

# PERSONAL EXPENSE TRACKER APPLICATION

## A PROJECT REPORT

*Submitted by*

**TEAM ID: PNT2022TMID47204**

<b>BHUVANESH S</b>	-	<b>822719104010</b>
<b>AVINASH R</b>	-	<b>822719104007</b>
<b>BALAMURUGAN V</b>	-	<b>822719104008</b>
<b>HARIPRASATH S</b>	-	<b>822719104024</b>

Date	23 November 2022
Team ID	PNT2022TMID47204
Project Name	Personal Expense Tracker Application

# TABLE OF CONTENTS

## **1. INTRODUCTION**

1. Project Overview
2. Purpose

## **2. LITERATURE SURVEY**

1. Existing problem
2. References
3. Problem Statement Definition

## **3. IDEATION & PROPOSED SOLUTION**

1. Empathy Map Canvas
2. Ideation & Brainstorming
3. Proposed Solution
4. Problem Solution fit

## **4. REQUIREMENT ANALYSIS**

1. Functional requirement
2. Non-Functional requirements

## **5. PROJECT DESIGN**

1. Data Flow Diagrams
2. Solution & Technical Architecture
3. User Stories

## **6. PROJECT PLANNING & SCHEDULING**

1. Sprint Planning & Estimation
2. Sprint Delivery Schedule
3. Reports from JIRA

## **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 & Project Demo Link

## **1. INTRODUCTION:**

### **1.1. Project Overview:**

It is important to keep track of your spending throughout the month since it helps you create a better monthly budget by giving your insight into how your money is being used. Consequently, the Personal Expense Tracker program will be utilized to keep track of all spending and may be used to manage all expenses that are organized by category. With the use of this program, we may group the invoices by the things they belong to, such as food, retail, stock, etc. The project is about personal expenses tracker application. Using this application, we can track our day-to-day expenses based on weekly, monthly and yearly basis. This tracker application allows us to maintain the complete record of our expenses which we spend and also represent the expenses in the form of charts and tables. It also helps with the expenses if the user exceeds the limit that has been set an email will be sent to user email.

### **1.2. Purpose:**

The expenses which we are spending on weekly, monthly and yearly basis allows us to know how much we are spending and how much we should reduce our expenses. This will help us to save money in the form of savings. This tracker application also shows the balance amount left after spending the entire expenses which will be useful for us to be careful for the next month before spending the money.

## **2. LITERATURE SURVEY:**

### **2.1. Existing problem:**

The expense tracker application is a web application. This application will not be provided for free. To get all the benefits of the application we need to pay some amount of money. In free application, the tracker only tracks daily expenses. But, in paid application, the tracker tracks monthly and yearly expenses also.

### **2.2. References:**

The following links are used for the references for building this personal tracker expenses web application:

1. Velmurugan A, Albert Mayan J, Niranjana P and Richard Francis "Expense Manager Application" Android studio, Kotlin, java, SQLite, AndroidOS, Figma designing tool (2020).
2. Nidhi Jitendra Jadhav, Rutuja Vijay Chakor, Trupti Mahesh Gunjal, Damayanti. D. Pawar "Expense Tracker" Mobile application, Using Database layer which holds all of the data and financial information, supported by User Interface (2022).
3. Tamia Ruvimbo Masendu, Aanajey Mani Tripath "Daily Expense Tracker" This application is a GUI (Graphics User Interface) based application. Technology used Java (Apache NetBeans IDE 13) and my MySQL Workbench (2022).

4. Velmurugan. R, Mrs. P. Usha “Expense Tracker Application” Java, Xml, MySQL (2021).
5. Dr. V. Geetha, G. Nikhitha, H. Sri Lasya, Dr. C.K. Gomathy “Expenditure Management System” JavaScript, JSX, React and Mango db (2022).

### 2.3. Problem Statement Definition:

It is important to keep track of your spending throughout the month since it helps you create a better monthly budget by giving your insight into how your money is being used. Consequently, the Personal Expense Tracker program will be utilized to keep track of all spending and may be used to manage all expenses that are organized by category. With the use of this program, we may group the invoices by the things they belong to, such as food, retail, stock, etc.

#### Customer Problem Statement:

I am	I'm trying to	But	Because	Which makes me feel
A software engineer	trying to manage his daily expenses	but	there is no time for him/her to calculate all the money spent	which feels him/her fear as they would use more money
A housewife	wants to calculate budget for household supplies monthly	but	it is difficult to track all the bills	helpless
A student	trying to track daily expenses	but	most of the students have less money	helpless
A middle class worker	trying to reduce daily expenses	but	cannot track as there is no easy way to categorise all expenses	frustrated

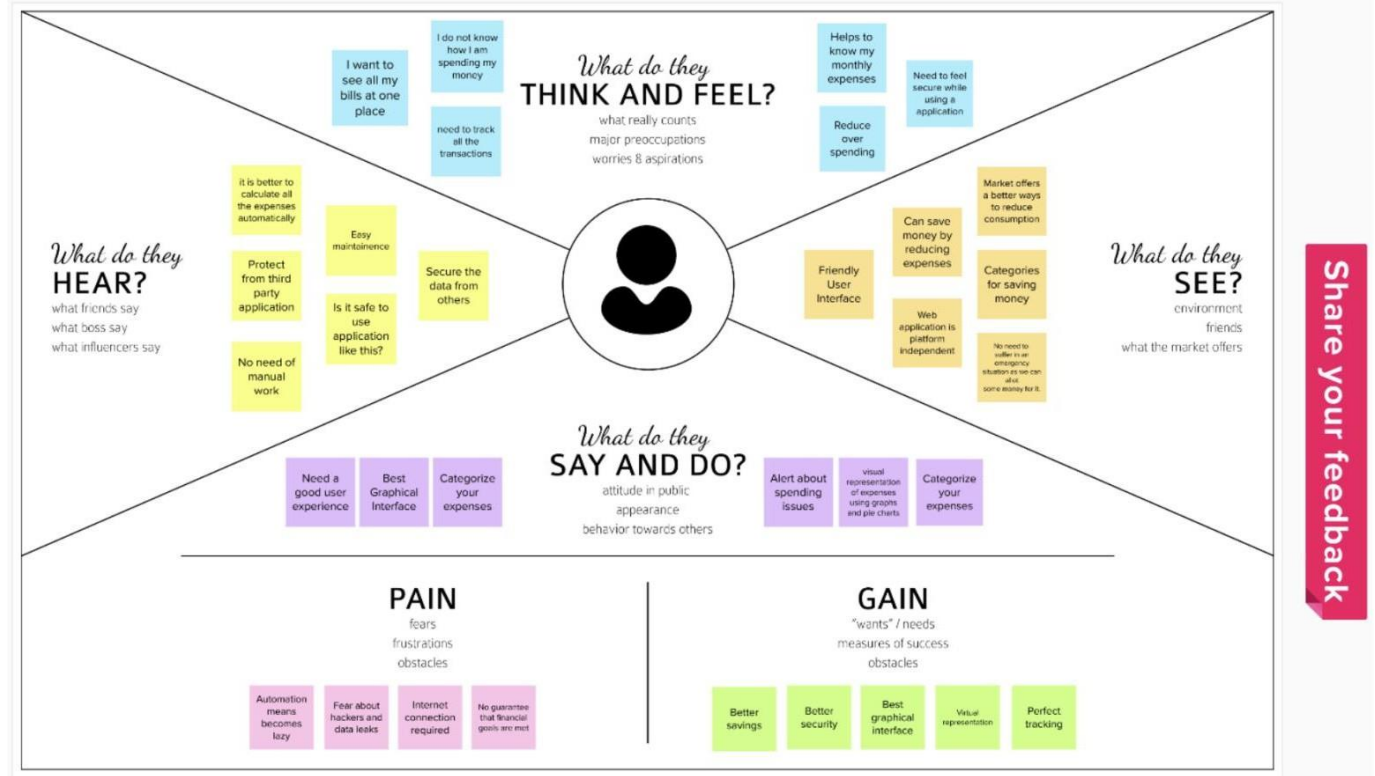
## 3. IDEATION & PROPOSED SOLUTION

### 3.1. Empathy Map Canvas

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviors and attitudes. It is a useful tool to help teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is

experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.

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




### 3.2. Ideation & Brainstorming

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions. Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

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

Template



## Brainstorm & idea prioritization

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

🕒 10 minutes to prepare

🕒 1 hour to collaborate

👥 2-8 people recommended

➔

**Before you collaborate**

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

🕒 10 minutes

A

**Team gathering**

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

B

**Set the goal**

Think about the problem you'll be focusing on solving in the brainstorming session.

C

**Learn how to use the facilitation tools**

Use the Facilitation Superpowers to run a happy and productive session.

Open article ➔

1


**Define your problem statement**

The problem statement is to create a application which helps the user to track all the expenses monthly and categorize the bills according to its features which will help the user to plan the budget for the future.

🕒 5 minutes

PROBLEM

How might we [your problem statement]?



### Key rules of brainstorming

To run an smooth and productive session

😊 Stay in topic.

💡 Encourage wild ideas.

🕒 Defer judgment.

👂 Listen to others.

🗣️ Go for volume.

👁️ If possible, be visual.

📝 Share template feedback

# Step-2: Brainstorm, Idea Listing and Grouping

2

## Brainstorm

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

10 minutes

**TIP**  
You can select a sticky note and hit the pencil (switch to sketch) icon to start drawing

### Prem Kashyap

- Show the monthly expenses
- Categorize the bills
- Set up some built-in budget plans
- automatically feeds the data from SMS or any app
- Control all your expenses
- Suggestions for budget planning
- Visualize the expenses monthly
- Connect LIFI apps which will have history of expenses
- Provide security to the application

### Sai Dayakar

- Prepare excel sheet of the monthly budget
- Automate the process
- Managing the expenses
- Checking account statements
- Tracking expenses regularly
- Finding suitable tool for tracking the expenses
- Keep a thorough paper trail
- Subtract your expenses from your income
- Stick to your spending limits

### Dinesh Kumar

- Keep track of purchases, savings
- prioritize spending
- make alert when over budget
- setting goals
- make reminders to pay bills
- allocate certain amounts to each transaction
- categorize expenses
- to merge expenses via LIFI connect how much you owe
- adjust budget based on balance

### Arokiya Nivin

- Helps to solve problems
- Categorize the expenses
- connects all LIFI id's
- Keep monthly expenses
- offline mode
- Joint accounts with family
- Monthly report for expenses
- Save money for future plans
- Helps you to stick on your budget

3

## Group Ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you can break it up into smaller sub-groups.

20 minutes

### Categorizing the expenses

- Categorize the expenses
- Prepare excel sheet of the monthly budget
- allocate certain amounts to each category
- Categorize the bills
- Managing the expenses
- categorize expenses
- Visualize the expenses monthly
- Subtract your expenses from your income
- adjust budget based on balance

### Other Important Features

- Set up some built-in budget plans
- Automate the process
- offline mode
- Suggestions for budget planning
- Finding suitable tool for tracking the expenses
- Joint accounts with family
- setting goals
- to merge expenses via LIFI connect how much you owe
- make alert when over budget

### Tracking the Expenses

- make alert when over budget
- keep track of purchases, savings
- automatically feeds the data from SMS or any app
- Show the monthly expenses
- Keep a thorough paper trail
- connects all LIFI id's
- Connect LIFI apps which will have history of expenses
- Tracking expenses regularly
- Monthly report for expenses
- Keep monthly expenses
- Subtract your expenses from your income
- allocate certain amounts to each category

### Step-3: Idea Prioritization





### 3.3. Proposed Solution

#### Solution description

At the instant, there is no as such complete solution present easily or we should say free of cost which enables a person 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 needs to be done by the user which may sometimes results in errors leading to losses. Due to lack of a complete tracking system, there is a constant overload to rely on the daily entry of the expenditure and total estimation till the end of the month.

#### Uniqueness

To manage the expenses in an easy manner, we are going to develop an web application, which will help to easily track the expenses and also helps to give helpful insights about financial management.

#### Customer Satisfaction

The user will get mail and SMS notification, when their expenses limit exceeds and it will also set a remainder, if a user forgets to make entry about his/her expenses. We will also introduce automated suggestions, which will help to plan the budget accordingly and we will categorize expenses based on entertainment, shopping etc.

#### Revenue Model

We can provide the application in a subscription based.

#### Scalability of the Solution

This application can handle large number of users simultaneously

### 3.4. Problem Solution fit

Define CS, fit into CC	<b>1. CUSTOMER SEGMENT(S)</b> <b>CS</b>  Most of the clients are working adults who are above 21 and college students who wants to save money	<b>6. CUSTOMER CONSTRAINTS</b> <b>CC</b>  1. An expense tracker is a software program or an application that helps you to keep accurate record of your income and expenses.  2. It is also commonly referred to us an expense manager. Many people in India have fixed incomes and acknowledge that they spend money towards the end of each month.  3. Internet connection is required to use this application.	<b>5. AVAILABLE SOLUTIONS</b> <b>AS</b>  For user convenience, this project is being developed on web application. Because they include an web application anytime they can create immediate expenses. This makes using this data contrary. We think that a practical design and a practical web application can solve this problems. Such an application is capable of keeping track of expenditure, providing a comprehensive view with user-friendly interface, and being enough intelligence to display the historyof expenditures indicating the application	Explore AS, differentiate							
	<b>2. JOBS-TO-BE-DONE / PROBLEMS</b> <b>J&amp;P</b>  1. Due to manual error in the expenses calculation , it is best to automate the function to track expenses.  2. There is a possibility that the bills may get lost as we customer will be calculating for a long term goal.  3. Alerting the client when the budget threshold crosses.  4. To Track all the transactions done by customer on daily bases.  5. Categorizing all the expenses based on what the customer is spending on like food , entertainment, etc.	<b>9. PROBLEM ROOT CAUSE</b> <b>RC</b>  1. You may rapidly pay for the invoices by using an expense tracker app that supports financial transaction using debit cards and credit cards and net banking.  2. Additionally, a spending tracking software will spend payment reminders and link payment to client accounts.	<b>7. BEHAVIOUR</b> <b>BE</b>  1. Customers get unlimited access to their calculation. This approach makes it very simple and really beneficial to estimate their expenditure and needs.  2. There will be a chat bot which will guide the customer and rectify their doubts		Focus on J&P, fit into BE, understand RC						
Identify strong TR & EM	<b>3. TRIGGERS</b> <b>TR</b>  1. Excessive spending of money. 2. No money in case of emergency. 3. By seeing its features in ads.	<b>10. YOUR SOLUTION</b> <b>SL</b>  1. This user benefit this application is developed as web application so customers can use this application anywhere like mobile, etc.  2. This makes using this data contrary. There is still complication in areas like there is no assurance for data compatible, there are chances of crucial inputs can be missed and the manual errors may seek in.  3. Such an application is capable of keeping track of expenditure, providing a comprehensive view with user-friendly interface, and being enough intelligence to display the history of expenditures indicating the application.	<b>8. CHANNELS of BEHAVIOUR</b> <b>CH</b>  Online: What kind of action do customer take online? Yes, mint's parent company, intuit, uses cutting-edge security and technology to protect the personal and financial data of its users. Multi-factor authentication as well as software and hardware encryption are security measures.  Offline: What kind of action do customer take offline? The most convenient and cost-free personal finance to this expense tracker Data can be exported as a CSV file and it can be used offline.	Map & SL, Grow the fituppp							
	<b>4. EMOTIONS: BEFORE / AFTER</b> <b>EM</b>  <table border="0"> <tr> <td><b>BEFORE</b></td> <td><b>AFTER</b></td> </tr> <tr> <td>• Anxious</td> <td>• Confident</td> </tr> <tr> <td>• Confused</td> <td>• Composed</td> </tr> <tr> <td>• Fear</td> <td>• Calm</td> </tr> </table>	<b>BEFORE</b>	<b>AFTER</b>		• Anxious	• Confident	• Confused	• Composed	• Fear	• Calm	
<b>BEFORE</b>	<b>AFTER</b>										
• Anxious	• Confident										
• Confused	• Composed										
• Fear	• Calm										

## 4. REQUIREMENT ANALYSIS

### 4.1. Functional requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Phone number. Registration through Gmail. Registration through Username and Gmail.
FR-2	User Confirmation	Confirmation via Email. Confirmation via OTP.
FR-3	User Login	Login using Gmail. Login through Username.
FR-4	Manage Expenses	Create or update new budget/expense limit. Manage expenses by categorizing the priority ones.
FR-5	Expense Tracker	Analyze the level of expenses in graphical report format and graphical representation of expenses based on daily, monthly, yearly usage and categorize the based on what customer is using for.
FR-6	Manage income and expenditure	Create or update income and expenditure details, then the app suggests better ideas for budgeting. Provides built-in plans for some certain budget goals.

