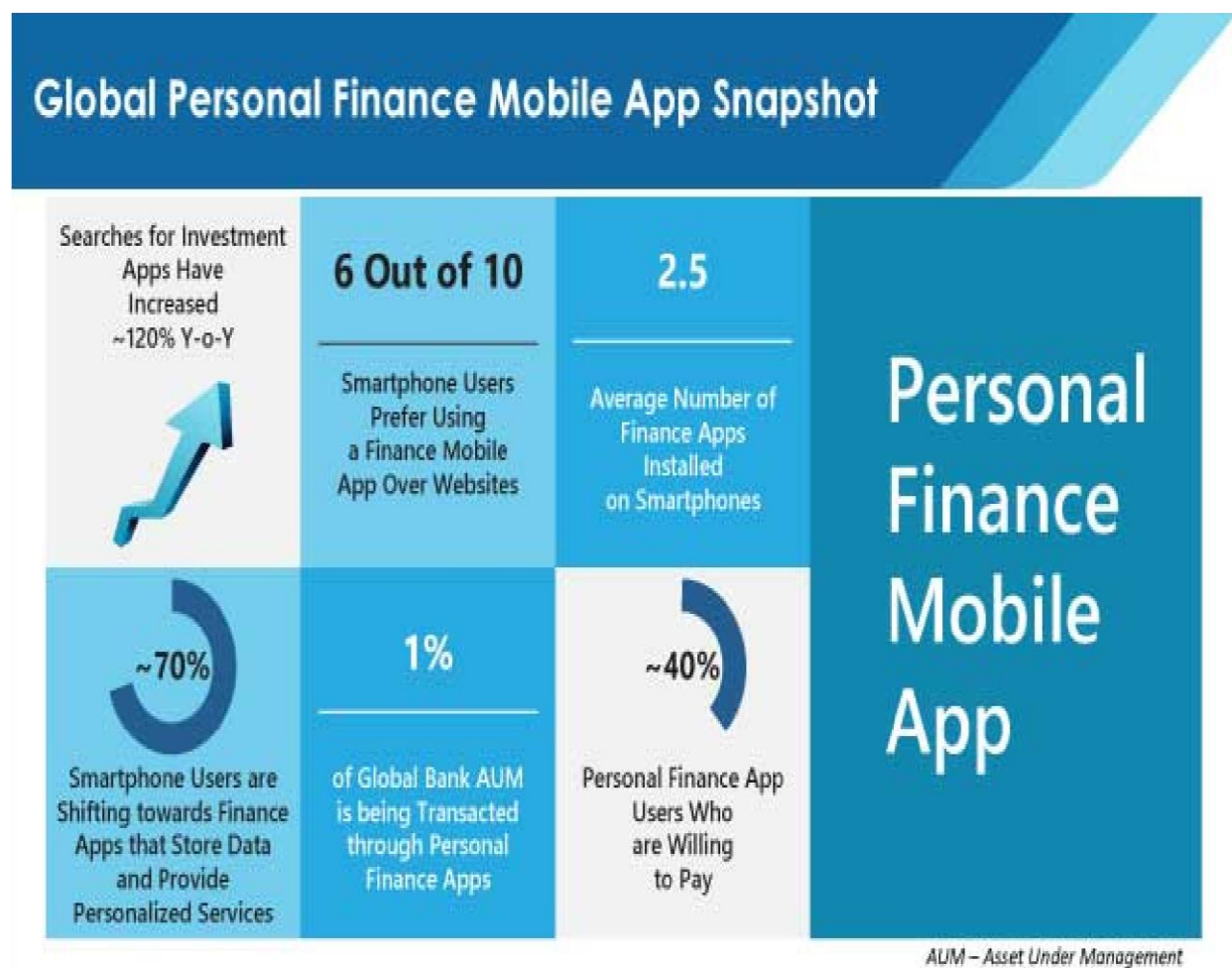
A comprehensive money management strategy requires clarity and conviction for decision-making. You will need a defined goal and a clear vision for grasping the business and personal finances. That's when an expense tracking app comes into the picture.

An expense tracking app is an exclusive suite of services for people who seek to handle their earnings and plan their expenses and savings

efficiently. It helps you track all transactions like bills, refunds, payrolls, receipts, taxes, etc., on a daily, weekly, and monthly basis.

## 2. LITERATURE SURVEY

### 2.1 Existing Problem



This above study shows that the people prefer and are getting comfortable in managing their finances through a mobile app and evolve from the age old paper system. They have

understood the benefits in leveraging digital tools to get better insights and a bigger look over their finances.

### **96% Indian parents feel their children lack financial know-how: Survey**



Indian teens are showing a strong inclination towards learning about new age digital financial solutions. (E+)

---

1 min read . Updated: 24 Aug 2022, 03:45 PM IST

Parents are worried that their children lack financial literacy unlike their global counterparts. They are looking for applications where their children can learn about spending money. This financial illiteracy is prevalent even among elders.

# Only 27% Adults, 16.7% of Indian Teenagers Financially Literate

• By: [Education and Careers Desk](#)

• [News18.com](#)

• Last Updated: JANUARY 11, 2022, 16:59 IST

• New Delhi



The financial literacy rate in Indian is lower than UK, Singapore and US, says the survey  
(Representative image)

The results show that respondents from South India performed the best with an average score of 47.86 per cent and east Indians underperformed with an average score of only 39.41 per cent.

• Follow us: [f Facebook](#)

[Twitter](#)

[Instagram](#)

[Telegram](#)

[Google News](#)

## Existing Solutions :

### **Money View - Expense Manager App**

Money View App reads all of the transactional SMS messages and provides you with real-time visibility into your finances. This app unearths the hidden financial data that sits idly in SMS logs and makes excellent use of it.

### **Key features of the App:**

- Check your bank account balances.
- See the most recent banking transactions.



- The Money View app automatically categorises your payments and displays major areas of spending.
- View weekly and monthly summaries to help you avoid overspending and improve the efficiency of your budget planning.
- It keeps track of your spending, sends personalised bill-paying reminders, finds relevant savings opportunities, and much more.
- Track your financial progress by looking at your spending trends over time.

### **Good budget - Budget & Finance App**

This personal finance manager app acts as a proactive budget planner, assisting you in staying on top of your budget, bills, and finances. The personal finance app was designed for simple, real-time budget and financial tracking, making it one of the best expense tracker apps in India.

#### **Key Features of the App:**

- Data is backed up automatically and securely to Good budget's website.
- Disaggregate expense transactions
- Transactions that are scheduled and envelope fills
- Save time by using intelligent payee and category suggestions.
- Transfer funds between Envelopes and Accounts with ease.
- Match the budget period to the real-life situation.
- Analyse spending with the Spending by Envelope Report.
- Use the Income vs. Spending Report to keep track of your cash flow.

- Export transactions to CSV
- Carryover any unused funds to the next month as a reward for your incredible self-control.
- Plan your finances ahead of time to stay on track with your budget.

### **Real byte Money Manager App**

You can use the budget planner and spending tracker to keep track of your personal and business financial transactions, review financial data on a daily/weekly/monthly basis, and manage your assets.

#### **Key Features of the App:**

- System of double-entry bookkeeping
- Management of budgets and expenses • Management of credit and debit cards • Get access to statistics immediately.
- Bookmarking feature
- Backup/restore function

### **Money - Budget Manager and Expense Tracker App**

Money tracks the user's expenses and compares them to the monthly income and the budget planner. Money's money manager app keeps your monthly budget in top shape. As a result, it could also serve as the best expense tracker app.

#### **Key Features of the App:**

- With the intuitive and simple-to-use interface, you can quickly add new records.
- Maintain a multi-currency track.
- Backup and export personal finance data with a single click.
- Protect your data with passcode protection.

- View your spending distribution on a simple chart, or get detailed information from the records list.
- Use a budget tracker to save money.
- Use your own Google Drive or Dropbox account to safely synchronise.
- Take control of recurring payments.
- Create multiple accounts.
- Use the built-in calculator to crunch numbers.

## **Wallet - Money, Budget, Finance & Expense Tracker App**

Wallet can automatically track your daily expenses by syncing your bank account, view weekly expense reports, plan your shopping expenses, and share specific features with your loved ones. You can manage your money with a wallet from anywhere and at any time.

### **Key Features of the App:**

- Transactions are automatically and securely synced, then intelligently categorised and budgeted.
- Simple graphs and financial overviews provide actionable insights into the state of your finances, including accounts, credit and debit cards, debts, and cash.
- Arrange your bills and keep track of their due dates.
- Examine upcoming payments and how they will affect your cash flow.

- Selected accounts can be shared with family, friends, or co-workers who need to work together on a budget. Everyone is welcome to contribute from any platform, including Android, iPhone, and the Web.
- Other features include support for multiple currencies, automatic cloud sync, receipt and warranty tracking, categories and templates, geo-mapping transactions, hash-tagging, shopping lists, exports to CSV/XLS/PDF, debt management, PIN security, standing orders, notifications, reports, and more.

## **Walnut - All Indian Banks Money Manager App**

Walnut automates and secures the tracking of your monthly expenses. You can stay within your budget, pay your bills on time, and save more money each month by using the Walnut app. They also provide personal loans.

### **Key Features of the App:**

- Keep a close eye on your credit card balances.
- Use BHIM UPI to send money.
- Locate ATMs that accept cash near you in real-time.
- Export your information and create expense reports (in PDF & CSV format).
- Verify the balance of your bank account.

- Keep track of train, cab, movie, and event reservations, among other things.
- Search and share information about places you visit with friends and social networks.
- Report your bank, card, or any other interesting messages directly from the app.

## 2.2 References

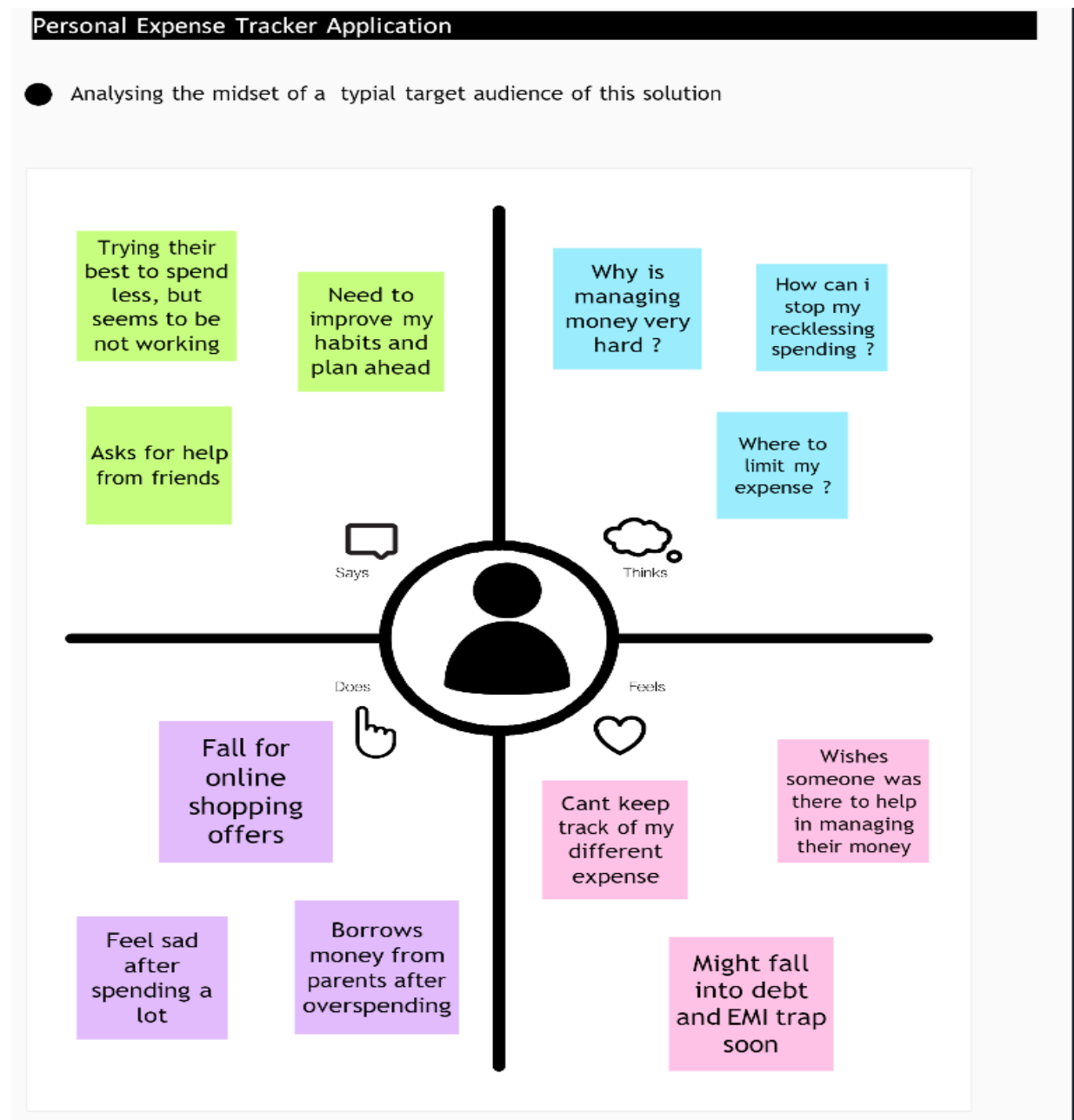
1. <https://moneyview.in/insights/best-personal-finance-management-apps-in-india>
2. <https://www.factmr.com/report/personal-finance-mobile-app-market>
3. <https://www.moneytap.com/blog/best-money-management-apps/>
4. <https://relevant.software/blog/personal-finance-app-like-mint/>
5. <https://www.onmanorama.com/lifestyle/news/2022/01/11/financial-literacy-trend-among-todays-youth-investment.html>
6. <https://www.livemint.com/money/personal-finance/96-indian-parents-feel-their-children-lack-financial-know-how-survey-11661336110855.html>
7. <https://economictimes.indiatimes.com/small-biz/money/importance-of-financial-literacy-amongst-youngsters/articleshow/85655134.cms>

### 2.3 Problem Statement Definition

- Modern education does not focus on finance management. This is primarily due to lack of resources and the Indian value system on giving money to children. Failing to teach this valuable knowledge had left many Indians to recklessly spend their income and fall into vicious cycles of EMI and debt.
- Many of them are just a month's salary away from bankruptcy. This issue is tackled by providing a web application for where people can plan their monthly expenses into categories, set alerts and get visual insights from their spending patterns.

### 3. IDEATION AND PROPOSED SOLUTION

#### 3.1 Empathy Map Canvas



## 3.2 Ideation & Brainstorming

### Brainstorm & idea prioritization

Use this template in your own brainstorming sessions as your team generates ideas. Organize and sort ideas into groups. You can use this template to generate ideas for a project or to generate ideas for a business idea.

1. Generate ideas

2. Sort ideas

3. Generate ideas

#### 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

Modern education does not focus on finance management. This is primarily due to lack of resources and the Indian value system on giving money to children. Failing to teach this valuable knowledge had left many Indians to recklessly spend their income and fall into vicious cycles of debt and debt. Many of them are just a month's salary away from bankruptcy.

This issue is tackled by providing a web application for where people can plan their monthly expenses into categories, set alerts and get visual insights from their spending patterns.