### 4.2. Non-functional Requirements:

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

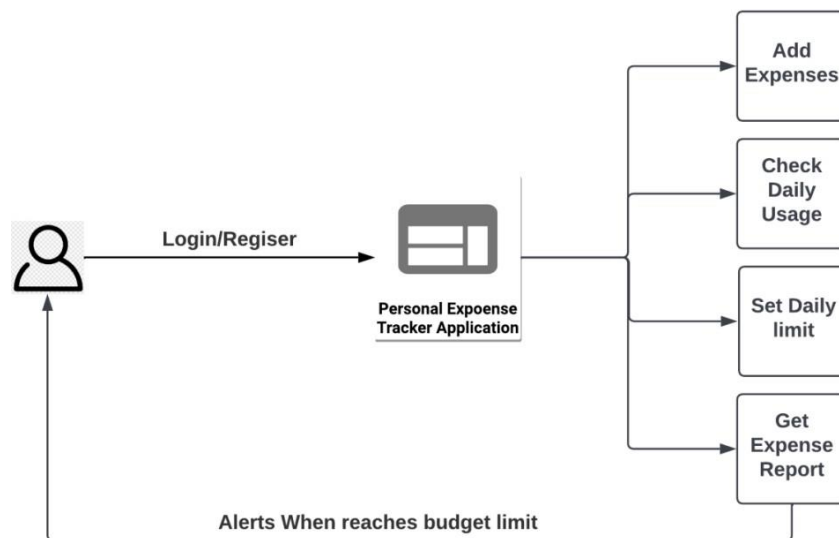
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	This system will be used by anyone who needs to manage their expenses and to make better budgeting ideas.
NFR-2	Security	This system prevents customer's data securely and protects from malware attacks or unauthorized access.
NFR-3	Reliability	This system is highly reliable and it reduces the manual work load.
NFR-4	Performance	It tracks the expenses and generates reports quickly. It engages users efficiently with better budgeting ideas.
NFR-5	Availability	User can make his/her reports offline and this report is operational at any time.
NFR-6	Scalability	This system has better storage capacity and it manages large no of user's data.

## 5. PROJECT DESIGN

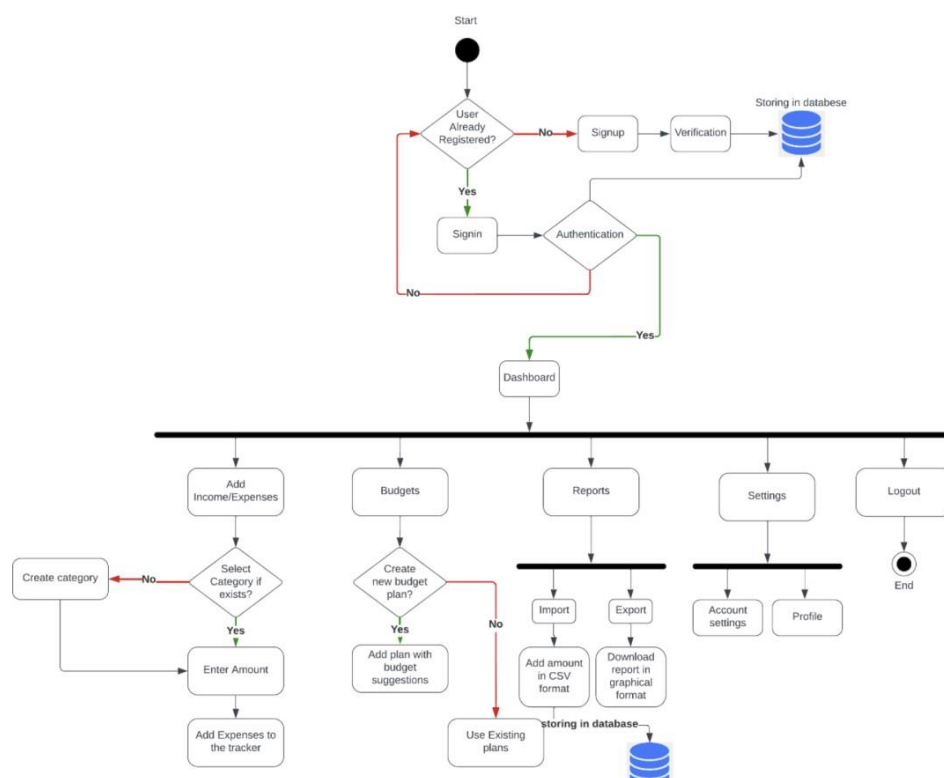
### 5.1. Data Flow Diagrams

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

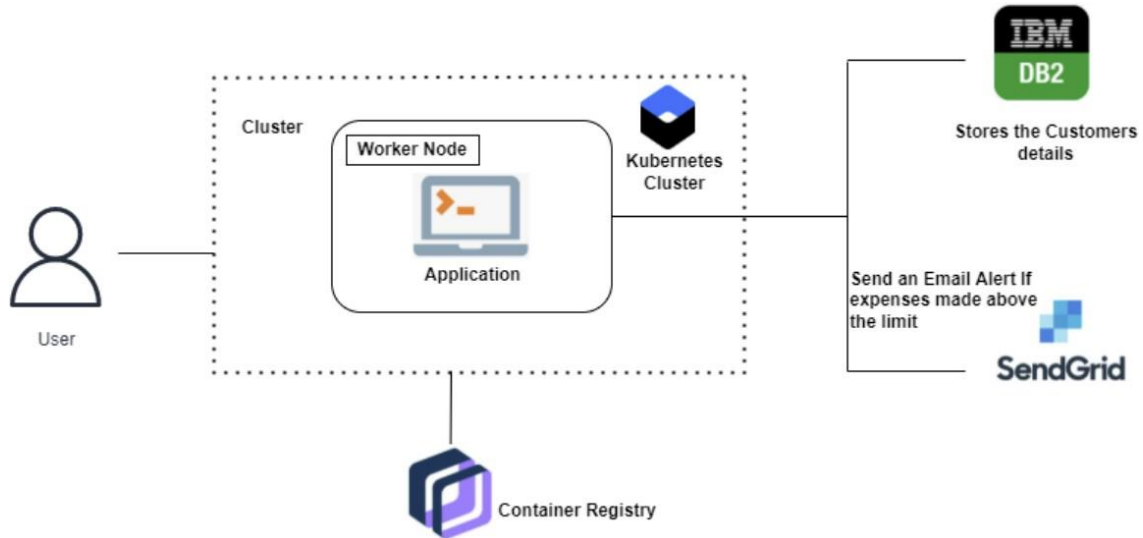
Simplified:



Industry Level:



## 5.2. Solution & Technical Architecture



## 5.3. User Stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
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	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Gmail / Phone Number	I can confirmation using OTP	Medium	Sprint-2
Customer (Web user/ Mobile User)	Login	USN-4	As a user, I can log into the application by entering email / username & password	I can access my account / dashboard	High	Sprint-1
Customer (Web user/ Mobile User)	Dashboard	USN-5	As a user, I can register for the application by Bank account number or using UPI id or manually upload expense or upload csv file	I can access my account / dashboard	Medium	Sprint-2
		USN-6	As a user, I will receive confirmation mail and OTP, once I have registered for the application	I can confirmation using OTP	High	Sprint-3
Customer (Mobile user)		USN-7	As a user, I can use all the services provided by the application in both online and offline		Medium	Sprint-2
Customer (Web User)		USN-8	As a user, I can use all the services provided by the application only in offline and platform independent		Medium	Sprint-3
Customer Care Executive	Customer support	USN-9	As a user, I can support for customers to automatic logging of recurring transactions.		High	Sprint-3
Administrator	Responsibility	USN-10	As a system administrator, provide security to the data and provide support to the user		High	Sprint-3

## 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 entering my email, password, and confirming my password.	3	High	Chilakamarthi Prem Kashyap
		USN-2	As a user, I can register for the application through the Gmail.	3	High	Gollapudi Sai Dayakar Reddy
	Confirmation	USN-3	As a user, I will receive confirmation email once I have registered for the application.	4	High	Arokia Nivin A
	Login	USN-4	As a user, I can log into the application by entering email & password.	3	High	Dinesh Kumar N
	Connecting to IBM DB2	USN-3	Linking database to store and verify login credentials of users.	3	High	Chilakamarthi Prem Kashyap
	Dashboard	USN-5	Logging in takes to the dashboard. Dashboard is used to access all the services provided to the user.	4	High	Chilakamarthi Prem Kashyap
Sprint 2	Workspace	USN-1	Workspace for personal expense tracking.	6	High	Chilakamarthi Prem Kashyap
	Charts	USN-2	Creating various graphs and statistics of customer's data.	4	Medium	Gollapudi Sai Dayakar Reddy
	Connecting to IBM DB2	USN-3	Linking database with dashboard.	5	High	Dinesh Kumar N
		USN-4	Making dashboard interactive with JS.	5	High	Arokia Nivin A

Sprint-3		USN-1	Wrapping up the server side works of frontend.	5	Medium	Chilakamarthi Prem Kashyap
	Watson Assistant	USN-2	Creating Chatbot for expense tracking and for clarifying user's query.	4	Medium	Gollapudi Sai Dayakar Reddy
	SendGrid	USN-3	Using SendGrid to send mail to the user about their expenses.	5	High	Dinesh Kumar N
		USN-4	Integrating both frontend and backend.	6	High	Arokia Nivin A
Sprint-4	Docker	USN-1	Creating image of website using docker.	5	High	Chilakamarthi Prem Kashyap
	Cloud Registry	USN-2	Uploading docker image to IBM Cloud registry.	5	High	Arokia Nivin A
	Kubernetes	USN-3	Create container using the docker image and hosting the site.	5	High	Dinesh Kumar N
	Exposing	USN-4	Exposing IP/Ports for the site.	5	High	Gollapudi Sai Dayakar Reddy

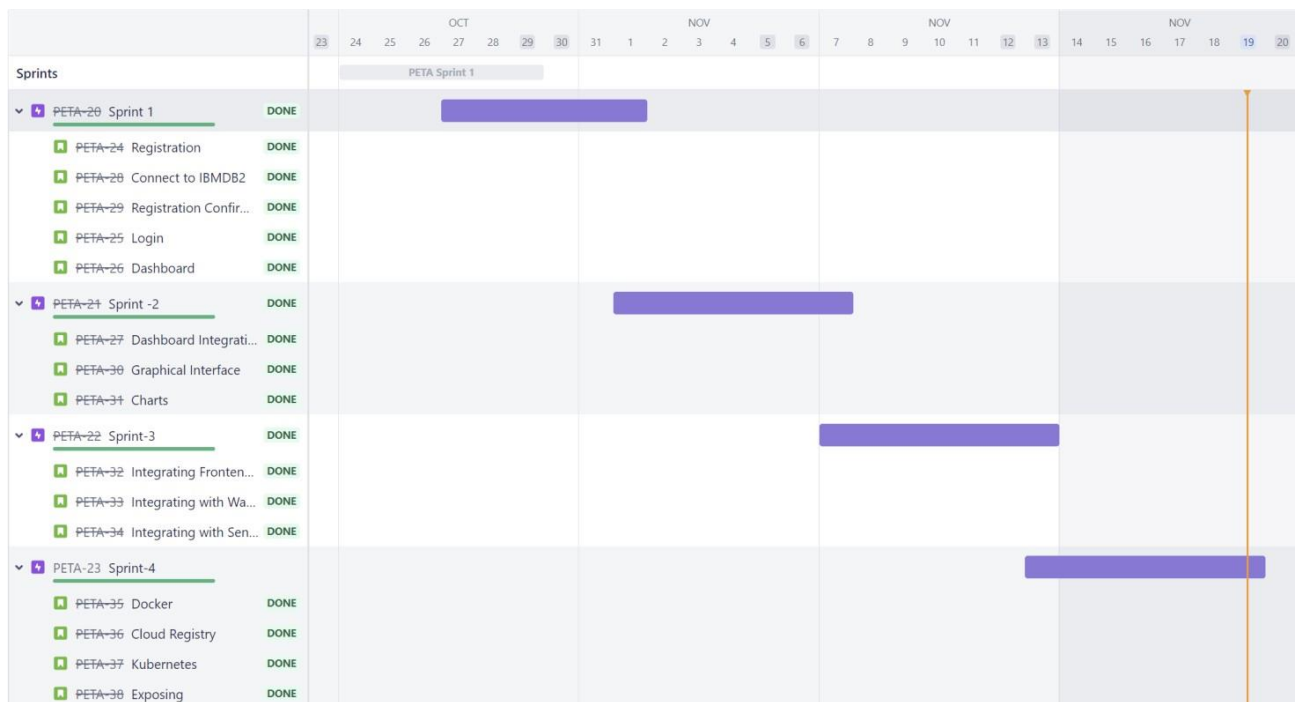
### 6.2. Sprint Delivery Schedule

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	26 Oct 2022	31 Nov 2022	20	12 Nov 2022
Sprint-2	20	6 Days	01 Nov 2022	07 Nov 2022	20	17 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	18 Nov 2022
Sprint-4	20	6 Days	13 Nov 2022	19 Nov 2022	20	19 Nov 2022

### Velocity:

Sprint	Sprint Duration	Average Velocity	Actual Velocity
Sprint-1	16	3.33	1.25
Sprint-2	16	3.33	1.26
Sprint-3	10	3.33	2
Sprint-4	6	3.33	3.33

## 6.3 Reports from JIRA



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

### 7.1. Importing required packages and connecting to IBM db2

```
from flask import (Flask, flash, redirect, render_template, request, session,url_for)
from flask_session import Session
from datetime import date

app = Flask(__name__)
app.debug=True
app.config["SESSION_PERMANENT"] = False
app.config["SESSION_TYPE"] = "filesystem"
app.config["SECRET_KEY"] = "Personal_Expense_Tracker_Application"
official_mail="expensetracker0046@gmail.com"

Session(app)

conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases
```



## 7.2. Home page routing

```
##### HOME #####

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

Routing for the home.html file is given.

## 7.3. Signup page routing

```
##### SIGN UP #####

@app.route("/signup", methods=('GET', 'POST'))
def signup():
    msg=""
    if request.method == 'POST':
        name = request.form['fname']
        username = name.split(" ")[0]
        email = request.form['femail']
        mobile = request.form['mobile']
        password = request.form['password']

        sql = "SELECT * FROM customers WHERE email=?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt,1,email)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)

        if account:
            return render_template("already_registered.html")
        elif not re.match('[^@]+@[^@]+\.[^@]+', email):
            msg = 'Invalid email address !'
            return render_template('signup.html',msg=msg)
        elif not re.match(r'^[a-zA-Z0-9]+([_ ])?[a-zA-Z0-9]+$', name):
            msg = 'Name must contain only characters, numbers !'
            return render_template('signup.html',msg=msg)
        elif not re.match(r'^(\d{1,2}\s)?1?-?\s?(?(\d{3})?\s-)?\d{3}[\s-]?\d{4}$', mobile):
            msg = 'Enter valid mobile number!'
            return render_template('signup.html',msg=msg)
        else:
            insert_sql = "INSERT INTO customers VALUES (?,?,,?,?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prepare_stmt, 1, name)
            ibm_db.bind_param(prepare_stmt, 2, username)
            ibm_db.bind_param(prepare_stmt, 3, email)
            ibm_db.bind_param(prepare_stmt, 4, mobile)
            ibm_db.bind_param(prepare_stmt, 5, password)
            ibm_db.execute(prepare_stmt)

            message = Mail(
                from_email = official_mail,
                to_emails = email,
                subject='Registration Successful',
                html_content='<h1>Hello '+name+', <br>Welcome to Personal Expense Tracker Application (PETA) .<br>Hope we have a great journey ahead.</h1>')
            try:
                sg = SendGridAPIClient(os.environ.get('SENDGRID_API_KEY'))
                sg.send(message)
            except Exception as e:
                print(e)

            return render_template('registration_success.html')

    return render_template('signup.html')
```

In this page registration is done and details are validated using regex and uploaded in customers table in ibmdb2. If registration is success then registration success mail is sent to the user email address.

## 7.4. About page routing

```
##### ABOUT #####

@app.route("/about")
def about():
    return render_template('about.html')
```

In the above picture aboutpage.html routing is done.

## 7.5. Signin page routing

```
##### SIGN IN #####

@app.route("/signin", methods=('GET', 'POST'))
def signin():
    if request.method == 'POST':
        femail = request.form['femail']
        fpassword = request.form['password']
        session["email"]=femail
        session["password"]=fpassword

        sql = "SELECT * FROM customers where email=? and password=?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt,1,femail)
        ibm_db.bind_param(stmt,2,fpassword)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)

        if account:
            return redirect("/dashboard")

        else:
            return render_template('invalid_credentials.html')

    return render_template('signin.html')
```

In signin page user can login using email and password if valid page will redirect to dashboard.  
If invalid then invalid\_credentials.html is rendered.

## 7.6. Dashboard page routing

In Dashboard page current monthly and yearly expenses are shown in a pie chart

Expense limits and expense usage in monthly and yearly wise is shown in dashboard.

If expense usage per month or year exceeds expense limit set by the user then a mail is sent to the user email that expense usage is exceeded.