#### Brainstorm

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

5 minutes

**Sathya Kumar V**

- Build Desktop Budgeting App
- Support Visual Graphs
- Generate monthly expense reports and expenditure limits
- Multiple login methods
- Provide multi-user support
- Secure personal information
- Generate PDF reports
- Export data as CSV
- Send email notifications

**Sathya Kumar P**

- Lightweight Web App
- Allow categories for expenditure
- Multiple login methods
- Provide multi-user support
- Secure personal information
- Generate PDF reports
- Export data as CSV
- Send email notifications

**Sathya Kumar V**

- Support the desktop app
- Generate monthly reports
- Send email notifications
- Secure personal information
- Generate PDF reports
- Export data as CSV
- Send email notifications

**Sathya Kumar P**

- Generate monthly reports
- Send email notifications
- Secure personal information
- Generate PDF reports
- Export data as CSV
- Send email notifications

#### Group Ideas

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

5 minutes

**Sathya Kumar V**

- Build Desktop Budgeting App
- Support Visual Graphs
- Generate monthly expense reports and expenditure limits
- Multiple login methods
- Provide multi-user support
- Secure personal information
- Generate PDF reports
- Export data as CSV
- Send email notifications

**Sathya Kumar P**

- Lightweight Web App
- Allow categories for expenditure
- Multiple login methods
- Provide multi-user support
- Secure personal information
- Generate PDF reports
- Export data as CSV
- Send email notifications

**Sathya Kumar V**

- Support the desktop app
- Generate monthly reports
- Send email notifications
- Secure personal information
- Generate PDF reports
- Export data as CSV
- Send email notifications

**Sathya Kumar P**

- Generate monthly reports
- Send email notifications
- Secure personal information
- Generate PDF reports
- Export data as CSV
- Send email notifications

#### Prioritize

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

5 minutes

**Sathya Kumar V**

- Build Desktop Budgeting App
- Support Visual Graphs
- Generate monthly expense reports and expenditure limits
- Multiple login methods
- Provide multi-user support
- Secure personal information
- Generate PDF reports
- Export data as CSV
- Send email notifications

**Sathya Kumar P**

- Lightweight Web App
- Allow categories for expenditure
- Multiple login methods
- Provide multi-user support
- Secure personal information
- Generate PDF reports
- Export data as CSV
- Send email notifications

**Sathya Kumar V**

- Support the desktop app
- Generate monthly reports
- Send email notifications
- Secure personal information
- Generate PDF reports
- Export data as CSV
- Send email notifications

**Sathya Kumar P**

- Generate monthly reports
- Send email notifications
- Secure personal information
- Generate PDF reports
- Export data as CSV
- Send email notifications

## 3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Building a personal finance tracking application that will imbibe good spending habits into students.
2.	Idea / Solution description	To build a web application that is deployed in IBM cloud and leverage mailing service like send grid to implement the same
3.	Novelty / Uniqueness	The stats generated with visual graphs are more effective than log books. It also helps in using technology to gain better insights from patterns.
4.	Social Impact / Customer Satisfaction	Better financial knowledge is gained. Gamified approach can be used to give self satisfaction. Reduced chances of bad debt in future.
5.	Business Model (Revenue Model)	Subscription can be incorporated to access premium tools within the app.
6.	Scalability of the Solution	As the application is containerized for deployment. It can be easily scaled in a cloud service provider like IBM.



## 3.1 Problem Solution fit

Define CS, fit into CC

### 1. CUSTOMER SEGMENT(S)

Who is your customer?

Predominantly Engineers who are just starting to earn and manage their personal finance. Typically from middle and lower class family, who badly need financial discipline.

CS

### 6. CUSTOMER CONSTRAINTS

What constraints prevent your customers from taking action or limit their choices

of solutions?

The impulse buying and lacking to awareness to

CC

### 5. AVAILABLE SOLUTIONS

Which solutions are available to the customers when they face the problem

Totally shunning to spend even on necessities under the impression that the spending could result in bad financial position.

The existing solutions are otherwise over complicated and designed to extract data from user.

Manual physical logging in time consuming.

AS

Explore AS, differentiate

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

### 2. JOBS-TO-BE-DONE / PROBLEMS

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

- ❖ Logging expenses into categories
- ❖ Show historical stats
- ❖ Generate insightful charts
- ❖ Alert user to imbibe good discipline.

J&P

### 9. PROBLEM ROOT CAUSE

What is the real reason that this problem exists?

Lack of proper education in financial literacy in school education. More children are not given pocket money to learn by spending/wasting less/saving.

RC

### 7. BEHAVIOUR

What does your customer do to address the problem and get the job done?

Get frustrated and fall into debt traps by taking unpayable loans for unnecessary items leading to increase in mental stress

BE

**3. TRIGGERS****TR**

What triggers customers to act?

Frequent sales in e-commerce platforms and seamless shopping experience online.

**4. EMOTIONS: BEFORE / AFTER****EM**

How do customers feel when they face a problem or a job and afterwards?

Dejected and paranoid about the future as they would need relatively more money to provide for a family and to handle unexpected financial needs.

**10. YOUR SOLUTION****SL**

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.

If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

Graphical Application with simple UI and to the point clutter free objective. Avoids provision to pay through the app, to minimize the spending and ensure that only necessary spendings are made. The aim is to make the spending process harder throughout the application and keep it clean.

**8. CHANNELS of BEHAVIOUR****CH****8.1 ONLINE**

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

1. Shop from e-commerce
2. Subscribe to OTT platforms
3. Order food frequently

**8.2 OFFLINE**

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

1. Shop in malls during sales
2. Keep the money somewhere around and forget about/lose it

## 4. REQUIREMENT ANALYSIS

### 4.1 Functional requirement

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Email/SignUp Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Add expenses	Enter the everyday expenses Split it into categories(example : food, petrol,movies)
FR-4	Reminder mail	Sending reminder mail on target (for ex : if user wants a reminder when his/her balance reaches some amount(5000)) Sending reminder mail to the user if he/she has not filled that day's expenses.
FR-5	Creating Graphs	Graphs showing everyday and weekly expenses. Categorical graphs on expenditure.
FR-6	Add salary	Users must enter the salary at the start of the month.
FR-7	Export CSV	User can export the raw data of their expenditure as CSV

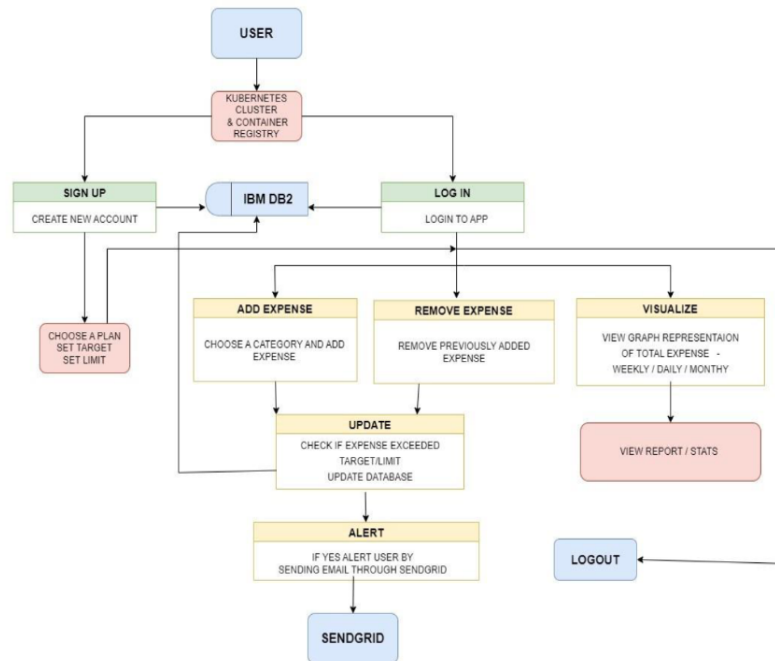
## 4.2 Non-Functional requirements

FR No.	Non-Functional Requirement	Description
NFR-1	<b>Usability</b>	A simple web application which is accessible across devices
NFR-2	<b>Security</b>	The OAuth Google sign in and email login are secure with hashed and salted secure storage of credentials.
NFR-3	<b>Reliability</b>	Containerized service ensures that new instance can kick up when there is a failure
NFR-4	<b>Performance</b>	The load is managed through the load balancer used with docker. Thus ensuring good performance
NFR-5	<b>Availability</b>	With load balancing and multiple container instances, the service is always available.
NFR-6	<b>Scalability</b>	Docker and Kubernetes are designed to accommodate scaling based on need