```
##### DASHBOARD #####

@app.route("/dashboard", methods=('GET', 'POST'))
def dashboard():
    m1=0
    y1=0
    limit=False
    msg=""
    months={}
    years={}
    session["date"] = date.today()
    session["date"] = str(session["date"])
    if request.method=="GET":
        query2="select * from expenses"
        exec_query=ibm_db.exec_immediate(conn,query2)
        row=ibm_db.fetch_both(exec_query)
        user=False
        while(row):
            if row[0]==session["email"]:
                if row[3][:7]==session["date"][:7]:
                    if row[3] not in months:
                        months[row[3]]=row[1]
                    else:
                        months[row[3]]+=row[1]
                if row[3][:4]==session["date"][:4]:
                    if row[3][:7] not in years:
                        years[row[3][:7]]=row[1]
                    else:
                        years[row[3][:7]]+=row[1]
            row=ibm_db.fetch_both(exec_query)

        query1="select * from limits"
        exec_query=ibm_db.exec_immediate(conn,query1)
        row=ibm_db.fetch_both(exec_query)
        k=False
        while(row):
            if row[0]==session["email"]:
                m1=row[1]
                y1=row[2]
            row=ibm_db.fetch_both(exec_query)

        session["current_month"]=[i for i in months.keys()]
        session["month_amount"]=[i for i in months.values()]
        session["current_year"]=[i for i in years.keys()]
        session["year_amount"]=[i for i in years.values()]

        if m1==0 and y1==0:
            msg="No expense limits are set, please set them to use the application"
        if sum(session["month_amount"])>m1 or sum(session["year_amount"])>y1:
            limit=True

        if limit==True:
            msg="Expense limit exceed Please manage your expenses."
            message = Mail(
                from_email = official_mail,
                to_emails = session["email"],
                subject='Expense limit reached',
                html_content='<h1> Monthly expense limit reached.<br>Please use money carefully or increase budget limit.</h1>')
            try:
                sg = SendGridAPIClient(os.environ.get('SENDGRID_API_KEY'))
                sg.send(message)
            except Exception as e:
                print(e)

    return render_template('dashboard.html',email=session["email"],current_month=json.dumps(session["current_month"]),msg=msg,current_year=json.dumps(session["current_year"]),month_amo
```

## 7.7. Profile routing

```
##### PROFILE #####

@app.route("/profile", methods=('GET', 'POST'))
def profile():
    sql = "SELECT * FROM customers"
    stmt = ibm_db.exec_immediate(conn, sql)
    dictionary = ibm_db.fetch_both(stmt)
    while(True):
        dictionary = ibm_db.fetch_both(stmt)
        if dictionary[2]==session["email"] and dictionary[4]==session["password"]:
            username=ibm_db.result(stmt,1)
            name=ibm_db.result(stmt,0)
            mobile=ibm_db.result(stmt,3)
            session["username"]=username
            break
    return render_template('profile.html',email=session["email"],username=username,mobile=mobile,name=name)
```

In profile page user can see all the user details used to signup.

## 7.8. Add manually page routing

```
##### ADD MANUALLY #####

@app.route("/add_manually" , methods=('GET', 'POST'))
def add_manually():
    msg=""
    if request.method == 'POST':
        amount = request.form['amount']
        category = request.form['category']
        date = request.form['date']

        if not re.match('^[0-9]*$', amount):
            msg = 'Invalid amount !'
            return render_template('add_manually.html',msg1=msg,email=session["email"])
        else:
            insert_sql = "INSERT INTO expenses VALUES (?,?,,?,current_date,current_time)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prepare_stmt, 1, session["email"])
            ibm_db.bind_param(prepare_stmt, 2, amount)
            ibm_db.bind_param(prepare_stmt, 3, category)
            ibm_db.bind_param(prepare_stmt, 4, date)
            ibm_db.execute(prepare_stmt)

            return render_template('add_manually.html',msg2="Added Successfully !",email=session["email"])

    return render_template('add_manually.html',email=session["email"])
```

In the add\_manually page expenses are added manually by user and stored in expenses table in ibmdb2.

## 7.9. Add CSV page routing

```
##### ADD CSV #####

@app.route("/add_csv" , methods=('GET', 'POST'))
def add_csv():
    if request.method == 'POST':
        dataframe = (request.form["csv_file"])
        dataframe1 = pd.read_excel(dataframe)
        for row in dataframe1.iterrows():
            insert_sql = "INSERT INTO expenses VALUES (?,?,,?,current_date,current_time)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prepare_stmt, 1, session["email"])
            ibm_db.bind_param(prepare_stmt, 2, row[1][0])
            ibm_db.bind_param(prepare_stmt, 3, row[1][1])
            ibm_db.bind_param(prepare_stmt, 4, row[1][2])
            ibm_db.execute(prepare_stmt)
        return render_template('add_csv.html',msg="Added Successfully !",email=session["email"])

    return render_template('add_csv.html',email=session["email"])
```

In this page user can add expenses using a csv file.

## 7.10. Table page routing

```
##### TABLE #####

@app.route("/table", methods=('GET', 'POST'))
def table():
    info=[]
    if request.method=="GET":
        query='''select * from expenses'''
        exec_query=ibm_db.exec_immediate(conn,query)
        row=ibm_db.fetch_both(exec_query)
        l=1
        while(row):
            if row[0]==session["email"]:
                info.append([l,row[0],row[1],row[2],row[3],row[4],row[5]])
                l+=1
            row=ibm_db.fetch_both(exec_query)
        return render_template('table.html',info=info,email=session["email"])
```

In table page all the expenses which are uploaded in the expenses table in ibmdb2 are shown.

## 7.11. Set limit page routing

```
##### SET LIMIT #####

@app.route("/set_limit", methods=('GET', 'POST'))
def set_limit():
    msg=""
    if request.method == 'POST':
        monthly = request.form['monthly']
        yearly = request.form['yearly']
        session["monthly"] = monthly
        session["yearly"] = yearly
        insert_sql = "INSERT INTO limits VALUES (?,?,?)"
        prep_stmt = ibm_db.prepare(conn, insert_sql)
        ibm_db.bind_param(prepare_stmt, 1, session["email"])
        ibm_db.bind_param(prepare_stmt, 2, monthly)
        ibm_db.bind_param(prepare_stmt, 3, yearly)
        ibm_db.execute(prepare_stmt)
        return render_template('set_limit.html',msg2="Added Successfully !",email=session["email"])

    return render_template('set_limit.html',email=session["email"])
```

Using this page user can set limit for expenses which will be uploaded in limits table in ibmdb2.

## 7.12. Monthly chart page routing

```
##### MONTHLY CHARTS #####

@app.route("/monthlycharts", methods=('GET', 'POST'))
def monthlycharts():
    expenses={}
    months={}
    if request.method=="GET":
        query='''select * from expenses'''
        exec_query=ibm_db.exec_immediate(conn,query)
        row=ibm_db.fetch_both(exec_query)
        while(row):
            if row['NAME']==session["email"]:
                if row['CATOGERY'] not in expenses:
                    expenses[row['CATOGERY']]=row['AMOUNT']
                else:
                    expenses[row['CATOGERY']]+=row['AMOUNT']

                dates=row['BILL_TIME'][:7]
                amount=row['AMOUNT']
                if dates not in months:
                    months[dates]=amount
                else:
                    months[dates]+=amount

            row=ibm_db.fetch_both(exec_query)
        session["piecategories"]=[i for i in expenses.keys()]
        session["pieamount"]=[i for i in expenses.values()]
        session["barcategories"]=[i for i in months.keys()]
        session["baramount"]=[i for i in months.values()]

        return render_template('monthlycharts.html',email=session["email"],piecategories=json.dumps(session["piecategories"]),pieamount=json.dumps(session["pieamount"]),

    return render_template('monthlycharts.html',email=session["email"])
```

In this page all the overall monthly usage of expense are shown and overall categories are shown.

## 7.13. Yearly chart routing

```
##### YEARLY CHARTS #####

@app.route("/yearlycharts", methods=('GET', 'POST'))
def yearlycharts():
    expenses={}
    yearly={}
    if request.method=="GET":
        query='''select * from expenses'''
        exec_query=ibm_db.exec_immediate(conn,query)
        row=ibm_db.fetch_both(exec_query)
        while(row):
            if row['NAME']==session["email"]:
                if row['CATOGERY'] not in expenses:
                    expenses[row['CATOGERY']]=row['AMOUNT']
                else:
                    expenses[row['CATOGERY']]+=row['AMOUNT']

                dates=row['BILL_TIME'][:4]
                amount=row['AMOUNT']
                if dates not in yearly:
                    yearly[dates]=amount
                else:
                    yearly[dates]+=amount

            row=ibm_db.fetch_both(exec_query)
        session["piecategories"]=[i for i in expenses.keys()]
        session["pieamount"]=[i for i in expenses.values()]
        session["yearbarcategories"]=[i for i in yearly.keys()]
        session["yearbaramount"]=[i for i in yearly.values()]

        return render_template('yearlycharts.html',email=session["email"],piecategories=json.dumps(session["piecategories"]),pieamount=json.dumps(session["pieamount"]),yearbarcategories=json.dumps(session["yearbarcategories"]),yearbaramount=json.dumps(session["yearbaramount"]),

    return render_template('yearlycharts.html',email=session["email"])
```

In this page all the overall yearly usage of expense are shown and overall categories are shown.

## 7.14. Logout page routing

```
##### LOG OUT #####

@app.route("/logout")
def logout():
    session.pop("email",default=None)
    session.pop("password",default=None)
    session.pop("username",default=None)
    session.pop("monthly",default=None)
    session.pop("yearly",default=None)
    session.pop("piecategories",default=None)
    session.pop("pieamount",default=None)
    session.pop("barcategories",default=None)
    session.pop("baramount",default=None)
    session.pop("yearbarcategories",default=None)
    session.pop("yearbaramount",default=None)
    return redirect("/login")

if __name__ == '__main__':
    app.run(host="0.0.0.0",port="8000")
```

In this page all the session variables used are deleted and user is logged out and this page is redirects to login page.

## 7.15. Dockerfile

```
Dockerfile > ...
1 FROM python:3.10.7
2 WORKDIR /app
3 COPY requirements.txt ./
4 RUN pip install -r requirements.txt
5 COPY . .
6 EXPOSE 8000
7 CMD ["python","./app.py"]
```

It has the information for creating image for this application which will containerize the application in docker desktop.

## 7.16. Requirements file

```
requirements.txt
1 ibm_db
2 pandas
3 matplotlib
4 sendgrid
5 flask
6 flask_session
```

It tell the packages needed to install while creating a new environment.

## 7.17. deployment.yaml