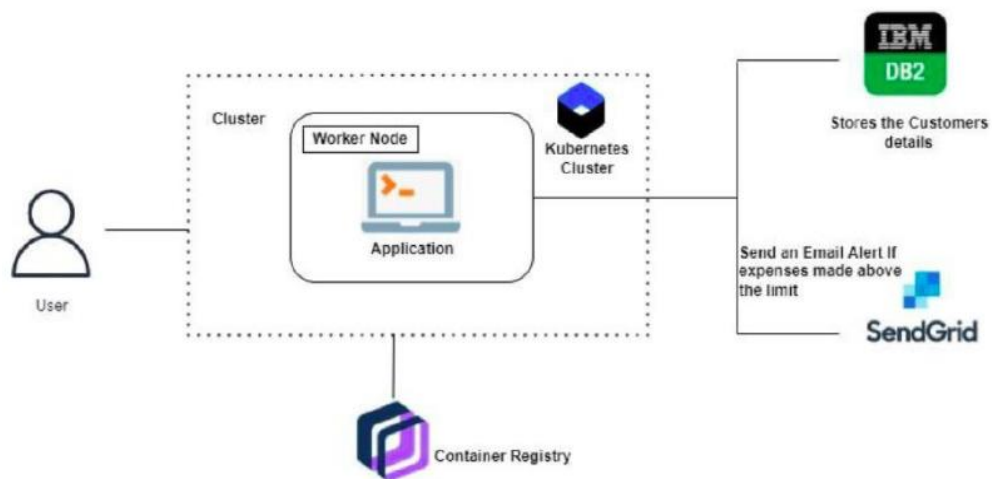
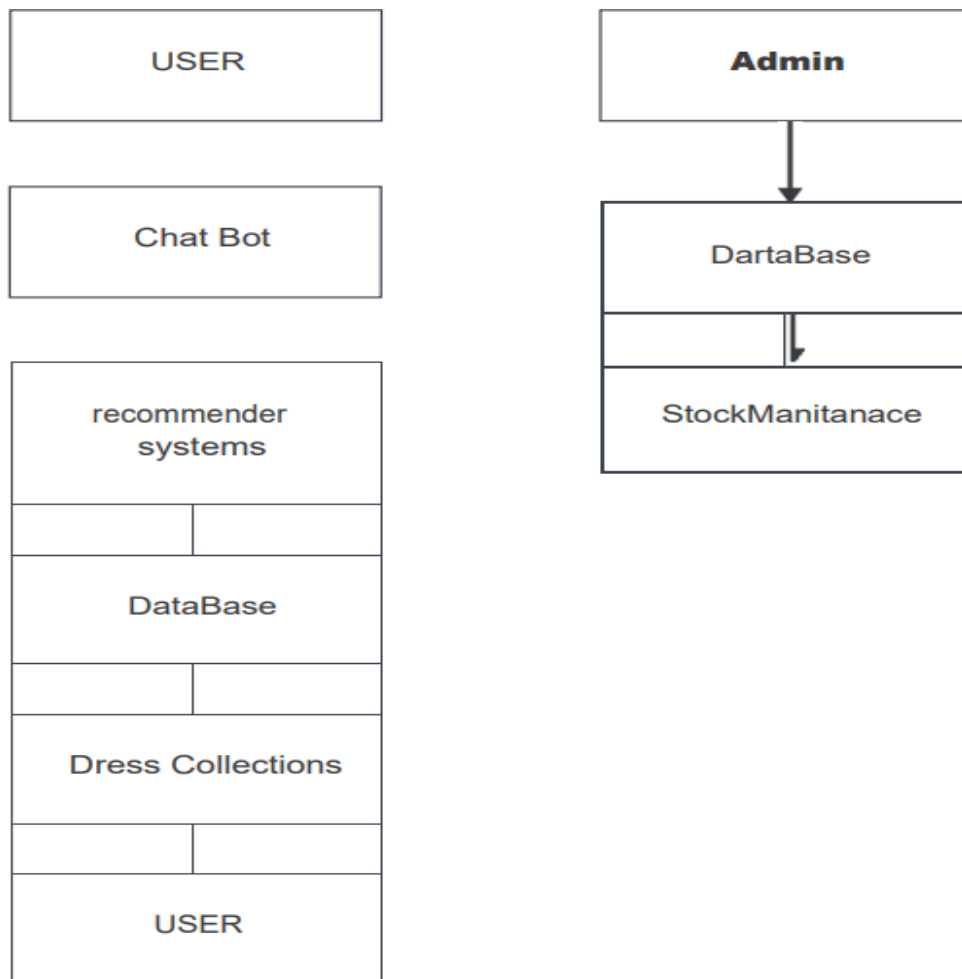
## 5. PROJECT DESIGN

### 5.1 Data Flow Diagrams

Data Flow Diagram of Personal Expense Tracker: DFD Level 2



### 5.2 Solution & Technical Architecture



## 5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.
	Login	USN-2	As a user, I can log into the application by entering email & password
	Add	USN -3	As a user, I can add in new expenses.
	Remove	USN - 4	As a user, I can remove previously added expenses.
	View	USN - 5	As a user, I can view my expenses in the form of graphs and get insights.
	Get alert message	USN - 6	As a user, I will get alert messages if I exceed my target amount.
Administrator	Add / remove user	USN - 7	As admin, I can add or remove user details on db2 manually.
		USN - 8	As admin, I can add or remove user details on sendgrid.

## 6. PROJECT PLANNING

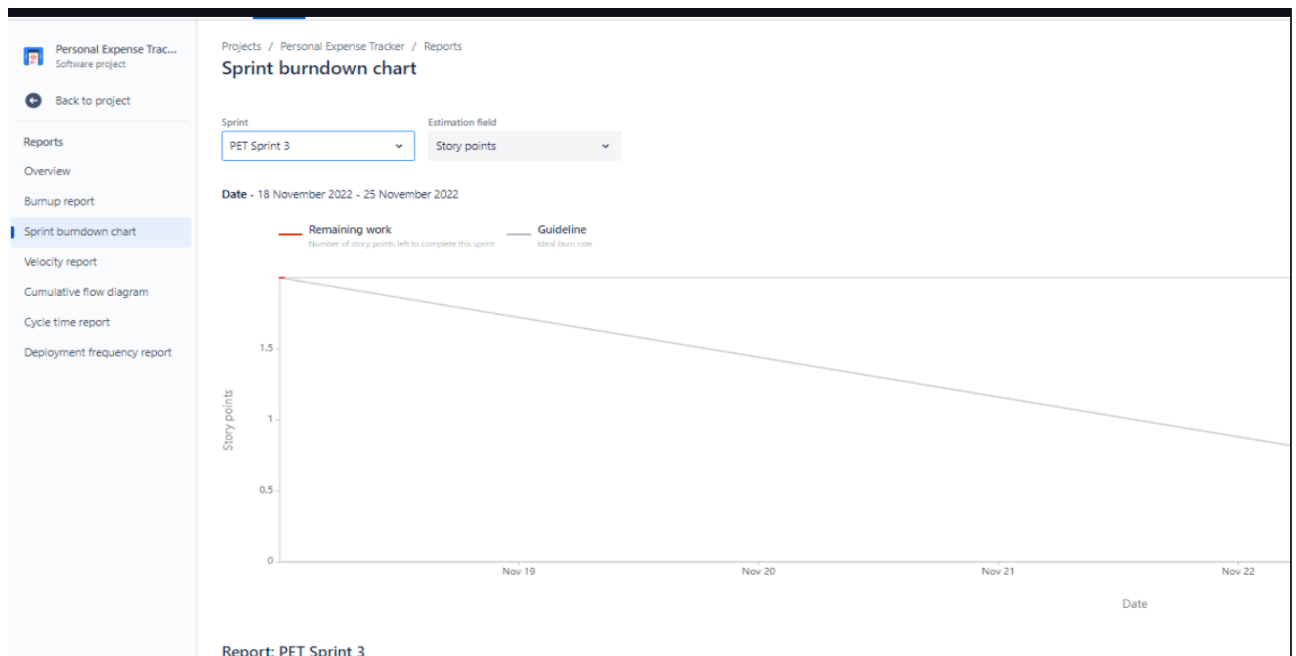
### 6.1 Sprint Planning & Estimation

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 entering my email, new password and confirming the same password.	2	High	Santhosh
		USN -2	As a user, I will receive confirmation email once I have registered for the application.	1	Low	Sathish P
	Login	USN -3	As a user, I can log into the application by entering email and password / Google OAuth.	2	High	SathyaNaryanan
	Dashboard	USN -4	Logging in takes the user to their dashboard.	1	Low	Satish kumar V
Sprint - 2		USN -5	As a user, I will update my salary at the start of each month.	1	Medium	Sathish Kumar P
		USN -6	As a user, I will set a target/limit to keep track of my expenditure.	1	Medium	Satish kumar V
	Workspace	USN -7	Workplace for personal expense tracking	1	Medium	SathyaNaryanan
	Charts	USN -8	Graphs to show weekly and everyday expenditure	2	High	Santhosh
		USN -9	As a user, I can export raw data as csv file.	1	Medium	SathyaNaryanan