```
! deployment.yaml > {} spec > {} template > {} spec > [ ] containers > {} 0 > [ ] ports > {} 0 > # containerPort
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: peta-application
5  spec:
6    selector:
7      app: peta-application
8    ports:
9      - protocol: "TCP"
10        port: 8000
11        targetPort: 8000
12        type: LoadBalancer
13
14  ---
15  apiVersion: apps/v1
16  kind: Deployment
17  metadata:
18    name: peta-application
19  spec:
20    selector:
21      matchLabels:
22        app: peta-application
23    replicas: 5
24    template:
25      metadata:
26        labels:
27          app: peta-application
28      spec:
29        containers:
30          - name: peta-application
31            image: peta-application
32            imagePullPolicy: IfNotPresent
33            ports:
34              - containerPort: 8000
```

This file is used to configure the image to Kubernetes cluster.

## 7.18. Customers Database

Name	Data type	Nullable	Length	Scale
NAME	VARCHAR	Y	255	0
USERNAME	VARCHAR	Y	255	0
EMAIL	VARCHAR	Y	255	0
MOBILENO	VARCHAR	Y	255	0
PASSWORD	VARCHAR	Y	255	0

In this database all the customer (user) details are stored.

## 7.19. Expenses Database

Name	Data type	Nullable	Length	Scale
NAME	VARCHAR	Y	255	0
AMOUNT	INTEGER	Y		0
CATOGERY	VARCHAR	Y	255	0
BILL_TIME	VARCHAR	Y	255	0
UPLOAD_DATE	VARCHAR	Y	255	0
UPLOAD_TIME	VARCHAR	Y	255	0

In this database all the expenses are stored.



7.20. Limits database

Name	Data type	Nullable	Length	Scale
NAME	VARCHAR	Y	255	0
MONTHLY_LIMIT	INTEGER	Y		0
YEARLY_LIMIT	INTEGER	Y		0

In this database all the use expense limits are stored.

8. TESTING

8.1. Test cases

Test cases for registration

Prem	Prem	sec19cs095@sairamtap.edu.in	0985764867	prem
Prem Kashyap	Prem	iamprem@gmail.com	8789806965	Prem@3456
Prem Kashyap	Prem	iamprem.2001@gmail.com	9502645945	iamprem
Prem Kashyap	Prem	prem123@gmail.com	9999999999	prem
Prem Kashyap	Prem	prem@gmail.com	0985764867	prem
Prem Kashyap	Prem	123asdfsdf@gmail.com	0985764867	asdfsdf
Prem Kashyap	Prem	nmani3008@gmail.com	0985764867	1234
Sai	Sai	Sai@gmail.com	0985764867	sai@123
Wick	Wick	wick2@gmail.com	0985764867	wick@123
abc123	abc123	abc123@gmail.com	1234567890	1234

Test cases for Expenses

iamprem.2001@gmail.com	10000	Transport	2022-11-09	2022-11-18	11.40.23
iamprem.2001@gmail.com	456	Bills and Fees	2022-07-11	2022-11-18	12.53.06
iamprem.2001@gmail.com	78686	Transport	2022-05-11	2022-11-18	12.53.05
iamprem.2001@gmail.com	77	Shopping	2022-04-11	2022-11-18	12.53.04
iamprem.2001@gmail.com	464	Food	2022-01-17	2022-11-18	12.53.03
iamprem.2001@gmail.com	43543	Household	2022-01-15	2022-11-18	12.53.03
iamprem.2001@gmail.com	343	Others	2022-01-11	2022-11-18	12.53.02

Test cases for limits

iamprem.2001@gmail.com	100000	1200000
iamprem.2001@gmail.com	100000	1200000
iamprem.2001@gmail.com	1	1
iamprem.2001@gmail.com	100000	1200000
iamprem.2001@gmail.com	1	1
prem123@gmail.com	100000	1200000
prem123@gmail.com	100000	1200000
prem123@gmail.com	1	1
prem123@gmail.com	50000	500000

8.2. User Acceptance Testing

				Date	15-Nov-22								
				Team ID	PNT/2022/MID/03865								
				Project Name	Personal Expense Tracker Application								
				Maximum Marks	4 marks								
Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID	Executed By
Home Page_TC_001	Functional , UI	Home Page	User should able to click the available buttons in the navigation bar to perform specific operations	None	1.Goto Homepage	Username: iamprem@gmail.com js password: iamprem@123	Application should show below UI elements: Navigation bar contains the following elements: a. Home button b. About button c. Register button d. Login button	Working as expected	Pass				Prem Kashyap
About Page_TC_002	Functional , UI	About Page	User should able to login and register in the website	Home	1.Click to login 2. Click to register	Username: iamprem@gmail.com js password: iamprem@123	Application should show below UI elements: a. Login button b. Register button	Working as expected	Pass				Sai Dayakar
Signup Page_TC_003	Functional , UI	Signup page	User should able to register into the website	Username, email address, mobile no, password	1.Enter details to register	Username: iamprem@gmail.com js password: iamprem@123	User should able to see the below message: a. Registration Successful	Working as expected	Pass				Arosha Nivin
Signup Page_TC_004	Functional , UI	Login page	Application should verify the user details and should enter the website	Login First	1.Enter details and click login	Username: iampremi23@gmail.com password: iamprem@321	Application should show already registered	Working as expected	Pass				Dhruvi Kumar
Login Page_TC_005	Functional , UI	Login page	User should able to login into the website and should get an error message using invalid credentials	Email address and password	1.Enter details and click login	Username: iamprem@gmail.com js password: iamprem@123	Application should show 'Incorrect credentials' message and should show login, register and home button	Working as expected	Pass				Prem Kashyap
LoginPage_TC_006	Functional , UI	Login page	Application should verify user credentials and should provide registration successful	Login First	1.Enter details and click login	Username: iamprem@gmail.com js	Application should show the following elements after successful login: a. Dashboard	Working as expected	Pass				Sai Dayakar
						password: iamprem@123							
ExpensePage_TC_0	Functional	Expense Page	Verify whether user is able to add expense	Add some expenses	Add date,expense name and amount	Food:2000	Application adds expenses	Working as expected	Pass				Sai Dayakar
ExpensePage_TC_0	Functional	Expense Page	Verify whether user is able to add expense	Add some expenses	Add date,expense name and amount	Cloths:500	Application adds expenses	Working as expected	Pass				Arosha Nivin
ExpensePage_TC_0	Functional	Expense Page	Verify whether user is able to verify expense	Update expenses	1.Update expenses	Update Food:3400	Application updates expenses	Working as expected	Pass				Sai Dayakar
AddLimit_TC_007	Functional	Add limit	Verify whether user is able to add limit or not	Add Limit	1.Add Limit	Limit:50000	Application add limits for expenses	Working as expected	Pass				Sai Dayakar
SendEmail Integration	Functional	Send email	Sending email after registration and if expense limit exceeds	Add email	Email	iampremi2001@gmail.com	Receive email to the email	Working as expected	Pass				Prem Kashyap


9. RESULTS

9.1. Performance Metrics


			NFT - Risk Assessment					
S.No	Project Name	Scope/feature	Functional Changes	Software Changes	Impact of Downtime	Load/Volumen Changes	Risk Score	Justification
1	Personal Expense Tracker Application	New	Low	High	Low	>10 to 30%	GREEN	Using this application user can manage and track all the expenses and set limits so that it is easier to stay in budget limits
			NFT - Detailed Test Plan					
S.No	Project Overview	NFT Test approach	Assumptions/Dependencies/Risks	Approvals/SignOff				
1	Personal Expense Tracker Application	Stress	App Crash/ Developer team/ Site Down	Approved				
2	Personal Expense Tracker Application	Load	Server Crash/ Developer team/ Server Down	Approved				
			End Of Test Report					
S.No	Project Overview	NFT Test approach	NFR - Met	Test Outcome	GO/NO-GO decision	Recommendations	Identified Defects (Detected/Closed/Open)	Approvals/SignOff
1	Personal Expense Tracker Application	Stress	Performance	CPU -01	GO	Amazon web services to increase Scalability	Closed	Approved
2	Personal Expense Tracker Application	Load	Scalability	DB Storage - 01	GO	Using free platforms to increase the data storage capability there we can increase user base and organizations on site.	Closed	Approved





## 9.2. Outputs



# SIGN-UP

 Username

 E-Mail

 Password

CREATE ACCOUNT

I ALREADY HAVE AN ACCOUNT

## Add Expense

Date

Time

Expense name

Expense Amount

Pay-Mode



Category



Add

## Edit Expense

Date

15-11-2022



Time

12:39



Expense name

Camera Rent

Expense Amount

4000

creditcard



rent



Update

2022-11-15

Camera Rent

₹ 4000

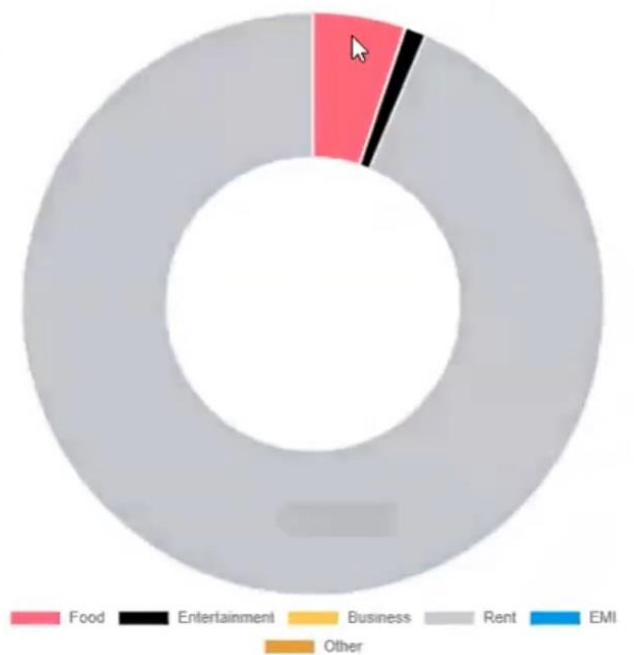
creditcard

rent

Edit

Delete

EXPENSE BREAKDOWN



### Expense Breakdown

Food	220
Entertainment	50
Business	0
Rent	4000
EMI	0
Other	0
Total	₹ 4270

## EXPENSES

2022-11-17

Snacks

₹ 220

cash

food

Edit

Delete

2022-11-16

Mov̤ie

₹ 50

cash

entertainment

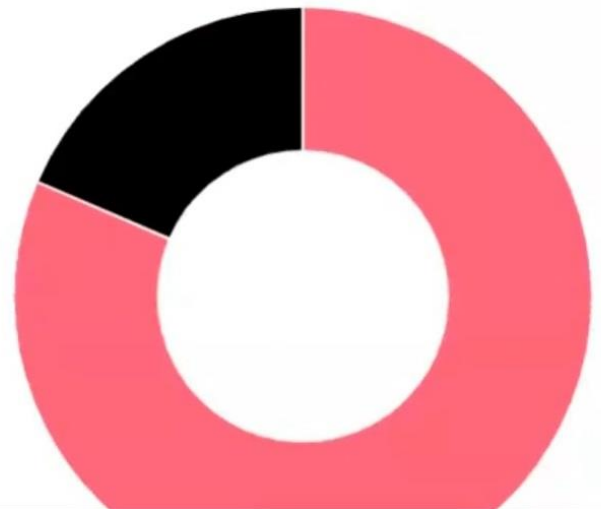
Edit

Delete

EXPENSE BREAKDOWN

### Expense Breakdown

Food	220
Entertainment	50
Business	0
Rent	0
EMI	0



Currently your MONTHLY limit is ₹ (300,)

ENTER the MONTHLY LIMIT to avoid over EXPENSES

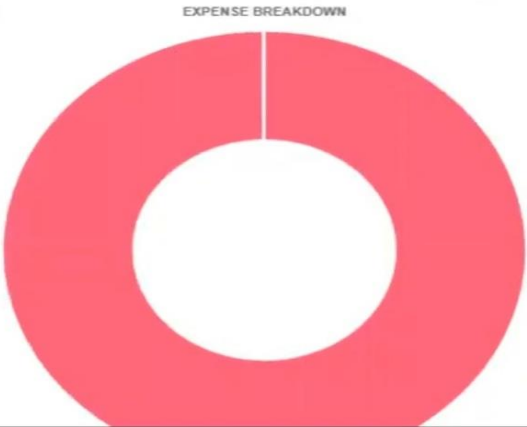
ENTER

Today Expense Breakdown

TIME	AMOUNT
20:37:00	220

Expense Breakdown BY Category

Food	220
Entertainment	0
Business	0
Rent	0
EMI	0

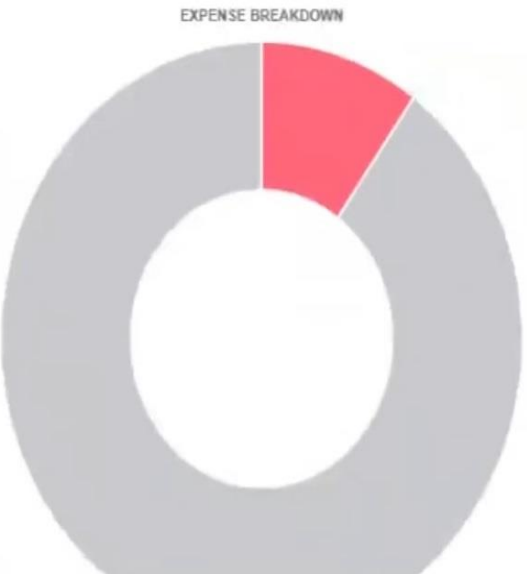


MONTH Expense Breakdown

MONTH	AMOUNT
November	2220

Expense Breakdown BY Category:  
CURRENT MONTH

Food	220
Entertainment	0
Business	0
Rent	2000

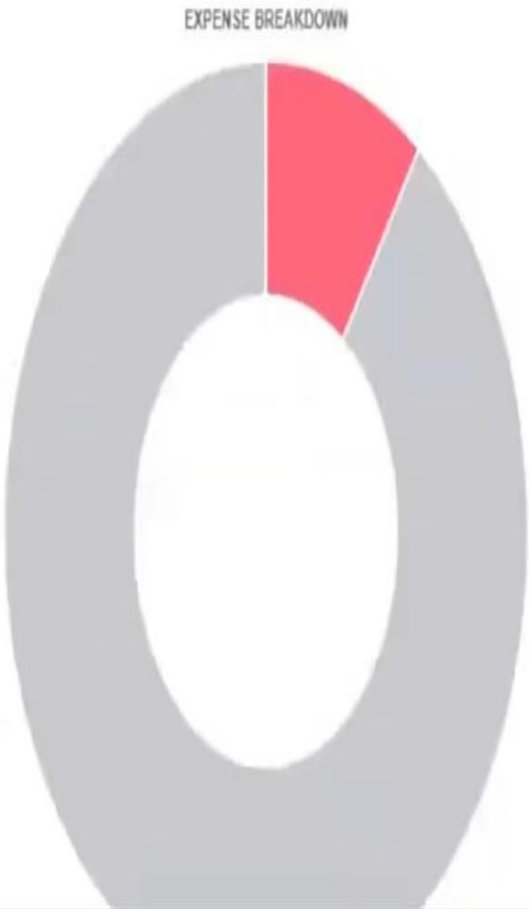


# YEAR Expense Breakdown

YEAR	AMOUNT
2022	2220

## Expense Breakdown BY Category: CURRENT YEAR

Food	220
Entertainment	0
Business	0
Rent	2000



## **ADVANTAGES DISADVANTAGES Advantages:**

- Easy to use.
- Simplicity
- Reliability
- Ability to collaborate
- Expense tracking

## **DISADVANTAGES**

- System crash
- Security attacks
- Incorrect Information
- Time taking for adding expenses.

## **10. CONCLUSION**

The conclusion for this project is that the Personal Expense Tracker Application that we have created in this project is used to track and manage all the expenses of a user, graphical interface of the application is easy to use. This application sends alert if user exceeds the budget limit for the current year or month.

## **11. FUTURE SCOPE**

- Future scope of this project is to integrate the application to detect online transactions done by the user and automatically add the expense details to the cloud.
- Develop a mobile app which can be used in no internet conditions also.

## **12. APPENDIX**

**Source code:**

**app.py:**

```
import re,ibm_db
import pandas as pd
import matplotlib.pyplot as plt
import json
import os
import openpyxl
```

```
from sendgrid import SendGridAPIClient
```

```
from sendgrid.helpers.mail import Mail
```

```
from flask import (Flask, flash, redirect, render_template, request, session,url_for)
```

```
from flask_session import Session
```

```
from datetime import date
```

```
app = Flask(_name_)
```

```
app.debug=True
```

```
app.config["SESSION_PERMANENT"] = False
```

```
app.config["SESSION_TYPE"] = "filesystem"
```

```
app.config["SECRET_KEY"] = "Personal_Expense_Tracker_Application"
```

```
official_mail="expensetracker0046@gmail.com"
```

```
Session(app)
```

```
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=30875;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=hhl91628;PWD=vM4lAZjxo4LsBPoJ", "", "")
```

```
#####
```

HOME

```
@app.route("/")
```

```
def home():
```

```
    return render_template("home.html")
```

```
#####
```

SIGN UP

```
@app.route("/signup", methods=('GET', 'POST'))
```

```
def signup():
```

```
    msg=""
```

```
    if request.method == 'POST':
```

```
        name = request.form['fname']
```

```
        username =name.split(" ")[0]
```

```
        email = request.form['femail']
```

```

mobile = request.form['mobile']
password = request.form['password']

sql = "SELECT * FROM customers WHERE email=?"
stmt = ibm_db.prepare(conn, sql)
ibm_db.bind_param(stmt,1,email)
ibm_db.execute(stmt)
account = ibm_db.fetch_assoc(stmt)

if account:
    return render_template("already_registered.html")
elif not re.match('[^@]+@[^@]+\.[^@]+', email):
    msg = 'Invalid email address !'
    return render_template('signup.html',msg=msg)
elif not re.match(r'^[a-zA-Z0-9]+([_ ]?[a-zA-Z0-9]+)*$', name):
    msg = 'Name must contain only characters, numbers !'
    return render_template('signup.html',msg=msg)
elif not re.match(r'^(\+?\d{1,2}\s?)?1?\-?\.?\s?(?(\d{3}\)?[s.-]?\d{3}[s.-]?\d{4})$', mobile):
    msg = 'Enter valid mobile number!'
    return render_template('signup.html',msg=msg)
else:
    insert_sql = "INSERT INTO customers VALUES (?, ?, ?, ?, ?)"
    prep_stmt = ibm_db.prepare(conn, insert_sql)
    ibm_db.bind_param(prepare_stmt, 1, name)
    ibm_db.bind_param(prepare_stmt, 2, username)
    ibm_db.bind_param(prepare_stmt, 3, email)
    ibm_db.bind_param(prepare_stmt, 4, mobile)
    ibm_db.bind_param(prepare_stmt, 5, password)
    ibm_db.execute(prepare_stmt)

message = Mail(
    from_email = official_mail,
    to_emails = email,
    subject='Registration Successful',

    html_content='<h1>Hello '+name+', <br>Welcome to Personal Expense Tracker
Application (PETA) .<br>Hope we have a great journey ahead.</h1>')

```



```

try:
    sg = SendGridAPIClient(os.environ.get('SENDGRID_API_KEY'))
    sg.send(message)
except Exception as e:
    print(e)

return render_template('registration_success.html')

```

```

return render_template('signup.html')

```

```

#####
#####

```

ABOUT

```

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

```

```

#####
#####

```

SIGN IN

```

@app.route("/signin" , methods=('GET', 'POST'))
def signin():
    if request.method == 'POST':
        femail = request.form['femail']
        fpassword = request.form['password']
        session["email"]=femail
        session["password"]=fpassword

        sql = "SELECT * FROM customers where email=? and password=?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt,1,femail)

```

```

        ibm_db.bind_param(stmt,2,fpassword)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)

    if account:
        return redirect("/dashboard")

    else:
        return render_template('invalid_credentials.html')

return render_template('signin.html')

```

```

#####
#####

```

DASHBOARD

```

@app.route("/dashboard" , methods=('GET', 'POST'))
def dashboard():
    ml=0
    yl=0
    limit=False
    msg=""
    months={ }
    years={ }
    session["date"] = date.today()
    session["date"] = str(session["date"])
    if request.method=="GET":
        query2="select * from expenses"
        exec_query=ibm_db.exec_immediate(conn,query2)
        row=ibm_db.fetch_both(exec_query)
        user=False
        while(row):
            if row[0]==session["email"]:
                if row[3][:7]==session["date"][:7]:
                    if row[3] not in months:

```

```

        months[row[3]]=row[1]
    else:
        months[row[3]]+=row[1]
    if row[3][:4]==session["date"][:4]:
        if row[3][:7] not in years:
            years[row[3][:7]]=row[1]
        else:
            years[row[3][:7]]+=row[1]
    row=ibm_db.fetch_both(exec_query)

```

```

query1="select * from limits"
exec_query=ibm_db.exec_immediate(conn,query1)
row=ibm_db.fetch_both(exec_query)
k=False
while(row):
    if row[0]==session["email"]:
        ml=row[1]
        yl=row[2]

    row=ibm_db.fetch_both(exec_query)

```

```

session["current_month"]=[i for i in months.keys()]
session["month_amount"]=[i for i in months.values()]
session["current_year"]=[i for i in years.keys()]
session["year_amount"]=[i for i in years.values()]

```

```

if ml==0 and yl==0:
    msg="No expense limits are set, please set them to use the application"
    if sum(session["month_amount"])>ml or sum(session["year_amount"])>yl:
        limit=True

```

```

if limit==True:
    msg="Expense limit exceed Please manage your expenses."
    message = Mail(
        from_email = official_mail,
        to_emails = session["email"],

```

```

        subject='Expense Limit reached',

        html_content='<h1> Monthly expense limit reached.<br>Please use money carefully or
increase budget limit.</h1>')

    try:

        sg = SendGridAPIClient(os.environ.get('SENDGRID_API_KEY'))

        sg.send(message)

    except Exception as e:

        print(e)

    return

render_template('dashboard.html',email=session["email"],current_month=json.dumps(session["
current_month"]),msg=msg,current_year=json.dumps(session["current_year"]),month_amount
=json.dumps(session["month_amount"]),year_amount=json.dumps(session["year_amount"]),m
onthly_a=sum(session["month_amount"]),yearly_a=sum(session["year_amount"]),monthly_l=
ml,yearly_l=yl)

```

```

##### PROFILE
#####

```

```

@app.route("/profile" , methods=('GET', 'POST'))
def profile():

    sql = "SELECT * FROM customers"

    stmt = ibm_db.exec_immediate(conn, sql)

    dictionary = ibm_db.fetch_both(stmt)

    while(True):

        dictionary = ibm_db.fetch_both(stmt)

        if dictionary[2]==session["email"] and dictionary[4]==session["password"]:

            username=ibm_db.result(stmt,1)

            name=ibm_db.result(stmt,0)

            mobile=ibm_db.result(stmt,3)

            session["username"]=username

            break

```

```

return
render_template('profile.html',email=session["email"],username=username,mobile=mobile,name=name)

```

```

##### ADD MANUALLY
#####

```

```

@app.route("/add_manually" , methods=('GET', 'POST'))
def add_manually():
    msg=""
    if request.method == 'POST':
        amount = request.form['amount']
        category = request.form['category']
        date = request.form['date']

        if not re.match('^[0-9]*$', amount):
            msg = 'Invalid amount !'
            return render_template('add_manually.html',msg1=msg,email=session["email"])
        else:
            insert_sql = "INSERT INTO expenses VALUES (?, ?, ?, ?, current_date, current_time)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prepare_stmt, 1, session["email"])
            ibm_db.bind_param(prepare_stmt, 2, amount)
            ibm_db.bind_param(prepare_stmt, 3, category)
            ibm_db.bind_param(prepare_stmt, 4, date)
            ibm_db.execute(prepare_stmt)

            return render_template('add_manually.html',msg2="Added Successfully",email=session["email"])

    return render_template('add_manually.html',email=session["email"])

```

```
#####  
#####
```

ADD CSV

```
@app.route("/add_csv" , methods=('GET', 'POST'))  
def add_csv():  
    if request.method == 'POST':  
        dataframe = (request.form["csv_file"])  
        if dataframe.split(".")[1] == 'xlsx':  
            df = pd.read_excel(dataframe.read(), engine='openpyxl')  
        elif dataframe.split(".")[1] == 'xls':  
            df = pd.read_excel(dataframe.read())  
        elif dataframe.split(".")[1] == 'csv':  
            df = pd.read_csv(dataframe.read())  
        for row in df.iterrows():  
            insert_sql = "INSERT INTO expenses VALUES (?, ?, ?, ?, current_date, current_time)"  
            prep_stmt = ibm_db.prepare(conn, insert_sql)  
            ibm_db.bind_param(prepare_stmt, 1, session["email"])  
            ibm_db.bind_param(prepare_stmt, 2, row[1][0])  
            ibm_db.bind_param(prepare_stmt, 3, row[1][1])  
            ibm_db.bind_param(prepare_stmt, 4, row[1][2])  
            ibm_db.execute(prepare_stmt)  
            return render_template('add_csv.html', msg="Added Successfully  
!", email=session["email"])  
  
        return render_template('add_csv.html', email=session["email"])
```

```
#####  
#####
```

TABLE

```
@app.route("/table", methods=('GET', 'POST'))  
def table():  
    info=[]
```