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint - 3	IBM DB2	USN -10	Linking database with dashboard	2	High	Satish kumar V
		USN -11	Making dashboard interactive with JS	2	High	Sathish P
	Watson Assistant	USN -12	Embedding Chatbot to clarify user's queries.	1	Low	SathyaNaryanan
	BCrypt	USN -13	Using BCrypt to store passwords securely.	1	Medium	Santhosh
	SendGrid	USN -14	Using SendGrid to send mail to the user. (To alert or remind)	1	Medium	Satish kumar V
Sprint - 4	Integration	USN -15	Integrating frontend and backend.	2	High	SathyaNaryanan
	Docker	USN -16	Creating Docker image of web app.	2	High	Santhosh
	Cloud Registry	USN -17	Uploading docker image to IBM cloud registry.	2	High	Satish kumar V
	Kubernetes	USN -18	Creating container using docker and hosting the webapp.	2	High	SathyaNaryanan
	Exposing Deployment	USN -19	Exposing IP/Ports for the site.	1	Medium	Sathish P

## 6.2 Reports from JIRA

### Burnt Down Chart



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

### 7.1 Feature 1

```
from flask import Flask, render_template, request, redirect, session
import re
import sendgrid
from flask_db2 import DB2
import ibm_db
import ibm_db_dbi
import os
from sendmail import sendmail
app = Flask(__name__)
app.secret_key = 'a'
app.config['database'] = 'bludb'
app.config['hostname'] = 'b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud'
app.config['port'] = '31249'
app.config['protocol'] = 'tcpip'
app.config['uid'] = 'xgl34124'
app.config['pwd'] = 'lkP1B5zjTXYPKZUK'
app.config['security'] = 'SSL'
try:
    mysql = DB2(app)
    conn_str='database=bludb;hostname=b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;port=31249;protocol=tcpip;\uid=xgl34124;pwd=lkP1B5zjTXYPKZUK;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())
#HOME--PAGE
@app.route("/home")
def home():
```



```

    return render_template("homepage.html")
@app.route("/")
def add():
    return render_template("home.html")
#SIGN--UP--OR--REGISTER
@app.route('/signup')
def signup():
    return render_template('signup.html')
@app.route('/register', methods =['GET', 'POST'])
def register():
    if request.method == "POST":
        user_name = request.form['username']
        email = request.form['email']
        pass_word = request.form['password']
        query = "INSERT INTO Admin (username,email,password)
values (?,?)"
        insert_stmt = ibm_db.prepare(ibm_db_conn, query)
        ibm_db.bind_param(insert_stmt, 1, user_name)
        ibm_db.bind_param(insert_stmt, 2, email)
        ibm_db.bind_param(insert_stmt, 3, pass_word)
        ibm_db.execute(insert_stmt)
        msg = 'Account Created Successfully'
        return render_template("signup.html", msg=msg)
@app.route("/signin",methods=['post','get'])
def signin():
    if request.method=="post":
        return render_template("login.html")
    return render_template("login.html")
@app.route('/login',methods =['GET', 'POST'])
def login():
    global userid
    msg = "
    if request.method == 'POST' :
        username = request.form['username']
</pre

```

```

password = request.form['password']
sql = "SELECT * FROM Admin WHERE username = ? and
password = ?"
stmt = ibm_db.prepare(ibm_db_conn, sql)
ibm_db.bind_param(stmt, 1, username)
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 Admin WHERE username = " + "\"" +
username + "\"" + " and password = " + "\"" + password + "\""
res = ibm_db.exec_immediate(ibm_db_conn, param)
dictionary = ibm_db.fetch_assoc(res)
# sendmail("hello sathish","sksathish592001@gmail.com")

```

```

if account:

```

```

    session['loggedin'] = True
    session['id'] = dictionary["ID"]
    userid = dictionary["ID"]
    session['username'] = dictionary["USERNAME"]
    session['email'] = dictionary["EMAIL"]

```

```

    return redirect('/home')

```

```

else:

```

```

    msg = 'Incorrect username / password !'

```

```

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

```

```

@app.route("/add")

```

```

def adding():

```

```

    return render_template('add.html')

```

```

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

```

```

def addexpense():

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

    print(date)
    p1 = date[0:10]
    p2 = date[11:13]
    p3 = date[14:]
    p4 = p1 + "-" + p2 + "." + p3 + ".00"
    print(p4)
    sql = "INSERT INTO Expense (userid, date, expensename,
amount, paymode, category) VALUES (?, ?, ?, ?, ?, ?)"
    stmt = ibm_db.prepare(ibm_db_conn, sql)
    ibm_db.bind_param(stmt, 1, session['id'])
    ibm_db.bind_param(stmt, 2, p4)
    ibm_db.bind_param(stmt, 3, expensename)
    ibm_db.bind_param(stmt, 4, amount)
    ibm_db.bind_param(stmt, 5, paymode)
    ibm_db.bind_param(stmt, 6, category)
    ibm_db.execute(stmt)

    print("Expenses added")

    # email part

    param = "SELECT * FROM Expense WHERE MONTH(date) =
MONTH(current timestamp) AND YEAR(date) = YEAR(current
timestamp) ORDER BY date DESC"
    res = ibm_db.exec_immediate(ibm_db_conn, param)
    dictionary = ibm_db.fetch_assoc(res)

```

```

expense = []
while dictionary != False:
    temp = []
    temp.append(dictionary["ID"])
    temp.append(dictionary["USERID"])
    temp.append(dictionary["DATE"])
    temp.append(dictionary["EXPENSENAME"])
    temp.append(dictionary["AMOUNT"])
    temp.append(dictionary["PAYMODE"])
    temp.append(dictionary["CATEGORY"])
    expense.append(temp)
    print(temp)
    dictionary = ibm_db.fetch_assoc(res)

total=0
for x in expense:
    total += int(x[4])
    param = "SELECT id, limit FROM limit WHERE userid = " +
str(session['id']) + " ORDER BY id DESC LIMIT 1"
    res = ibm_db.exec_immediate(ibm_db_conn, param)
    dictionary = ibm_db.fetch_assoc(res)
    row = []
    s = 0
    while dictionary != False:
        temp = []
        temp.append(dictionary["LIMIT"])
        row.append(temp)
        dictionary = ibm_db.fetch_assoc(res)
        s = temp[len(temp)-1]
    if total > int(s):
        msg = "Hello " + session['username'] + " , " + "you have crossed
the monthly limit of Rs. " + str(s) + "/- !!!" + "\n" + "Thank you, " +
"\n" + "Team Personal Expense Tracker."
        sendmail(msg,session['email'])

```

```

#limit
@app.route("/limit" )
def limit():
    return redirect('/limitn')
@app.route("/limitnum" , methods = ['POST' ])
def limitnum():
    if request.method == "POST":
        number= request.form['number']
        # cursor = mysql.connection.cursor()
        # cursor.execute('INSERT INTO limits VALUES (NULL, % s,
% s) ',(session['id'], number))
        # mysql.connection.commit()
        sql = "INSERT INTO limit (userid, limit) VALUES (?, ?)"
        stmt = ibm_db.prepare(ibm_db_conn, sql)
        ibm_db.bind_param(stmt, 1, session['id'])
        ibm_db.bind_param(stmt, 2, number)
        ibm_db.execute(stmt)
    return redirect('/limitn')
@app.route("/limitn")
def limitn():
    # cursor = mysql.connection.cursor()
    # cursor.execute('SELECT limitss FROM `limits` ORDER BY
`limits`.`id` DESC LIMIT 1')
    # x= cursor.fetchone()
    # s = x[0]
    for x in expense:
        total += int(x[4])
        if x[6] == "food":
            t_food += int(x[4])

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

        elif x[6] == "business":

```

```

        t_business += int(x[4])
    elif x[6] == "rent":
        t_rent += int(x[4])

    elif x[6] == "EMI":
        t_EMI += int(x[4])
    elif x[6] == "other":
        t_other += int(x[4])
print(total)
print(t_food)
print(t_entertainment)
print(t_business)
print(t_rent)
print(t_EMI)
print(t_other)
return render_template("today.html", texpanse = texpanse,
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 )

temp = []
temp.append(dictionary["ID"])
temp.append(dictionary["USERID"])
temp.append(dictionary["DATE"])
temp.append(dictionary["EXPENSENAME"])
temp.append(dictionary["AMOUNT"])
temp.append(dictionary["PAYMODE"])
temp.append(dictionary["CATEGORY"])
expense.append(temp)
print(temp)
for x in expense:
    total += int(x[4])
    if x[6] == "food":
        t_food += int(x[4])

```

```

        elif x[6] == "entertainment":
            t_entertainment += int(x[4])
        elif x[6] == "business":
            t_business += int(x[4])
        elif x[6] == "rent":
            t_rent += int(x[4])
        elif x[6] == "EMI":
            t_EMI += int(x[4])
        elif x[6] == "other":
            t_other += int(x[4])
    print(total)
    print(t_food)
    print(t_entertainment)
    print(t_business)
    print(t_rent)
    print(t_EMI)
    print(t_other)
    return render_template("today.html", texpanse = texpanse,
expense = expense, total = total ,
                        t_food = t_food,t_entertainment = t_entertainment,
                        t_business = t_business, t_rent = t_rent,
                        t_EMI = t_EMI, t_other = t_other )

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

```

## 7.2 Feature 2

### Home.html

```
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">

  <link rel="stylesheet" href="..\static\css\home.css">
  <title>My Website</title>
</head>

<body>
  <!-- Header -->
  <section id="header">
    <div class="header container">
      <div class="nav-bar">
        <div class="brand">
          <a href="#hero">
            <h1><span>B</span>udget <span>T</span>racker</h1>
          </a>
        </div>
        <div class="nav-list">
          <div class="hamburger">
            <div class="bar"></div>
          </div>
          <ul>
            <li><a href="#hero" data-after="Home">Home</a></li>
            <li><a href="#services" data-
after="Service">Services</a></li>
```



```

        <li><a href="#about" data-after="About">About</a></li>
        <li><a href="#contact" data-
after="Contact">Contact</a></li>
        <LI><a href="/signin" data-after="Login">-Login-</a></LI>
    </ul>
</div>
</div>
</div>
</div>
</section>
<!-- End Header -->

```

```

<!-- Hero Section -->
<section id="hero">
    <div class="hero container">
        <div>
            <h1>Hello, <span></span></h1>
            <h1>Welcome To <span></span></h1>
            <h1>Personal Expense Tracker App <span></span></h1>
            <a href="/signup" type="button" class="cta">Sign-up</a>
        </div>
    </div>
</section>
<!-- End Hero Section -->

```

```

<!-- Service Section -->
<section id="services">
    <div class="services container">
        <div class="service-top">
            <h1 class="section-title">Serv<span>i</span>ces</h1>
            <p>Budget Tracker provides a many services to the customer
and industries. Financial solutions to meet your needs whatever your
money goals,there is a Budget solution to help you reach them </p>
        </div>
    </div>

```

```

<section id="footer">
  <div class="footer container">
    <div class="brand">
      <h1><span>B</span>udget <span>T</span>racker</h1>
    </div>
    <h2>Your Complete Financial Solution</h2>
    <div class="social-icon">
      <div class="social-item">
        <a href="#"></a>
      </div>
      <div class="social-item">
        <a href="#"></a>
      </div>
      <div class="social-item">
        <a href="#"></a>
      </div>
    </div>
    <p>Copyright © 2019-2023 GCE CSE girls . All rights
reserved</p>
  </div>
</section>
<!-- End Footer -->
<script src="..\static\js\home.js"></script>
</body>
</html>

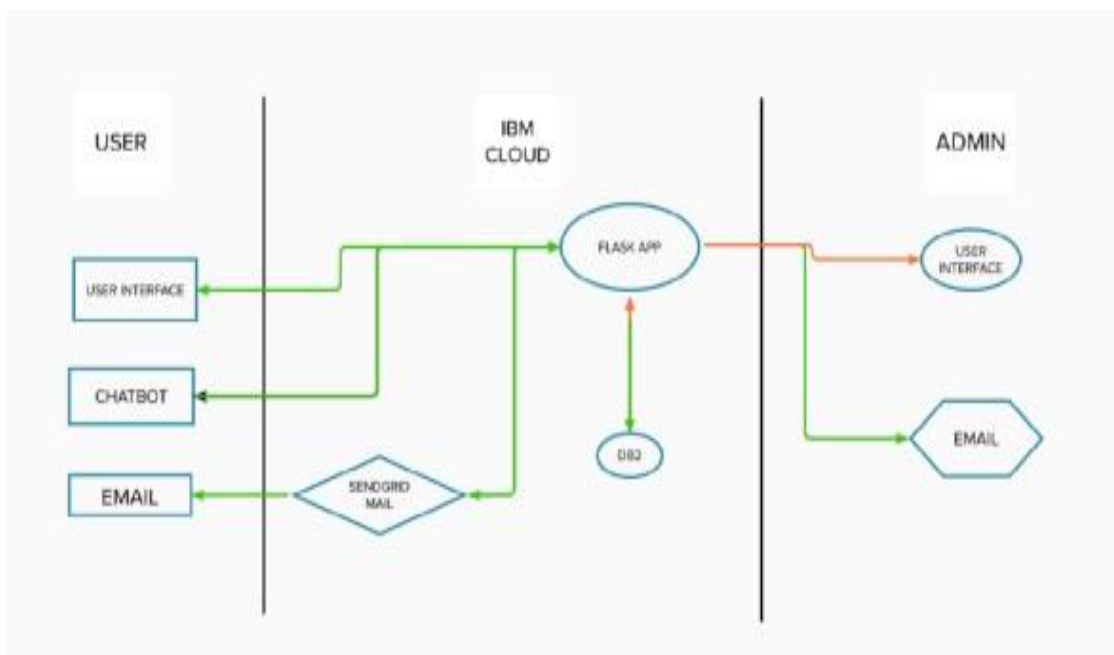
```

## 8 TESTING

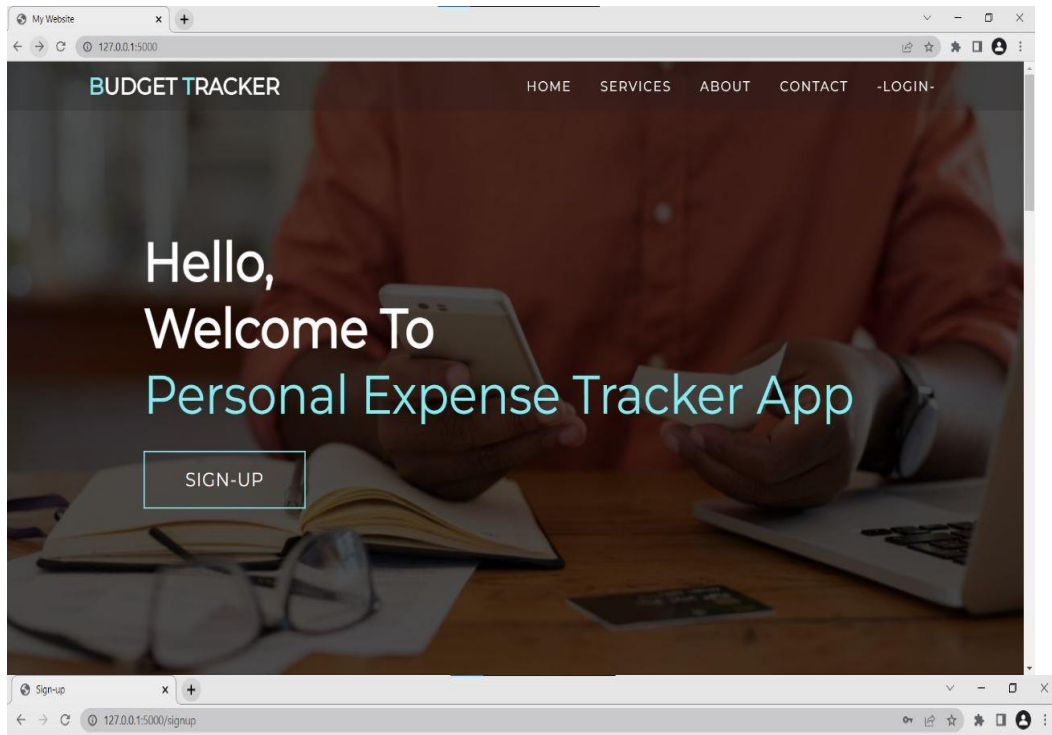
Black Box testing is the method that does not consider the internal structure, design, and product implementation to be tested. In other words, the tester does not know its internal functioning. The Black Box only evaluates the external behavior of the system. The inputs received by the system and the outputs or responses it produces are tested.