```

if request.method=="GET":
    query="select * from expenses"
    exec_query=ibm_db.exec_immediate(conn,query)
    row=ibm_db.fetch_both(exec_query)
    l=1
    while(row):
        if row[0]==session["email"]:
            info.append([l,row[0],row[1],row[2],row[3],row[4],row[5]])
            l+=1
        row=ibm_db.fetch_both(exec_query)
    return render_template('table.html',info=info,email=session["email"])

```

```

##### SET LIMIT
#####

```

```

@app.route("/set_limit" , methods=('GET', 'POST'))
def set_limit():
    msg=""
    if request.method == 'POST':
        monthly = request.form['monthly']
        yearly = request.form['yearly']
        session["monthly"] = monthly
        session["yearly"] = yearly
        insert_sql = "INSERT INTO limits VALUES (?,?)"
        prep_stmt = ibm_db.prepare(conn, insert_sql)
        ibm_db.bind_param(prepare_stmt, 1, session["email"])
        ibm_db.bind_param(prepare_stmt, 2, monthly)
        ibm_db.bind_param(prepare_stmt, 3, yearly)
        ibm_db.execute(prepare_stmt)
        return render_template('set_limit.html',msg2="Added Successfully
!",email=session["email"])

    return render_template('set_limit.html',email=session["email"])

```

```
##### MONTHLY
CHARTS #####
```

```
@app.route("/monthlycharts" , methods=('GET', 'POST'))
def monthlycharts():
    expenses={}
    months={}
    if request.method=="GET":
        query="select * from expenses"
        exec_query=ibm_db.exec_immediate(conn,query)
        row=ibm_db.fetch_both(exec_query)
        while(row):
            if row['NAME']==session["email"]:
                if row['CATOGERY'] not in expenses:
                    expenses[row['CATOGERY']]=row['AMOUNT']
                else:
                    expenses[row['CATOGERY']]+=row['AMOUNT']

            dates=row['BILL_TIME'][:7]
            amount=row['AMOUNT']
            if dates not in months:
                months[dates]=amount
            else:
                months[dates]+=amount

            row=ibm_db.fetch_both(exec_query)
        session["piecategories"]=[i for i in expenses.keys()]
        session["pieamount"]=[i for i in expenses.values()]
        session["barcategories"]=[i for i in months.keys()]
        session["baramount"]=[i for i in months.values()]

        return

render_template('monthlycharts.html',email=session["email"],piecategories=json.dumps(sessio
```



```
n["piecategories"],pieamount=json.dumps(session["pieamount"]),barcategories=json.dumps(session["barcategories"]),baramount=json.dumps(session["baramount"]),length=json.dumps(len(session["barcategories"])),s=sum(session["baramount"]))
```

```
return render_template('monthlycharts.html',email=session["email"])
```

```
##### YEARLY CHARTS
#####
```

```
@app.route("/yearlycharts" , methods=('GET', 'POST'))
def yearlycharts():
    expenses={ }
    yearly={ }
    if request.method=="GET":
        query="select * from expenses"
        exec_query=ibm_db.exec_immediate(conn,query)
        row=ibm_db.fetch_both(exec_query)
        while(row):
            if row['NAME']==session["email"]:
                if row['CATOGERY'] not in expenses:
                    expenses[row['CATOGERY']]=row['AMOUNT']
                else:
                    expenses[row['CATOGERY']]+=row['AMOUNT']

            dates=row['BILL_TIME'][:4]
            amount=row['AMOUNT']
            if dates not in yearly:
                yearly[dates]=amount
            else:
                yearly[dates]+=amount

        row=ibm_db.fetch_both(exec_query)
        session["piecategories"]=[i for i in expenses.keys()]
```

```

        session["pieamount"]=[i for i in expenses.values()]
        session["yearbarcategories"]=[i for i in yearly.keys()]
        session["yearbaramount"]=[i for i in yearly.values()]

    return

render_template('yearlycharts.html',email=session["email"],piecategories=json.dumps(session["piecategories"]),pieamount=json.dumps(session["pieamount"]),barcategories=json.dumps(session["yearbarcategories"]),baramount=json.dumps(session["yearbaramount"]),length=json.dumps(len(session["yearbarcategories"])),s=sum(session["pieamount"]))

return render_template('yearlycharts.html',email=session["email"])

##### LOG OUT #####
#####

@app.route("/logout")
def logout():
    session.pop("email",default=None)
    session.pop("password",default=None)
    session.pop("username",default=None)
    session.pop("monthly",default=None)
    session.pop("yearly",default=None)
    session.pop("piecategories",default=None)
    session.pop("pieamount",default=None)
    session.pop("barcategories",default=None)
    session.pop("baramount",default=None)
    session.pop("yearbarcategories",default=None)
    session.pop("yearbaramount",default=None)
    return redirect("/login")

if __name__ == '__main__':
    app.run(host="0.0.0.0",port="8000")

```

## home.html:

```
<!DOCTYPE html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Home</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css"
rel="stylesheet">
  <link href="https://getbootstrap.com/docs/5.2/assets/css/docs.css" rel="stylesheet">
  <link rel="canonical" href="https://getbootstrap.com/docs/5.2/examples/cover/">
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"></script>
  <link href="cover.css" rel="stylesheet">
  <script>
    window.watsonAssistantChatOptions = {
      integrationID: "e15973c3-bf08-44d2-b1a3-fcaef746b52c", // The ID of this integration.
      region: "au-syd", // The region your integration is hosted in.
      serviceInstanceID: "561acd5a-e866-4ae1-8b04-999c9b3d1b84", // The ID of your service
instance.
      onLoad: function(instance) { instance.render(); }
    };
    setTimeout(function(){
      const t=document.createElement('script');
      t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" +
(window.watsonAssistantChatOptions.clientVersion || 'latest') +
"/WatsonAssistantChatEntry.js";
      document.head.appendChild(t);
    });
  </script>
  <nav class="navbar navbar-expand-md navbar-dark bg-dark shadow p-3 mb-5">
    <div class="container-fluid">
      <div class="mx-auto order-0">
        <a class="navbar-brand mx-auto" href="/">Personal Expense Tracker
Application</a>
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target=".dual-collapse2">
```

```
        <span class="navbar-toggler-icon"></span>
    </button>
</div>
<div class="navbar-collapse collapse w-100 order-3 dual-collapse2">
    <ul class="navbar-nav ms-auto">
        <li class="nav-item active">
            <a class="nav-link" href="/">Home</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="about">About</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="signup">Register</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="signin">Login</a>
        </li>
    </ul>
</div>
</div>
</nav>
</head>

<body class="text-bg-white">
    <div class="cover-container d-flex w-100 h-100 p-3 mx-auto flex-column">

        <div class="row mb-3">

            <div class="col-6 themed-grid-col">
                
            </div>

            <div class="col-6 themed-grid-col">
                <h1>Personal Expense Tracker Application</h1>
                <br>
```

<p class="lead">Personal Expense Tracker is an application which helps the customer to track and manage all the expenses that are done by them.</p>

<p class="lead">

<a href="about" class="btn btn-lg btn-secondary fw-bold border-white bg-dark">Learn more</a>

</p>

</div>

</div>

</body>

### About.html:

<!DOCTYPE html>

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>About</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css" rel="stylesheet">

<link href="https://getbootstrap.com/docs/5.2/assets/css/docs.css" rel="stylesheet">

<script

src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"></script>

<link href="static\css\features.css" rel="stylesheet">

<link rel="canonical" href="https://getbootstrap.com/docs/5.2/examples/features/">

<link href=" ../assets/dist/css/bootstrap.min.css" rel="stylesheet">

<script>

window.watsonAssistantChatOptions = {

integrationID: "e15973c3-bf08-44d2-b1a3-fcaef746b52c", // The ID of this integration.

region: "au-syd", // The region your integration is hosted in.

serviceInstanceID: "561acd5a-e866-4ae1-8b04-999c9b3d1b84", // The ID of your service instance.

onLoad: function(instance) { instance.render(); }

};

setTimeout(function(){

const t=document.createElement('script');

```

        t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" +
(window.watsonAssistantChatOptions.clientVersion || 'latest') +
"/WatsonAssistantChatEntry.js";
        document.head.appendChild(t);
    });
</script>
</head>
<body>
    <nav class="navbar navbar-expand-md navbar-dark bg-dark shadow p-3 mb-5">
        <div class="container-fluid">
            <div class="mx-auto order-0">
                <a class="navbar-brand mx-auto" href="/">Personal Expense Tracker
Application</a>
                <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target=".dual-collapse2">
                    <span class="navbar-toggler-icon"></span>
                </button>
            </div>
            <div class="navbar-collapse collapse w-100 order-3 dual-collapse2">
                <ul class="navbar-nav ms-auto">
                    <li class="nav-item active">
                        <a class="nav-link" href="/">Home</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="signup">Register</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="signin">Login</a>
                    </li>
                </ul>
            </div>
        </div>
    </nav>

    <div class="container px-4 py-5">

```

```

<div class="row row-cols-1 row-cols-md-2 align-items-md-center g-5 py-5">
  <div class="d-flex flex-column align-items-start gap-2">
    <h3 class="fw-bold">Personal Expense Tracker Application</h3>
    <p class="text-muted">Personal Expense Tracker is an application which helps the
customer to track and manage all the expenses that are done by them.<br>Some of the
important features are mentioned in the side section.</p>
    <h2 class="pb-2 border-bottom"></h2>
    <p>If you are interested you can register using below register button.<br>If you are
already a member , Thanks for using the application. You can login using below login
button.</p>
    <div>
      <h2 class="pb-2 border-bottom"></h2>
      <a href="signin" class="btn btn-primary btn-lg">Login</a>
      <a href="signup" class="btn btn-primary btn-lg">Register</a>
    </div>
  </div>
  <div class="row row-cols-1 row-cols-sm-2 g-4">
    <div class="d-flex flex-column gap-2">
      <div
        class="feature-icon-small d-inline-flex align-items-center justify-content-center text-
bg-primary bg-gradient fs-4 rounded-3">
        <svg class="bi" width="1em" height="1em">
          <svg xmlns="http://www.w3.org/2000/svg" width="22" height="22"
fill="currentColor" class="bi bi-wallet2" viewBox="0 0 16 16">
            <path d="M12.136 3.26A1.5 1.5 0 0 1 14 1.78V3h.5A1.5 1.5 0 0 1 16 4.5v9a1.5
1.5 0 0 1-1.5 1.5h-13A1.5 1.5 0 0 1 0 13.5v-9a1.5 1.5 0 0 1 1.432-1.499L12.136 3.26zM5.562
3H13V1.78a.5.5 0 0 0-.621-.484L5.562 3zM1.5 4a.5.5 0 0 0-.5.5v9a.5.5 0 0 0 .5.5h13a.5.5 0 0
0 .5-.5v-9a.5.5 0 0 0-.5-.5h-13z"/>
          </svg>
        </svg>
      </div>
      <h4 class="fw-semibold mb-0">Expense Tracking</h4>
      <p class="text-muted">Bills and money spent by the member using this application
can track their expenses and find how much they have spent in a particular time period.</p>
    </div>
  </div>
  <div class="d-flex flex-column gap-2">

```





```

<div class="d-flex flex-column gap-2">
  <div
    class="feature-icon-small d-inline-flex align-items-center justify-content-center text-
    bg-primary bg-gradient fs-4 rounded-3">
    <svg class="bi" width="1em" height="1em">
      <svg xmlns="http://www.w3.org/2000/svg" width="22" height="22"
      fill="currentColor" class="bi bi-ui-checks-grid" viewBox="0 0 16 16">
        <path d="M2 10h3a1 1 0 0 1 1 1v3a1 1 0 0 1-1 1H2a1 1 0 0 1-1-1v-3a1 1 0 0 1 1-
        1zm9-9h3a1 1 0 0 1 1 1v3a1 1 0 0 1-1 1h-3a1 1 0 0 1-1-1V2a1 1 0 0 1 1-1zm0 9a1 1 0 0 0-1
        1v3a1 1 0 0 0 1 1h3a1 1 0 0 0 1-1v-3a1 1 0 0 0-1-1h-3zm0-10a2 2 0 0 0-2 2v3a2 2 0 0 0 2
        2h3a2 2 0 0 0 2-2V2a2 2 0 0 0-2-2h-3zM2 9a2 2 0 0 0-2 2v3a2 2 0 0 0 2 2h3a2 2 0 0 0 2-2v-
        3a2 2 0 0 0-2-2H2zm7 2a2 2 0 0 1 2-2h3a2 2 0 0 1 2 2v3a2 2 0 0 1-2 2h-3a2 2 0 0 1-2-2v-3zM0
        2a2 2 0 0 1 2-2h3a2 2 0 0 1 2 2v3a2 2 0 0 1-2 2H2a2 2 0 0 1-2-2V2zm5.354 8.54a.5.5 0 1 0-
        .708-.708L3 3.793l-.646-.647a.5.5 0 1 0-.708.708l1 1a.5.5 0 0 0 .708 0l2-2z"/>
      </svg>
    </svg>
  </div>
  <h4 class="fw-semibold mb-0">Categorized Expenses</h4>
  <p class="text-muted">All the expenses are catogerized according to the type of
  usage like food , entertainment e.t.c.</p>
</div>
</div>
</div>
</div>
</main>

```

```

<script src="../assets/dist/js/bootstrap.bundle.min.js"></script>

```

```

</body>
</html>

```

## Signup.html:

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="">
    <title>Register</title>
    <link rel="canonical" href="https://getbootstrap.com/docs/5.2/examples/heroes/">
    <link href="static\css\assets\dist\css\bootstrap.min.css" rel="stylesheet">
    <link href="heroes.css" rel="stylesheet">
    <script>
      window.watsonAssistantChatOptions = {
        integrationID: "e15973c3-bf08-44d2-b1a3-fcaef746b52c", // The ID of this integration.
        region: "au-syd", // The region your integration is hosted in.
        serviceInstanceID: "561acd5a-e866-4ae1-8b04-999c9b3d1b84", // The ID of your
service instance.
        onLoad: function(instance) { instance.render(); }
      };
      setTimeout(function(){
        const t=document.createElement('script');
          t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" +
(window.watsonAssistantChatOptions.clientVersion || 'latest') +
"/WatsonAssistantChatEntry.js";
        document.head.appendChild(t);
      });
    </script>
  </head>
  <body>
    <nav class="navbar navbar-expand-md navbar-dark bg-dark shadow p-3 mb-5">
      <div class="container-fluid">
        <div class="mx-auto order-0">
          <a class="navbar-brand mx-auto" href="/">Personal Expense Tracker
Application</a>
          <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target=".dual-collapse2">
```

```

        <span class="navbar-toggler-icon"></span>
    </button>
</div>
<div class="navbar-collapse collapse w-100 order-3 dual-collapse2">
    <ul class="navbar-nav ms-auto">
        <li class="nav-item active">
            <a class="nav-link" href="/">Home</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="signin">Login</a>
        </li>
    </ul>
</div>
</div>
</nav>
</div>
<main>
    <div class="container -fluid">
        <div class="row align-items-center row mb-3">
            <div class="col-sm-4 glass">
                <form class="shadow p-3 mb-5 bg-white rounded p-4 p-md-5 border rounded-3
bg-light needs-validation" novalidate action = "{{ url_for('signup') }}" method = "POST">
                    <h2 class="fw-bold mb-2 text-uppercase">Register</h2>
                    <p class="text-dark-50 mb-5">Please enter your details to register!</p>

                    <div class="form-floating mb-3">
                        <input type="text" class="form-control" name="fname" placeholder="Prem
Kashyap">
                        <label for="floatingInput">Name</label>
                    </div>

                    <div class="form-floating mb-3">
                        <input type="email" class="form-control" name="femail"
placeholder="iamprem@gmail.com">
                        <label for="floatingInput">Email address</label>

```

</div>

```
<div class="form-floating mb-3">
    <input type="tel" class="form-control" name="mobile"
placeholder="9979787978">
    <label for="floatingInput">Mobile No</label>
</div>
```

```
<div class="form-floating mb-3">
    <input type="password" class="form-control" name="password"
placeholder="helloworld">
    <label for="floatingPassword">Password</label>
</div>
```

<p class="link-danger">{{ msg }}</p>

<button class="w-100 btn btn-lg btn-primary" type="submit">Register</button>

<hr class="my-4">

<p class="small fw-bold mt-2 pt-1 mb-0">Already registered? <a href="signin"
class="link-danger">Login</a></p>

</div>

</form>

```
<div class="col-8">
    
</div>
```

</div>

</div>

</div>

</div>

</div>

</main>

<script>

```

(function()
{
    'use strict';
    window.addEventListener('load', function()
    {
        var forms = document.getElementsByClassName('needs-validation');
        var validation = Array.prototype.filter.call(forms, function(form)
        {
            form.addEventListener('submit', function(event)
            {
                if (form.checkValidity() === False)
                {
                    event.preventDefault();
                    event.stopPropagation();
                }
                form.classList.add('was-validated');
            }, false);
        });
    }, false);
});

</script>

</body>
</html>

```

### **Signin.html:**

```

<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="">
    <title>Login</title>

```

```

        <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.13.0/css/all.min.css">
    <link rel="canonical" href="https://getbootstrap.com/docs/5.2/examples/heroes/">
    <link href="static\css\assets\dist\css\bootstrap.min.css" rel="stylesheet">
    <script>
        window.watsonAssistantChatOptions = {
            integrationID: "e15973c3-bf08-44d2-b1a3-fcaef746b52c", // The ID of this integration.
            region: "au-syd", // The region your integration is hosted in.
            serviceInstanceID: "561acd5a-e866-4ae1-8b04-999c9b3d1b84", // The ID of your
service instance.
            onLoad: function(instance) { instance.render(); }
        };
        setTimeout(function(){
            const t=document.createElement('script');
                t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" +
(window.watsonAssistantChatOptions.clientVersion || 'latest') +
"/WatsonAssistantChatEntry.js";
            document.head.appendChild(t);
        });
    </script>
</head>
<body>
    <nav class="navbar navbar-expand-md navbar-dark bg-dark shadow p-3 mb-5">
        <div class="container-fluid">
            <div class="mx-auto order-0">
                <a class="navbar-brand mx-auto" href="/">Personal Expense Tracker
Application</a>
                <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target=".dual-collapse2">
                    <span class="navbar-toggler-icon"></span>
                </button>
            </div>
            <div class="navbar-collapse collapse w-100 order-3 dual-collapse2">
                <ul class="navbar-nav ms-auto">
                    <li class="nav-item active">
                        <a class="nav-link" href="/">Home</a>

```

```

        </li>
        <li class="nav-item">
            <a class="nav-link" href="signup">Register</a>
        </li>
    </ul>
</div>
</div>
</nav>
</div>
<main>
    <div class="container px-lg-5 ">
        <div class="row align-items-center">
            <div class="col-sm-8">
                
            </div>
            <div class="col-sm-4 bg- glass">
                <form class="shadow p-3 mb-5 bg-white rounded p-4 p-md-5 border rounded-3
bg-light needs-validation" novalidate action = "{ { url_for('signin') } }" method = "POST">
                    <h2 class="fw-bold mb-2 text-uppercase">Login</h2>
                    <p class="text-dark-50 mb-5">Please enter your email and password!</p>
                    <div class="form-floating mb-3">
                        <input type="email" class="form-control" name="femail" id="floatingInput"
placeholder="name@example.com" required>
                        <label for="floatingInput">Email address</label>
                    </div>
                    <div class="form-floating mb-3">
                        <input type="password" class="form-control" name="password"
id="floatingPassword" placeholder="Password" required>
                        <label for="floatingPassword">Password</label>
                    </div>
                    <button class="w-100 btn btn-lg btn-primary" type="submit">Login</button><a
href="#"!>Forgot password?</a>
                    <hr class="my-4">

```

```
                <p class="small fw-bold mt-2 pt-1 mb-0">Don't have an account? <a
href="signup" class="link-danger">Register</a></p>
```

```
            </div>
```

```
        </form>
```

```
    </div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</main>
```

```
</body>
```

```
</html>
```

### **Dashboard.html:**

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
    <head>
```

```
        <meta charset="utf-8" />
```

```
        <meta http-equiv="X-UA-Compatible" content="IE=edge" />
```

```
        <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no"
```

```
/>
```

```
    <title>Dashboard</title>
```

```
        <link href="https://cdn.jsdelivr.net/npm/simple-datatables@latest/dist/style.css"
rel="stylesheet" />
```

```
        <link href="static\css\styles.css" rel="stylesheet" />
```

```
            <script src="https://use.fontawesome.com/releases/v6.1.0/js/all.js"
crossorigin="anonymous"></script>
```

```
    <script>
```

```
        window.watsonAssistantChatOptions = {
```

```
            integrationID: "e15973c3-bf08-44d2-b1a3-fcaef746b52c", // The ID of this integration.
```

```
            region: "au-syd", // The region your integration is hosted in.
```

```
            serviceInstanceID: "561acd5a-e866-4ae1-8b04-999c9b3d1b84", // The ID of your
service instance.
```

```
            onLoad: function(instance) { instance.render(); }
```

```
        };
```

```
        setTimeout(function(){
```



```

        const t=document.createElement('script');
            t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" +
(window.watsonAssistantChatOptions.clientVersion || 'latest') +
"/WatsonAssistantChatEntry.js";
        document.head.appendChild(t);
    });
</script>
</head>
<body class="sb-nav-fixed">
    <nav class="sb-topnav navbar navbar-expand navbar-dark bg-dark">
        <!-- Navbar Brand-->
        <a class="navbar-brand ps-2" href="dashboard">Personal Expense Tracker</a>
        <!-- Sidebar Toggle-->
            <button class="btn btn-link btn-sm order-1 order-lg-0 me-4 me-lg-0"
id="sidebarToggle" href="#"><i class="fas fa-bars"></i></button>
        <!-- Navbar Search-->
        <form class="d-none d-md-inline-block form-inline ms-auto me-0 me-md-3 my-2 my-
md-0">
            </form>
        <!-- Navbar-->
        <ul class="navbar-nav ms-auto ms-md-0 me-3 me-lg-4">
            <li class="nav-item dropdown">
                <a class="nav-link dropdown-toggle" id="navbarDropdown" href="#"
role="button" data-bs-toggle="dropdown" aria-expanded="false"><i class="fas fa-user fa-
fw"></i></a>
                    <ul class="dropdown-menu dropdown-menu-end" aria-
labelledby="navbarDropdown">
                        <li><a class="dropdown-item" href="profile">Profile</a></li>
                        <li><hr class="dropdown-divider" /></li>
                        <li><a class="dropdown-item" href="signin">Logout</a></li>
                    </ul>
                </li>
            </ul>
        </nav>
        <div id="layoutSidenav">
            <div id="layoutSidenav_nav">

```

```

<nav class="sb-sidenav accordion sb-sidenav-dark" id="sidenavAccordion">
  <div class="sb-sidenav-menu">
    <div class="nav">
      <div class="sb-sidenav-menu-heading">Application</div>
      <a class="nav-link" href="dashboard">
        <div class="sb-nav-link-icon"><i class="fas fa-tachometer-alt"></i></div>
        Dashboard
      </a>
      <div class="sb-sidenav-menu-heading">Services</div>
      <a class="nav-link collapsed" href="#" data-bs-toggle="collapse" data-bs-
target="#collapseLayouts" aria-expanded="false" aria-controls="collapseLayouts">
        <div class="sb-nav-link-icon"><i class="fas fa-columns"></i></div>
        Add Expenses
        <div class="sb-sidenav-collapse-arrow"><i class="fas fa-angle-
down"></i></div>
      </a>
      <div class="collapse" id="collapseLayouts" aria-labelledby="headingOne"
data-bs-parent="#sidenavAccordion">
        <nav class="sb-sidenav-menu-nested nav">
          <a class="nav-link" href="add_manually">Add Manually</a>
          <a class="nav-link" href="add_csv">Add using CSV</a>
        </nav>
      </div>

      <a class="nav-link collapsed" href="#" data-bs-toggle="collapse" data-bs-
target="#collapsePages" aria-expanded="false" aria-controls="collapsePages">
        <div class="sb-nav-link-icon"><i class="fas fa-book-open"></i></div>
        Budget Management
        <div class="sb-sidenav-collapse-arrow"><i class="fas fa-angle-
down"></i></div>
      </a>
      <div class="collapse" id="collapsePages" aria-labelledby="headingTwo"
data-bs-parent="#sidenavAccordion">
        <nav class="sb-sidenav-menu-nested nav">
          <a class="nav-link" href="set_limit">Setting Expense Limits</a>
        </nav>

```

</div>

<div class="sb-sidenav-menu-heading">Addons</div>

<a class="nav-link" href="monthlycharts">

<div class="sb-nav-link-icon"><i class="fas fa-chart-area"></i></div>

Monthly Charts

</a>

<a class="nav-link" href="yearlycharts">

<div class="sb-nav-link-icon"><i class="fas fa-chart-area"></i></div>

Yearly Charts

</a>

<a class="nav-link" href="table">

<div class="sb-nav-link-icon"><i class="fas fa-table"></i></div>

Tables

</a>

</div>

</div>

<div class="sb-sidenav-footer">

<div class="small">Logged in as:</div>

{{email}}

</div>

</nav>

</div>

<div id="layoutSidenav\_content">

<main>

<div class="container-fluid px-4">

<h1 class="mt-4">Dashboard</h1>

<ol class="breadcrumb mb-4">

<li class="breadcrumb-item active">Personal Expense Tracker

Application</li>

</ol>

<div class="row">

<div class="col-6">

<div class="card bg-primary text-white mb-4">

<div class="card-body">Monthly Details</div>

```
<div class="card-footer d-flex align-items-center justify-content-
between">
    <a class="small text-white stretched-link"
href="monthlycharts">View Details</a>
    <div class="small text-white"><i class="fas fa-angle-
right"></i></div>
    </div>
</div>
</div>
<div class="col-6">
    <div class="card bg-primary text-white mb-4">
        <div class="card-body">Yearly Details</div>
        <div class="card-footer d-flex align-items-center justify-content-
between">
            <a class="small text-white stretched-link"
href="yearlycharts">View Details</a>
            <div class="small text-white"><i class="fas fa-angle-
right"></i></div>
        </div>
    </div>
</div>
</div>
</div>
<div>
    <p class="link-danger">{{ msg }}</p>
    <div class="row">
        <div class="col-6">
            <div class="card bg-dark text-white mb-4">
                <div class="card-body">Monthly Limit: {{ monthly_l }}<br>Monthly
Expenses: {{ monthly_a }}</div>
                <div class="card-footer d-flex align-items-center justify-content-
between">
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
<div class="col-6">
```

```

        <div class="card bg-dark text-white mb-4">
            <div class="card-body">Yearly Limit: {{yearly_l}}<br>Yearly
Expenses: {{yearly_a}}</div>
            <div class="card-footer d-flex align-items-center justify-content-
between">

        </div>
    </div>
</div>
<div class="row justify-content-around">
    <div class="col-lg-6">
        <div class="card mb-4">
            <div class="card-header">
                <i class="fas fa-chart-pie me-1"></i>
                Pie Chart - Current Month Expenses.
            </div>
            <div class="card-body"><canvas id="myPieChart" width="60%"
height="30%"></canvas></div>
            <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
            <script>
                const pie = document.getElementById('myPieChart');
                let current_month = JSON.parse({{ current_month | tojson }})
                let month_amount = JSON.parse({{ month_amount | tojson }})
                var myPieChart = new Chart(pie, {
                    type: 'pie',
                    data: {
                        labels: current_month,
                        datasets: [{
                            data: month_amount,
                            backgroundColor: ['#007eef', '#dee545', '#ffcee7',
'#28aee5', '#ff7eef', '#dcff45', '#ffffc7', '#28a745', '#ff7bff', '#dc35aa', '#ffcaa7', '#2aa745'],
                        }],
                    },
                });

```

```

</script>

</div>
</div>
<div class="col-lg-6">
  <div class="card mb-4">
    <div class="card-header">
      <i class="fas fa-chart-pie me-1"></i>
      Pie Chart - Current Year Expenses.
    </div>
    <div class="card-body"><canvas id="myPieChartyear"
width="100%" height="50"></canvas></div>
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
    <script>
      const pieyear = document.getElementById('myPieChartyear');
      let current_year = JSON.parse({ { current_year | tojson } })
      let year_amount = JSON.parse({ { year_amount | tojson } })
      var myPieChart = new Chart(pieyear, {
        type: 'pie',
        data: {
          labels: current_year,
          datasets: [{
            data: year_amount,
            backgroundColor: ['#007eef', '#dee545', '#ffcee7',
'#28aee5', '#ff7eef', '#dcff45', '#ffffc7', '#28a745', '#ff7bff', '#dc35aa', '#ffcaa7', '#2aa745'],
          }],
        },
      });
    </script>

  </div>
</div>
</div>

```

```

        <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
crossorigin="anonymous"></script>
        <script src="static\js\scripts.js"></script>
                <script src="https://cdn.jsdelivr.net/npm/chart.js/2.8.0/Chart.min.js"
crossorigin="anonymous"></script>
                        <script src="https://cdn.jsdelivr.net/npm/simple-datatables@latest"
crossorigin="anonymous"></script>
        </body>
</html>

```

### **add\_manually.html:**

```

<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8" />
        <meta http-equiv="X-UA-Compatible" content="IE=edge" />
        <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no"
/>
        <title>Dashboard</title>
        <link href="https://cdn.jsdelivr.net/npm/simple-datatables@latest/dist/style.css"
rel="stylesheet" />
        <link href="static\css\styles.css" rel="stylesheet" />
                <script src="https://use.fontawesome.com/releases/v6.1.0/js/all.js"
crossorigin="anonymous"></script>
        <script>
            window.watsonAssistantChatOptions = {
                integrationID: "e15973c3-bf08-44d2-b1a3-fcaef746b52c", // The ID of this integration.
                region: "au-syd", // The region your integration is hosted in.
                serviceInstanceID: "561acd5a-e866-4ae1-8b04-999c9b3d1b84", // The ID of your
service instance.
                onLoad: function(instance) { instance.render(); }
            };

```

```

        setTimeout(function(){
            const t=document.createElement('script');
                t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" +
(window.watsonAssistantChatOptions.clientVersion || 'latest') +
"/WatsonAssistantChatEntry.js";
            document.head.appendChild(t);
        });
    </script>
</head>
<body class="sb-nav-fixed">
    <nav class="sb-topnav navbar navbar-expand navbar-dark bg-dark">
        <!-- Navbar Brand-->
        <a class="navbar-brand ps-2" href="dashboard">Personal Expense Tracker</a>
        <!-- Sidebar Toggle-->
            <button class="btn btn-link btn-sm order-1 order-lg-0 me-4 me-lg-0"
id="sidebarToggle" href="#"><i class="fas fa-bars"></i></button>
        <!-- Navbar Search-->
        <form class="d-none d-md-inline-block form-inline ms-auto me-0 me-md-3 my-2 my-
md-0">
            </form>
        <!-- Navbar-->
        <ul class="navbar-nav ms-auto ms-md-0 me-3 me-lg-4">
            <li class="nav-item dropdown">
                <a class="nav-link dropdown-toggle" id="navbarDropdown" href="#"
role="button" data-bs-toggle="dropdown" aria-expanded="false"><i class="fas fa-user fa-
fw"></i></a>
                    <ul class="dropdown-menu dropdown-menu-end" aria-
labelledby="navbarDropdown">
                        <li><a class="dropdown-item" href="profile">Profile</a></li>
                        <li><hr class="dropdown-divider" /></li>
                        <li><a class="dropdown-item" href="signin">Logout</a></li>
                    </ul>
                </li>
            </ul>
        </div id="layoutSidenav">

```



```

<div id="layoutSidenav_nav">
  <nav class="sb-sidenav accordion sb-sidenav-dark" id="sidenavAccordion">
    <div class="sb-sidenav-menu">
      <div class="nav">
        <div class="sb-sidenav-menu-heading">Application</div>
        <a class="nav-link" href="dashboard">
          <div class="sb-nav-link-icon"><i class="fas fa-tachometer-alt"></i></div>
          Dashboard
        </a>
        <div class="sb-sidenav-menu-heading">Services</div>
        <a class="nav-link collapsed" href="#" data-bs-toggle="collapse" data-bs-
target="#collapseLayouts" aria-expanded="false" aria-controls="collapseLayouts">
          <div class="sb-nav-link-icon"><i class="fas fa-columns"></i></div>
          Add Expenses
          <div class="sb-sidenav-collapse-arrow"><i class="fas fa-angle-
down"></i></div>
        </a>
        <div class="collapse" id="collapseLayouts" aria-labelledby="headingOne"
data-bs-parent="#sidenavAccordion">
          <nav class="sb-sidenav-menu-nested nav">
            <a class="nav-link" href="add_manually">Add Manually</a>
            <a class="nav-link" href="add_csv">Add using CSV</a>
          </nav>
        </div>

        <a class="nav-link collapsed" href="#" data-bs-toggle="collapse" data-bs-
target="#collapsePages" aria-expanded="false" aria-controls="collapsePages">
          <div class="sb-nav-link-icon"><i class="fas fa-book-open"></i></div>
          Budget Management
          <div class="sb-sidenav-collapse-arrow"><i class="fas fa-angle-
down"></i></div>
        </a>
        <div class="collapse" id="collapsePages" aria-labelledby="headingTwo"
data-bs-parent="#sidenavAccordion">
          <nav class="sb-sidenav-menu-nested nav">
            <a class="nav-link" href="set_limit">Setting Expense Limits</a>

```

</nav>

</div>

<div class="sb-sidenav-menu-heading">Addons</div>

<a class="nav-link" href="monthlycharts">

<div class="sb-nav-link-icon"><i class="fas fa-chart-area"></i></div>

Monthly Charts

</a>

<a class="nav-link" href="yearlycharts">

<div class="sb-nav-link-icon"><i class="fas fa-chart-area"></i></div>

Yearly Charts

</a>

<a class="nav-link" href="table">

<div class="sb-nav-link-icon"><i class="fas fa-table"></i></div>

Tables

</a>

</div>

</div>

<div class="sb-sidenav-footer">

<div class="small">Logged in as:</div>

{{email}}

</div>

</nav>

</div>

<div id="layoutSidenav\_content">

<main>

<div class="container-fluid px-4">

<h1 class="mt-4">Add Expenses Manually</h1>

<div class="container ">

<div class="row align-items-center">

<div class="col-sm-8">



</div>