The White Box test method is the one that looks at the code and structure of the product to be tested and uses that knowledge to perform the tests. This method is used in the Unit testing phase, although it can also occur in other stages such as Integration tests. For the execution of this method, the tester or the person who will use this method must have extensive knowledge of the technology used to develop the program.

### 8.1 Test Cases



## 9 RESULT



### SIGN UP

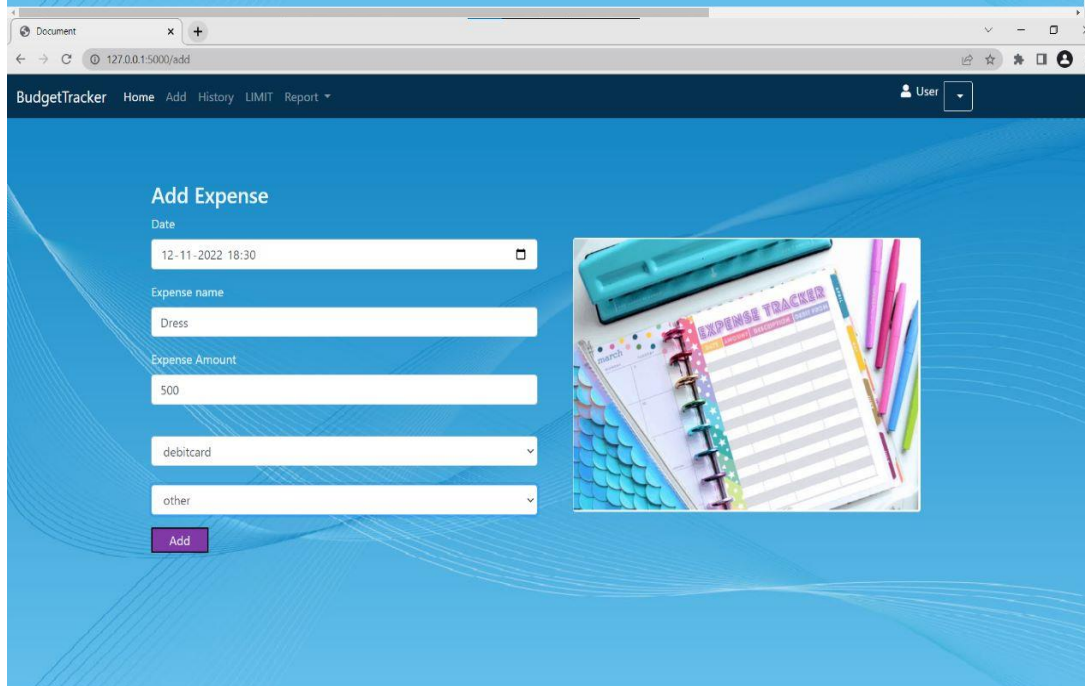
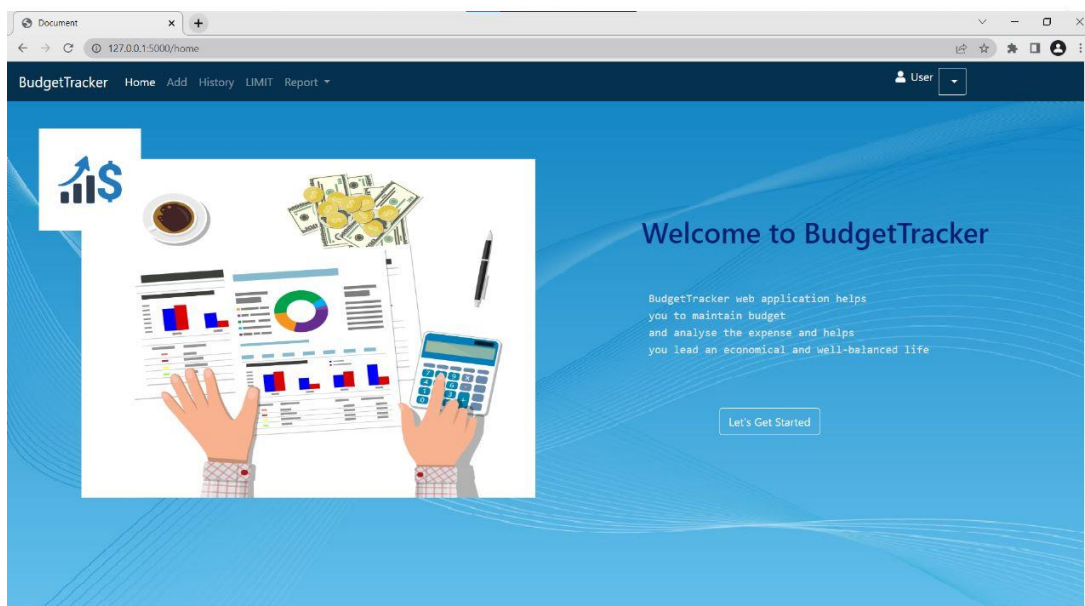
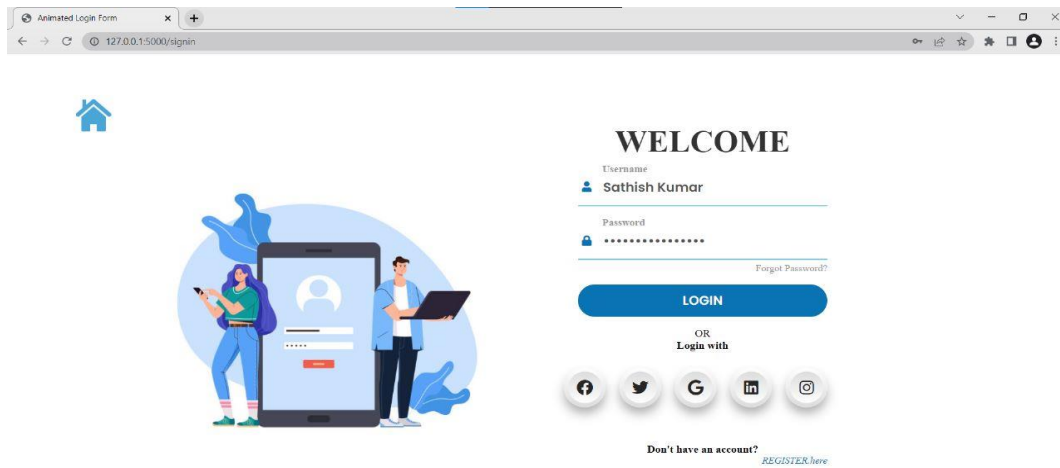
OR  
Sign-up with



☒ I read and agree to Terms & Conditions

CREATE ACCOUNT

Already have an account? [Sign in](#)



Document
127.0.0.1:5000/limitn
BudgetTracker
Home
Add
History
LIMIT
Report
User

Currently your MONTHLY limit is ₹ /-

ENTER the MONTHLY LIMIT to avoid over EXPENSES

BudgetTracker
Home
Add
History
LIMIT
Report
User

### EXPENSES

2022-11-14 22:40:00	food	₹ 5000	debitcard	entertainment	Edit	Delete
2022-11-13 23:30:00	dress	₹ 4000	debitcard	entertainment	Edit	Delete

BudgetTracker
Home
Add
History
LIMIT
Report
User

### Today Expense Breakdown

TIME	AMOUNT
22:40:00	5000

### Expense Breakdown BY Category

Food	0
Entertainment	5000
Business	0
Rent	0
EMI	0
Other	0
Total	₹ 5000

EXPENSE BREAKDOWN BY CATEGORY

Legend: Food, Entertainment, Business, Rent, EMI, Other

## 9.1 Performance Metrics

The performance of a recommendation algorithm is evaluated by using some specific metrics that indicate the accuracy of the system. The type of metric used depends on the type of filtering technique. Root Mean Square Error (RMSE), Receiver Operating Characteristics (ROC), Area Under Curve (AUC), Precision, Recall and F1 score is generally used to evaluate the performance or accuracy of the recommendation algorithms.