```

<div class="col-sm-4 bg- glass">
    <form name= "add" class="shadow p-3 mb-5 bg-white rounded p-4 p-md-5
border rounded-3 bg-light needs-validation" novalidate action = "{{ url_for('add_manually')
}}" method = "POST">
        <h2 class="fw-bold mb-2 text-uppercase">Add Expenses</h2>
        <p class="text-dark-50 mb-5">Please add your expense according to
inputs</p>
        <div class="form-floating mb-3">
            <input type="text" class="form-control" name="amount" id="amount"
placeholder="1234" required>
            <label for="amount">Amount</label>

        </div>
        <div class="form mb-3">
            <select class="form-select" class="form-control" name="category"
id="category" required>
                <option selected>Select a Category</option>
                <option value="Food">Food</option>
                <option value="Shopping">Shopping</option>
                <option value="Transport">Transport</option>
                <option value="Bills and Fees">Bills and Fees</option>
                <option value="Entertainment">Entertainment</option>
                <option value="Travel">Travel</option>
                <option value="Healthcare">Healthcare</option>
                <option value="Household">Household</option>
                <option value="Others">Others</option>
            </select>
        </div>

        <div class="input-group date" data-provide="datepicker">
            <input type="date" name="date" class="form-control" required>
            <div class="input-group-addon">
                <span class="glyphicon glyphicon-th"></span>
            </div>
        </div>
    </div>

```

```

        <br>
        <p class="link-danger">{{ msg1 }}</p>
        <p class="link-safe">{{ msg2 }}</p><br>

        <button class="w-100 btn btn-lg btn-primary" href="#" type="submit">Add
Expenses</button>
    </div>
</form>
</div>
</div>
</div>
</div>
</div>

</main>
</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
crossorigin="anonymous"></script>
<script src="static\js\scripts.js"></script>
<script src="https://cdn.jsdelivr.net/npm/chart.js/2.8.0/Chart.min.js"
crossorigin="anonymous"></script>
<script src="static\css\assets\demo\chart-area-demo.js"></script>
<script src="static\css\assets\demo\chart-bar-demo.js"></script>
<script src="https://cdn.jsdelivr.net/npm/simple-datatables@latest"
crossorigin="anonymous"></script>
<script src="static\js\datatables-simple-demo.js"></script>
</body>
</html>

```

### **Add\_csv.html:**

```

<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8" />
        <meta http-equiv="X-UA-Compatible" content="IE=edge" />
        <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no"
/>

```

```

<title>Dashboard</title>

    <link href="https://cdn.jsdelivr.net/npm/simple-datatables@latest/dist/style.css"
rel="stylesheet" />
    <link href="static\css\styles.css" rel="stylesheet" />
    <script src="https://use.fontawesome.com/releases/v6.1.0/js/all.js"
crossorigin="anonymous"></script>
    <script>
        window.watsonAssistantChatOptions = {
            integrationID: "e15973c3-bf08-44d2-b1a3-fcaef746b52c", // The ID of this integration.
            region: "au-syd", // The region your integration is hosted in.
            serviceInstanceID: "561acd5a-e866-4ae1-8b04-999c9b3d1b84", // The ID of your
service instance.

            onLoad: function(instance) { instance.render(); }
        };
        setTimeout(function(){
            const t=document.createElement('script');
                t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" +
(window.watsonAssistantChatOptions.clientVersion || 'latest') +
"/WatsonAssistantChatEntry.js";
            document.head.appendChild(t);
        });
    </script>
</head>
<body class="sb-nav-fixed">
    <nav class="sb-topnav navbar navbar-expand navbar-dark bg-dark">
        <!-- Navbar Brand-->
        <a class="navbar-brand ps-2" href="dashboard">Personal Expense Tracker</a>
        <!-- Sidebar Toggle-->
        <button class="btn btn-link btn-sm order-1 order-lg-0 me-4 me-lg-0"
id="sidebarToggle" href="#"><i class="fas fa-bars"></i></button>
        <!-- Navbar Search-->
        <form class="d-none d-md-inline-block form-inline ms-auto me-0 me-md-3 my-2 my-
md-0">
            </form>
        <!-- Navbar-->
        <ul class="navbar-nav ms-auto ms-md-0 me-3 me-lg-4">

```

```

<li class="nav-item dropdown">
    <a class="nav-link dropdown-toggle" id="navbarDropdown" href="#"
role="button" data-bs-toggle="dropdown" aria-expanded="false"><i class="fas fa-user fa-
fw"></i></a>

    <ul class="dropdown-menu dropdown-menu-end" aria-
labelledby="navbarDropdown">
        <li><a class="dropdown-item" href="profile">Profile</a></li>
        <li><hr class="dropdown-divider" /></li>
        <li><a class="dropdown-item" href="signin">Logout</a></li>
    </ul>
</li>
</ul>
</nav>
<div id="layoutSidenav">
    <div id="layoutSidenav_nav">
        <nav class="sb-sidenav accordion sb-sidenav-dark" id="sidenavAccordion">
            <div class="sb-sidenav-menu">
                <div class="nav">
                    <div class="sb-sidenav-menu-heading">Application</div>
                    <a class="nav-link" href="dashboard">
                        <div class="sb-nav-link-icon"><i class="fas fa-tachometer-alt"></i></div>
                        Dashboard
                    </a>
                    <div class="sb-sidenav-menu-heading">Services</div>
                    <a class="nav-link collapsed" href="#" data-bs-toggle="collapse" data-bs-
target="#collapseLayouts" aria-expanded="false" aria-controls="collapseLayouts">
                        <div class="sb-nav-link-icon"><i class="fas fa-columns"></i></div>
                        Add Expenses
                        <div class="sb-sidenav-collapse-arrow"><i class="fas fa-angle-
down"></i></div>
                    </a>
                    <div class="collapse" id="collapseLayouts" aria-labelledby="headingOne"
data-bs-parent="#sidenavAccordion">
                        <nav class="sb-sidenav-menu-nested nav">
                            <a class="nav-link" href="add_manually">Add Manually</a>
                            <a class="nav-link" href="add_csv">Add using CSV</a>

```

</nav>

</div>

<a class="nav-link collapsed" href="#" data-bs-toggle="collapse" data-bs-target="#collapsePages" aria-expanded="false" aria-controls="collapsePages">

<div class="sb-nav-link-icon"><i class="fas fa-book-open"></i></div>

Budget Management

<div class="sb-sidenav-collapse-arrow"><i class="fas fa-angle-down"></i></div>

</a>

<div class="collapse" id="collapsePages" aria-labelledby="headingTwo" data-bs-parent="#sidenavAccordion">

<nav class="sb-sidenav-menu-nested nav">

<a class="nav-link" href="set\_limit">Setting Expense Limits</a>

</nav>

</div>

<div class="sb-sidenav-menu-heading">Addons</div>

<a class="nav-link" href="monthlycharts">

<div class="sb-nav-link-icon"><i class="fas fa-chart-area"></i></div>

Monthly Charts

</a>

<a class="nav-link" href="yearlycharts">

<div class="sb-nav-link-icon"><i class="fas fa-chart-area"></i></div>

Yearly Charts

</a>

<a class="nav-link" href="table">

<div class="sb-nav-link-icon"><i class="fas fa-table"></i></div>

Tables

</a>

</div>

</div>

<div class="sb-sidenav-footer">

<div class="small">Logged in as:</div>

{{email}}

```

        </div>
    </nav>
</div>
<div id="layoutSidenav_content" class="bg w-100">
    <p></p>
    <main>
        <div class="container-fluid px-4">
            <h1 class="mt-4">Add Expenses using CSV file</h1>

            <div class="row align-items-center">
                <div class="col-sm-6">
                    
                </div>
                <div class="col-sm-4 bg- glass">
                    <form class="shadow p-3 mb-5 bg-white rounded p-4 p-md-5 border rounded-3 bg-
light needs-validation" novalidate action = "{ { url_for('add_csv') } }" method = "POST">
                        <h2 class="fw-bold mb-2 text-uppercase">Add Expenses using CSV files</h2>
                        <p class="text-dark-50 mb-5">Please upload csv file with proper table names as
shown<br> 1. Amount column.<br> 2. Category column.<br> 3. Date of Bill column
Format (YYYY-MM-DD).</p>
                        <div class="form mb-3">
                            <p>Upload CSV File</p>
                            <input type="file" class="form-control" name="csv_file" id="floatingInput"
required>
                        </div>
                        <p class="link-safe">{ { msg } }</p>
                        <p class="link-danger">{ { msg2 } }</p>
                        <button class="w-100 btn btn-lg btn-primary" type="submit">Upload</button>
                    </div>
                </form>
            </div></div>
</div>

```



```

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
crossorigin="anonymous"></script>
<script src="static\js\scripts.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.8.0/Chart.min.js"
crossorigin="anonymous"></script>
<script src="static\css\assets\demo\chart-area-demo.js"></script>
<script src="static\css\assets\demo\chart-bar-demo.js"></script>
<script src="https://cdn.jsdelivr.net/npm/simple-datatables@latest"
crossorigin="anonymous"></script>
<script src="static\js\datatables-simple-demo.js"></script>

</body>
</html>

```

### **set\_limit.html:**

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8" />
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no"
/>
<title>Dashboard</title>
<link href="https://cdn.jsdelivr.net/npm/simple-datatables@latest/dist/style.css"
rel="stylesheet" />
<link href="static\css\styles.css" rel="stylesheet" />
<script src="https://use.fontawesome.com/releases/v6.1.0/js/all.js"
crossorigin="anonymous"></script>
<script>
window.watsonAssistantChatOptions = {
integrationID: "e15973c3-bf08-44d2-b1a3-fcaef746b52c", // The ID of this integration.
region: "au-syd", // The region your integration is hosted in.
serviceInstanceID: "561acd5a-e866-4ae1-8b04-999c9b3d1b84", // The ID of your
service instance.
onLoad: function(instance) { instance.render(); }

```

```

    };
    setTimeout(function(){
        const t=document.createElement('script');
            t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" +
(window.watsonAssistantChatOptions.clientVersion || 'latest') +
"/WatsonAssistantChatEntry.js";
        document.head.appendChild(t);
    });
</script>
</head>
<body class="sb-nav-fixed">
    <nav class="sb-topnav navbar navbar-expand navbar-dark bg-dark">
        <!-- Navbar Brand-->
        <a class="navbar-brand ps-2" href="dashboard">Personal Expense Tracker</a>
        <!-- Sidebar Toggle-->
            <button class="btn btn-link btn-sm order-1 order-lg-0 me-4 me-lg-0"
id="sidebarToggle" href="#!"><i class="fas fa-bars"></i></button>
        <!-- Navbar Search-->
        <form class="d-none d-md-inline-block form-inline ms-auto me-0 me-md-3 my-2 my-
md-0">
            </form>
        <!-- Navbar-->
        <ul class="navbar-nav ms-auto ms-md-0 me-3 me-lg-4">
            <li class="nav-item dropdown">
                <a class="nav-link dropdown-toggle" id="navbarDropdown" href="#"
role="button" data-bs-toggle="dropdown" aria-expanded="false"><i class="fas fa-user fa-
fw"></i></a>
                    <ul class="dropdown-menu dropdown-menu-end" aria-
labelledby="navbarDropdown">
                        <li><a class="dropdown-item" href="profile">Profile</a></li>
                        <li><hr class="dropdown-divider" /></li>
                        <li><a class="dropdown-item" href="signin">Logout</a></li>
                    </ul>
                </li>
            </ul>
        </nav>

```

```

<div id="layoutSidenav">
  <div id="layoutSidenav_nav">
    <nav class="sb-sidenav accordion sb-sidenav-dark" id="sidenavAccordion">
      <div class="sb-sidenav-menu">
        <div class="nav">
          <div class="sb-sidenav-menu-heading">Application</div>
          <a class="nav-link" href="dashboard">
            <div class="sb-nav-link-icon"><i class="fas fa-tachometer-alt"></i></div>
            Dashboard
          </a>
          <div class="sb-sidenav-menu-heading">Services</div>
          <a class="nav-link collapsed" href="#" data-bs-toggle="collapse" data-bs-
target="#collapseLayouts" aria-expanded="false" aria-controls="collapseLayouts">
            <div class="sb-nav-link-icon"><i class="fas fa-columns"></i></div>
            Add Expenses
            <div class="sb-sidenav-collapse-arrow"><i class="fas fa-angle-
down"></i></div>
          </a>
          <div class="collapse" id="collapseLayouts" aria-labelledby="headingOne"
data-bs-parent="#sidenavAccordion">
            <nav class="sb-sidenav-menu-nested nav">
              <a class="nav-link" href="add_manually">Add Manually</a>
              <a class="nav-link" href="add_csv">Add using CSV</a>
            </nav>
          </div>

          <a class="nav-link collapsed" href="#" data-bs-toggle="collapse" data-bs-
target="#collapsePages" aria-expanded="false" aria-controls="collapsePages">
            <div class="sb-nav-link-icon"><i class="fas fa-book-open"></i></div>
            Budget Management
            <div class="sb-sidenav-collapse-arrow"><i class="fas fa-angle-
down"></i></div>
          </a>
          <div class="collapse" id="collapsePages" aria-labelledby="headingTwo"
data-bs-parent="#sidenavAccordion">
            <nav class="sb-sidenav-menu-nested nav">

```

```

        <a class="nav-link" href="set_limit">Setting Expense Limits</a>
    </nav>
</div>

<div class="sb-sidenav-menu-heading">Addons</div>
<a class="nav-link" href="monthlycharts">
    <div class="sb-nav-link-icon"><i class="fas fa-chart-area"></i></div>
    Monthly Charts
</a>
<a class="nav-link" href="yearlycharts">
    <div class="sb-nav-link-icon"><i class="fas fa-chart-area"></i></div>
    Yearly Charts
</a>
<a class="nav-link" href="table">
    <div class="sb-nav-link-icon"><i class="fas fa-table"></i></div>
    Tables
</a>
</div>
</div>
<div class="sb-sidenav-footer">
    <div class="small">Logged in as:</div>
    {{email}}
</div>
</nav>
</div>
<div id="layoutSidenav_content">

<main>
    <div class="container-fluid px-4">
        <h1 class="mt-4">Setting limit for Expenses</h1>
    <div class="container px-lg-5 ">
        <div class="row align-items-center">
            <div class="col-sm-8">
                

```

```

</div>
<div class="col-sm-4 bg- glass">
    <form name= "add" class="shadow p-3 mb-5 bg-white rounded p-4 p-md-5
border rounded-3 bg-light needs-validation" novalidate action = "{{ url_for('set_limit') }}"
method = "POST">
        <h2 class="fw-bold mb-2 text-uppercase">Set Limit for Expenses</h2>
        <p class="text-dark-50 mb-5">Add limits for your expenses</p>

        <div class="form-floating mb-3">
            <input type="text" class="form-control" name="monthly" id="monthly"
placeholder="1234" required>
            <label for="monthly">Monthly Limit</label>
        </div>

        <div class="form-floating mb-3">
            <input type="text" class="form-control" name="yearly" id="yearly"
placeholder="1234" required>
            <label for="yearly">Yearly Limit</label>
        </div>

        <p class="link-danger">{{ msg1 }}</p>
        <p class="link-safe">{{ msg2 }}</p><br>

        <button class="w-100 btn btn-lg btn-primary" href="#" type="submit">Add
Expenses</button>
    </div>
</form>
</div>
</div>
</div>
</main>
</div>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
crossorigin="anonymous"></script>

```

```

<script src="static\js\scripts.js"></script>
        <script    src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.8.0/Chart.min.js"
crossorigin="anonymous"></script>
        <script src="static\css\assets\demo\chart-area-demo.js"></script>
        <script src="static\css\assets\demo\chart-bar-demo.js"></script>
                <script    src="https://cdn.jsdelivr.net/npm/simple-datatables@latest"
crossorigin="anonymous"></script>
        <script src="static\js\datatables-simple-demo.js"></script>
</body>
</html>

```

### Monthlycharts.html:

```

<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8" />
        <meta http-equiv="X-UA-Compatible" content="IE=edge" />
        <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no"
/>
        <title>Dashboard</title>
        <link    href="https://cdn.jsdelivr.net/npm/simple-datatables@latest/dist/style.css"
rel="stylesheet" />
        <link href="https://getbootstrap.com/docs/5.2/assets/css/docs.css" rel="stylesheet">
        <link rel="canonical" href="https://getbootstrap.com/docs/5.2/examples/cover/">
        <link href="static\css\charts.css" rel="stylesheet"/>
        <link href="static\css\styles.css" rel="stylesheet"/>
        <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css"
rel="stylesheet">
                <script    src="https://use.fontawesome.com/releases/v6.1.0/js/all.js"
crossorigin="anonymous"></script>
        <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
                                <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"></script>
        <script>
            window.watsonAssistantChatOptions = {
                integrationID: "e15973c3-bf08-44d2-b1a3-fcaef746b52c", // The ID of this integration.