**Root-mean square error (RMSE).** RMSE is widely used in evaluating and comparing the performance of a recommendation system model compared to other models. A lower RMSE value indicates higher performance by the recommendation model. RMSE, as mentioned by [69], can be as represented as follows: \_\_\_\_\_

$$RMSE = \sqrt{\frac{1}{N_p} \sum_{u,i} (p_{ui} - r_{ui})^2} \quad (1)$$

where,  $N_p$  is the total number of predictions,  $p_{ui}$  is the predicted rating that a user  $u$  will select an item  $i$  and  $r_{ui}$  is the real rating.

**Precision.** Precision can be defined as the fraction of correct recommendations or predictions (known as True Positive) to the total number of recommendations provided, which can be as represented as follows:

$$\text{Precision} = \frac{\text{True Positive (TP)}}{\text{True Positive (TP)} + \text{False Positive (FP)}} \quad (2)$$

It is also defined as the ratio of the number of relevant recommended items to the number of recommended items expressed as percentages.

**Recall.** Recall can be defined as the fraction of correct recommendations or predictions (known as True Positive) to the total number of correct relevant recommendations provided, which can be as represented as follows:

$$\text{Recall} = \frac{\text{True Positive (TP)}}{\text{True Positive (TP)} + \text{False Negative (FN)}} \quad (3)$$

It is also defined as the ratio of the number of relevant recommended items to the total number of relevant items expressed as percentages.

**F1 Score.** F1 score is an indicator of the accuracy of the model and



ranges from 0 to 1, where a value close to 1 represents higher recommendation or prediction accuracy. It represents precision and recall as a single metric and can be as represented as follows:

$$F1 \text{ score} = 2 \times \text{Precision} \times \text{Recall} / (\text{Precision} + \text{Recall}) \quad (4)$$

**Coverage.** Coverage is used to measure the percentage of items which are recommended by the algorithm among all of the items.

**Accuracy.** Accuracy can be defined as the ratio of the number of total correct recommendations to the total recommendations provided, which can be as represented as follows:

$$\text{Accuracy} = \text{TP} + \text{FN} / (\text{TP} + \text{FN} + \text{TN} + \text{FP}) \quad (5)$$

**Intersection over union (IoU).** It represents the accuracy of an object detector used on a specific dataset [70].

$$\text{IoU} = \text{TP} / (\text{TP} + \text{FN} + \text{FP}) \quad (6)$$

**ROC.** ROC curve is used to conduct a comprehensive assessment of the algorithm's performance [57].

**AUC.** AUC measures the performance of recommendation and its baselines as well as the quality of the ranking based on pairwise comparisons [5].

**Rank aware top-N metrics.** The rank aware top-N recommendation metric finds some of the interesting and unknown items that are presumed to be most attractive to a user [71]. Mean reciprocal rank (MRR), mean average precision (MAP) and normalized discounted cumulative gain (NDCG) are three most popular rank aware metrics.

**MRR.** MRR is calculated as a mean of the reciprocal of the position or rank of first relevant recommendation [72,73]. MRR as mentioned by [72,73] can be expressed as follows:

$$\text{MRR} = 1 / N_u \sum_{u \in N_u} 1 / L_{nu}[k] \in R_u \quad (7)$$

where  $u$ ,  $N_u$  and  $R_u$  indicate specific user, total number of users and the set of items rated by the user, respectively.  $L$  indicates list of ranking length ( $n$ ) for user ( $u$ ) and  $k$  represents the position of the item found in the he lists  $L$ .

**MAP:** MAP is calculated by determining the mean of average precision at the points where relevant products or items are found. MAP as mentioned by [73] can be expressed as follows.

$$\text{MAP} = 1 / N_u |R_u| \sum_{n=1}^k (L_{nu}[k] \in R_u) P_u @ k \quad (8)$$



where  $P_u$  represents precision in selecting relevant item for the user.

NDCG: NDCG is calculated by determining the graded relevance and positional information of the recommended items, which can be expressed as follows [73].

$$NDCG_u = \sum_{n,k=1} G(u,n,k)D(k) / \sum_{n,k=1} G^*(u,n,k)D(k) \quad (9)$$

where  $D(k)$  is a discounting function,  $G(u, n, k)$  is the gain obtained recommending an item found at  $k$ -th position from the list  $L$  and  $G^*(u, n, k)$  is the gain related to  $k$ -th item in the ideal ranking of  $n$  size for  $u$  user.

## 10 ADVANTAGES & EXISTING SYSTEM

### Advantages

1. The best organizations have a way of tracking and handling these reimbursements. This ideal practice guarantees that the expenses tracked are accurately and in a timely manner. From a company perspective, timely settlements of these expenses when tracked well will certainly boost employees' morale
2. Financially Aware and Improve Money Management tracking your expenditures ensures you achieve your project financial targets. How is that? By clearly understanding your project spending using project budget limits, you can aptly make the necessary changes to complete your project within time and budget.
3. Effective expense tracking and reporting to avoid conflict. As a project manager or business owner, you can set clear policies for the expense types and reimbursement limits to avoid misunderstandings are about costs. Tracking the project expenses by asking team members to provide receipts is helpful to avoid conflict and maintain compliance also. An excellent reporting mechanism is extremely helpful to support the amount to be reimbursed to your team and also invoicing to your customer.
4. Helps anticipate the costs of similar projects When you formally track and report expenses, you have a permanent documentation which helps you correctly anticipate expenses for similar projects in the future. This is even more significant when it comes to budget-making process.

5. Tracking the amount of money spent on the projects is important to invoice customers and determine the cost & profitability analysis when your company is providing services to another company. On the other hand, expense tracking or internal project is important for cost and ROI calculation. Understanding how this money is being utilized across the project is such a significant issue. The consequence for not properly tracking and reporting project expenses may lead to a budgetary issues.

## **EXISTING SYSTEM**

### **A) How it Actually Works**

In existing, we need to maintain the Excel sheets, CSV etc. files for the user daily and monthly expenses. In existing, there is no as such complete solution to keep a track of its daily expenditure easily. To do so a person has to keep a log in a diary or in a computer, also all the calculations need to be done by the user which may sometimes result in errors leading to losses.

### **B) Drawbacks of the Existing System**

There can be many disadvantages of using a manual accounting system. Accounting, for any business, can be a complex undertaking. A manual accounting system requires you to understand the accounting process in a way that may be unnecessary with a computerized accounting system. This can be an advantage or a disadvantage, depending on the person doing the bookkeeping; often, a specially trained professional is needed to ensure that accounting is done properly. Unraveling the complexity of your financial records by hand may be time consuming. Since it takes time to generate reports.

## **11 CONCLUSION & FUTURE SCOPE**

### **11.1 CONCLUSION**

In conclusion, developing a personal budget and tracking all expenses and spending is a crucial aspect of personal finances. Set aside a fixed amount in a savings account, they say you should always have three months worth of your living expenses in a savings account in case of emergencies.

### **11.2 FUTURE SCOPE**

The main objective of this project is support to the user to sustain all financial activities like digital automated dairy. This application helps the user to avoid unexpected expenses and bad financial situations.

- Using this application, users can manage all financial data and track all expense and income category wise.
- Creating a category and recording all expenses and income under the category.
- Enable the notification system user get notification daily at a specific time that can help the user insert expense and income.
- Backup and Restore all information.
- Report are generated in PDF format in category wise or time period

## **12 APPENDIX**

FRS can be defined as a means of feature matching between fashion products and users or consumers under specific matching criteria. Different research addressed apparel attributes such as the formulation of colors, clothing shapes, outfit or styles, patterns or prints and fabric structures or textures. Guan et al. studied these features using image recognition, product attribute extraction and feature encoding. Researchers have also considered user features such as facial features, body shapes, personal choice or preference, locations and wearing occasions in predicting users' fashion interests. A well-defined user profile can differentiate a more personalized or customized recommendation system from a conventional system. Various research projects on apparel recommendation systems with personalized styling

guideline and intelligent recommendation engines have been conducted based on similarity recommendation and expert advisor recommendation systems.

### **GitHub & Project Demo Link**

<https://github.com/IBM-EPBL/IBM-Project-19135-1659693712.git>

[https://drive.google.com/file/d/1uycqvRA\\_wG-v-uR1r0YrwmQ63CvIqUTt/view?usp=sharing](https://drive.google.com/file/d/1uycqvRA_wG-v-uR1r0YrwmQ63CvIqUTt/view?usp=sharing)