```

```

    region: "au-syd", // The region your integration is hosted in.
    serviceInstanceID: "561acd5a-e866-4ae1-8b04-999c9b3d1b84", // The ID of your
service instance.

    onLoad: function(instance) { instance.render(); }
};
setTimeout(function(){
    const t=document.createElement('script');
        t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" +
(window.watsonAssistantChatOptions.clientVersion || 'latest') +
"/WatsonAssistantChatEntry.js";
        document.head.appendChild(t);
    });
</script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/4.0.1/chart.min.js"
integrity="sha512-
tQYZBKe34uzoeOjY9jr3MX7R/mo7n25vnqbnrkskGr4D6YOoPYSpyafUAzQVjV6xAozAqU
FIEFsCO4z8mnVBXA==" crossorigin="anonymous" referrerpolicy="no-referrer"></script>
</head>
<body class="sb-nav-fixed">
    <nav class="sb-topnav navbar navbar-expand navbar-dark bg-dark">
        <!-- Navbar Brand-->
        <a class="navbar-brand ps-2" href="dashboard">Personal Expense Tracker</a>
        <!-- Sidebar Toggle-->
        <button class="btn btn-link btn-sm order-1 order-lg-0 me-4 me-lg-0"
id="sidebarToggle" href="#"><i class="fas fa-bars"></i></button>
        <!-- Navbar Search-->
        <form class="d-none d-md-inline-block form-inline ms-auto me-0 me-md-3 my-2 my-
md-0">
        </form>
        <!-- Navbar -->
        <ul class="navbar-nav ms-auto ms-md-0 me-3 me-lg-4">
            <li class="nav-item dropdown">
                <a class="nav-link dropdown-toggle" id="navbarDropdown" href="#"
role="button" data-bs-toggle="dropdown" aria-expanded="false"><i class="fas fa-user fa-
fw"></i></a>

```

```

        <ul class="dropdown-menu dropdown-menu-end" aria-
labelledby="navbarDropdown">
            <li><a class="dropdown-item" href="profile">Profile</a></li>
            <li><hr class="dropdown-divider" /></li>
            <li><a class="dropdown-item" href="signin">Logout</a></li>
        </ul>
    </li>
</ul>
</nav>
<div id="layoutSidenav">
    <div id="layoutSidenav_nav">
        <nav class="sb-sidenav accordion sb-sidenav-dark" id="sidenavAccordion">
            <div class="sb-sidenav-menu">
                <div class="nav">
                    <div class="sb-sidenav-menu-heading">Application</div>
                    <a class="nav-link" href="dashboard">
                        <div class="sb-nav-link-icon"><i class="fas fa-tachometer-alt"></i></div>
                        Dashboard
                    </a>
                    <div class="sb-sidenav-menu-heading">Services</div>
                    <a class="nav-link collapsed" href="#" data-bs-toggle="collapse" data-bs-
target="#collapseLayouts" aria-expanded="false" aria-controls="collapseLayouts">
                        <div class="sb-nav-link-icon"><i class="fas fa-columns"></i></div>
                        Add Expenses
                        <div class="sb-sidenav-collapse-arrow"><i class="fas fa-angle-
down"></i></div>
                    </a>
                    <div class="collapse" id="collapseLayouts" aria-labelledby="headingOne"
data-bs-parent="#sidenavAccordion">
                        <nav class="sb-sidenav-menu-nested nav">
                            <a class="nav-link" href="add_manually">Add Manually</a>
                            <a class="nav-link" href="add_csv">Add using CSV</a>
                        </nav>
                    </div>

```



```
<a class="nav-link collapsed" href="#" data-bs-toggle="collapse" data-bs-target="#collapsePages" aria-expanded="false" aria-controls="collapsePages">
```

```
<div class="sb-nav-link-icon"><i class="fas fa-book-open"></i></div>
```

```
Budget Management
```

```
<div class="sb-sidenav-collapse-arrow"><i class="fas fa-angle-down"></i></div>
```

```
</a>
```

```
<div class="collapse" id="collapsePages" aria-labelledby="headingTwo" data-bs-parent="#sidenavAccordion">
```

```
<nav class="sb-sidenav-menu-nested nav">
```

```
<a class="nav-link" href="set_limit">Setting Expense Limits</a>
```

```
</nav>
```

```
</div>
```

```
<div class="sb-sidenav-menu-heading">Addons</div>
```

```
<a class="nav-link" href="monthlycharts">
```

```
<div class="sb-nav-link-icon"><i class="fas fa-chart-area"></i></div>
```

```
Monthly Charts
```

```
</a>
```

```
<a class="nav-link" href="yearlycharts">
```

```
<div class="sb-nav-link-icon"><i class="fas fa-chart-area"></i></div>
```

```
Yearly Charts
```

```
</a>
```

```
<a class="nav-link" href="table">
```

```
<div class="sb-nav-link-icon"><i class="fas fa-table"></i></div>
```

```
Tables
```

```
</a>
```

```
</div>
```

```
</div>
```

```
<div class="sb-sidenav-footer">
```

```
<div class="small">Logged in as:</div>
```

```
{{email}}
```

```
</div>
```

```
</nav>
```

```
</div>
```

```

<div id="layoutSidenav_content">
  <main>
    <div class="container-fluid px-4">
      <h1 class="mt-4">Monthly Expenses Details</h1>
      <ol class="breadcrumb mb-4">
        <li class="breadcrumb-item"><a href="dashboard">Dashboard</a></li>
        <li class="breadcrumb-item active">Monthly Expense Details</li>
      </ol>
      <div class="card mb-4">
        <div class="card-body">
          In this section the below charts shows the monthly expenses and expenses
          according to the category.
          <br>Monthly limit and monthly expeneses are also shown here.

        </div>
      </div>
      <div class="row">
        <div class="col-lg-6">
          <div class="card mb-4">
            <div class="card-header">
              <i class="fas fa-chart-bar me-1"></i>
              Bar Chart - Monthly Expenses<br>Total Expenses = { { s } }
            </div>
            <div class="card-body"><canvas id="myBarChart" width="100%"
height="50"></canvas></div>
            <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
            <script>
              const bar = document.getElementById('myBarChart');
              let barcategories= JSON.parse({ { barcategories | tojson } })
              let baramount= JSON.parse({ { baramount | tojson } })
              let length= JSON.parse({ { length | tojson } })
              var myLineChart = new Chart(bar, {
                type: 'bar',
                data: {
                  labels: barcategories,
                  datasets: [{

```

```

        label: "Monthly",
        borderColor: ['#0ffedf', '#dcef45', '#ff2107', '#284e45', '#0eeff',
'#da323e', '#eeef27', '#3ea745', '#553eaf', '#ad3a45', '#21c107', '#baa745'],
        data: baramount,
        backgroundColor: ['#0ffedf', '#dcef45', '#ff2107',
'#284e45', '#0eeff', '#da323e', '#eeef27', '#3ea745', '#553eaf', '#ad3a45', '#21c107', '#baa745'],
    }],
},
options: {
    scales: {
        xAxes: [{
            time: {
                unit: 'month'
            },
            gridLines: {
                display: false
            },
            ticks: {
                maxTicksLimit: length
            }
        }],
        yAxes: [{
            ticks: {
                min: 0,
                max: 100000,
                maxTicksLimit: 5
            },
            gridLines: {
                display: true
            }
        }],
    },
    legend: {
        display: false
    }
}

```

```

    });

</script>
</div>
</div>
<div class="col-lg-6">
  <div class="card mb-4">
    <div class="card-header">
      <i class="fas fa-chart-pie me-1"></i>
      Pie Chart -Categories Expenses
      <br>Total Expenses = { { s } }
    </div>
    <div class="card-body"><canvas id="myPieChart" width="100%"
height="50"></canvas></div>
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
    <script>
      const pie = document.getElementById('myPieChart');
      let piecategories= JSON.parse({ { piecategories | tojson } })
      let pieamount= JSON.parse({ { pieamount | tojson } })
      var myPieChart = new Chart(pie, {
        type: 'pie',
        data: {
          labels: piecategories,
          datasets: [{
            data: pieamount,
            backgroundColor: ['#0f71ef', '#111245', '#11ce17',
'#281ee5', '#ff7e2f', '#dc2f45', '#ff3fc7', '#282745', '#ff722f', '#d115aa', '#ffc1a7', '#2aa145'],
          }],
        },
      });
    </script>
  </div>
</div>
</div>

```

```

        </div>
    </main>
</body>
</html>

```

### Yearly\_charts.html:

```

<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8" />
        <meta http-equiv="X-UA-Compatible" content="IE=edge" />
        <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no"
    />
        <title>Dashboard</title>
        <link href="static\css\styles.css" rel="stylesheet" />
        <link href="static\css\charts.css" rel="stylesheet"/>
        <link href="https://cdn.jsdelivr.net/npm/simple-datatables@latest/dist/style.css"
rel="stylesheet" />
        <link href="https://getbootstrap.com/docs/5.2/assets/css/docs.css" rel="stylesheet">
        <link rel="canonical" href="https://getbootstrap.com/docs/5.2/examples/cover/">
        <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css"
rel="stylesheet">
        <script src="https://use.fontawesome.com/releases/v6.1.0/js/all.js"
crossorigin="anonymous"></script>
        <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
        <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"></script>
        <script>
            window.watsonAssistantChatOptions = {
                integrationID: "e15973c3-bf08-44d2-b1a3-fcaef746b52c", // The ID of this integration.
                region: "au-syd", // The region your integration is hosted in.
                serviceInstanceID: "561acd5a-e866-4ae1-8b04-999c9b3d1b84", // The ID of your
service instance.
                onLoad: function(instance) { instance.render(); }
            };
            setTimeout(function(){

```

```

const t=document.createElement('script');
        t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" +
(window.watsonAssistantChatOptions.clientVersion || 'latest') +
"/WatsonAssistantChatEntry.js";
        document.head.appendChild(t);
    });
</script>
        <script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/4.0.1/chart.min.js"
integrity="sha512-
tQYZBKe34uzoeOjY9jr3MX7R/mo7n25vnqbnrkskGr4D6YOoPYSpyafUAzQVjV6xAozAqU
FIEFsCO4z8mnVBXA==" crossorigin="anonymous" referrerpolicy="no-referrer"></script>
    </head>
    <body class="sb-nav-fixed">
        <nav class="sb-topnav navbar navbar-expand navbar-dark bg-dark">
            <!-- Navbar Brand-->
            <a class="navbar-brand ps-2" href="dashboard">Personal Expense Tracker</a>
            <!-- Sidebar Toggle-->
            <button class="btn btn-link btn-sm order-1 order-lg-0 me-4 me-lg-0"
id="sidebarToggle" href="#"><i class="fas fa-bars"></i></button>
            <!-- Navbar Search-->
            <form class="d-none d-md-inline-block form-inline ms-auto me-0 me-md-3 my-2 my-
md-0">
            </form>
            <!-- Navbar-->
            <ul class="navbar-nav ms-auto ms-md-0 me-3 me-lg-4">
                <li class="nav-item dropdown">
                    <a class="nav-link dropdown-toggle" id="navbarDropdown" href="#"
role="button" data-bs-toggle="dropdown" aria-expanded="false"><i class="fas fa-user fa-
fw"></i></a>
                    <ul class="dropdown-menu dropdown-menu-end" aria-
labelledby="navbarDropdown">
                        <li><a class="dropdown-item" href="profile">Profile</a></li>
                        <li><hr class="dropdown-divider" /></li>
                        <li><a class="dropdown-item" href="signin">Logout</a></li>
                    </ul>
                </li>

```

```

        </ul>
    </nav>
    <div id="layoutSidenav">
        <div id="layoutSidenav_nav">
            <nav class="sb-sidenav accordion sb-sidenav-dark" id="sidenavAccordion">
                <div class="sb-sidenav-menu">
                    <div class="nav">
                        <div class="sb-sidenav-menu-heading">Application</div>
                        <a class="nav-link" href="dashboard">
                            <div class="sb-nav-link-icon"><i class="fas fa-tachometer-alt"></i></div>
                            Dashboard
                        </a>
                        <div class="sb-sidenav-menu-heading">Services</div>
                        <a class="nav-link collapsed" href="#" data-bs-toggle="collapse" data-bs-
target="#collapseLayouts" aria-expanded="false" aria-controls="collapseLayouts">
                            <div class="sb-nav-link-icon"><i class="fas fa-columns"></i></div>
                            Add Expenses
                            <div class="sb-sidenav-collapse-arrow"><i class="fas fa-angle-
down"></i></div>
                        </a>
                        <div class="collapse" id="collapseLayouts" aria-labelledby="headingOne"
data-bs-parent="#sidenavAccordion">
                            <nav class="sb-sidenav-menu-nested nav">
                                <a class="nav-link" href="add_manually">Add Manually</a>
                                <a class="nav-link" href="add_csv">Add using CSV</a>
                            </nav>
                        </div>

                        <a class="nav-link collapsed" href="#" data-bs-toggle="collapse" data-bs-
target="#collapsePages" aria-expanded="false" aria-controls="collapsePages">
                            <div class="sb-nav-link-icon"><i class="fas fa-book-open"></i></div>
                            Budget Management
                            <div class="sb-sidenav-collapse-arrow"><i class="fas fa-angle-
down"></i></div>
                        </a>

```

```
        <div class="collapse" id="collapsePages" aria-labelledby="headingTwo"
data-bs-parent="#sidenavAccordion">
            <nav class="sb-sidenav-menu-nested nav">
                <a class="nav-link" href="set_limit">Setting Expense Limits</a>
            </nav>
        </div>
```

```
    <div class="sb-sidenav-menu-heading">Addons</div>
    <a class="nav-link" href="monthlycharts">
        <div class="sb-nav-link-icon"><i class="fas fa-chart-area"></i></div>
        Monthly Charts
    </a>
    <a class="nav-link" href="yearlycharts">
        <div class="sb-nav-link-icon"><i class="fas fa-chart-area"></i></div>
        Yearly Charts
    </a>
    <a class="nav-link" href="table">
        <div class="sb-nav-link-icon"><i class="fas fa-table"></i></div>
        Tables
    </a>
</div>
```

```
</div>
<div class="sb-sidenav-footer">
    <div class="small">Logged in as:</div>
    {{email}}
</div>
</nav>
</div>
<div id="layoutSidenav_content">
    <main>
        <div class="container-fluid px-4">
            <h1 class="mt-4">Yearly Charts</h1>
            <ol class="breadcrumb mb-4">
                <li class="breadcrumb-item"><a href="dashboard">Dashboard</a></li>
                <li class="breadcrumb-item active">Yearly Charts</li>
```



</ol>

<div class="card mb-4">

<div class="card-body">

In this section the below charts shows the Yearly expenses and expenses according to the category.

<br>Monthly limit and yearly expenses are also shown here.

</div>

</div>

<div class="row">

<div class="col-lg-6">

<div class="card mb-4">

<div class="card-header">

<i class="fas fa-chart-bar me-1"></i>

Bar Chart - Yearly Expenses<br>Total Expenses = { { s } }

</div>

<div class="card-body"><canvas id="myBarChart" width="100%" height="50"></canvas></div>

<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

<script>

const bar = document.getElementById('myBarChart');

let barcategories= JSON.parse({ { barcategories | tojson } })

let baramount= JSON.parse({ { baramount | tojson } })

let length= JSON.parse({ { length | tojson } })

var myLineChart = new Chart(bar, {

type: 'bar',

data: {

labels: barcategories,

datasets: [{

label: "Yearly",

borderColor: ['#0ffedf', '#dcef45', '#ff2107', '#284e45', '#0eeff', '#da323e', '#eeef27', '#3ea745', '#553eaf', '#ad3a45', '#21c107', '#baa745'],

data: baramount,

backgroundColor: ['#0ffedf', '#dcef45', '#ff2107', '#284e45', '#0eeff', '#da323e', '#eeef27', '#3ea745', '#553eaf', '#ad3a45', '#21c107', '#baa745'],

}},

},

```
options: {
  scales: {
    xAxes: [{
      time: {
        unit: 'Years'
      },
      gridLines: {
        display: false
      },
      ticks: {
        maxTicksLimit: length
      }
    }],
    yAxes: [{
      ticks: {
        min: 0,
        max: 100000,
        maxTicksLimit: 5
      },
      gridLines: {
        display: true
      }
    }],
  },
  legend: {
    display: false
  }
}
});
```

</script>

</div>

</div>

<div class="col-lg-6">

<div class="card mb-4">

<div class="card-header">

```

        <i class="fas fa-chart-pie me-1"></i>
        Pie Chart -Categories Expenses
        <br>Total Expenses = { { s } }
    </div>

    <div class="card-body"><canvas id="myPieChart" width="100%"
height="50"></canvas></div>

    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
    <script>
        const pie = document.getElementById('myPieChart');
        let piecategories= JSON.parse({ { piecategories | tojson } })
        let pieamount= JSON.parse({ { pieamount | tojson } })
        var myPieChart = new Chart(pie, {
            type: 'pie',
            data: {
                labels: piecategories,
                datasets: [{
                    data: pieamount,
                    backgroundColor: ['#0f71ef', '#d12245', '#11ce17',
'#281ee5','#ff7e2f', '#dc2f45', '#ff3fc7', '#282745','#ff722f', '#d115aa', '#ffc1a7', '#2aa145'],
                }],
            },
        });
    </script>
    </div>
</div>
</div>
</div>
</body>
</html>

```