

KONGU ENGINEERING COLLEGE

(An Autonomous Institution)

PROJECT

Personal Expense Tracker Application

TANUJ KUMAR M
(737819CSR206)

SRIBALAJI M
(737819CSR190)

THARUN PRASAD A
(737819CSR208)

VASANTH BABU J
(737819CSR216)

TEAM ID:
PNT2022TMID04498

TABLE OF CONTENTS

CHAPTER NO.

TITLE

1 INTRODUCTION

1.1 Project Overview

1.2 Purpose

2 LITERATURE REVIEW

2.1 Existing problem

2.2 References

2.3 Problem Statement Definition

3 IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

3.2 Ideation & Brainstorming

3.3 Proposed Solution

3.4 Problem Solution fit

4 REQUIREMENT ANALYSIS

4.1 Functional requirement

4.2 Non-Functional requirements

5 PROJECT DESIGN

5.1 Data Flow Diagrams Solution & Technical Architecture

5.2 User Stories

6 PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

6.2 Sprint Delivery Schedule

6.3 Reports from JIRA

7 CODING & SOLUTIONING

7.1 Feature 1

7.2 Database Schema

8 TESTING

8.1 Test Cases

8.2 User Acceptance Testing

9 RESULTS

9.1 Performance Metrics

10 ADVANTAGES & DISADVANTAGES

11 CONCLUSION

12 FUTURE SCOPE

13 APPENDIX

1. Introduction

1.1 Project Overview:

This project is based on an expense and income tracking system. This project aims to create an easy, faster and smooth tracking system between the expense and the income. This project also offers some opportunities that will help the user to how to manage the expenses in efficient way and also set have an option to set a limit for the amount to be used for that particular month. So, for the better expense tracking system, we developed our project that will help the users a lot.

1.2 Purpose:

An expense tracking app is an exclusive suite of services for people who seek to handle their earnings and plan their expenses and savings. When you track your spending, you know where your money goes and you can ensure that your money is used wisely. Tracking your expenditures also allows you to understand why you're in debt and how you got there. This will then help you design a be fitting strategy of getting out of debt. Many people in India live on a fixed income, and they find that towards the end of the month they don't have sufficient money to meet their needs.

2.Literature Survey

2.1 Existing Problem:

In existing, we need to maintain the Excel sheets, CSV files for the user daily, weekly and monthly expenses and there is no as such complete solution to keep a track of its daily expenses easily. To do so a person as to keep a log in a diary or in a computer system, also all the calculations need to be done by the user which may sometimes results in mistakes leading to losses. The existing system is not user friendly because data is not maintained perfectly. A writing audit is a study of insightful sources on a particular research. We found various similar products that have already been developed in the market. Unlike all those products, Personal Expense tracker (PET) provides security and graphical results. We provide the user to enter their wish-list before any purchase. It generates notifications to notify user about their money management and put a limit to weekly, monthly, expenses.

2.2 Problem Statement Definition:

Every earning people are mostly obsessed at the end of the month as they cannot remember where all of their money have gone when they have spent and ultimately have to sustain in little money minimizing their essential needs. There is no as such complete solution present easily to keep track of its daily expenditure easily and notify them if they are going to have money shortage. Personal finance applications will ask users to add their expenses and based on their expenses wallet balance will be updated which will be visible to the user. Also, users can get an analysis of their expenditure in graphical forms. They have an option to set a limit for the amount to be used for that particular month if the limit is exceeded the user will be notified with an alert. the main purpose of our application is to track the user's expenses.

3. Ideation and Proposed Solution

3.1 Empathy Map Canvas:

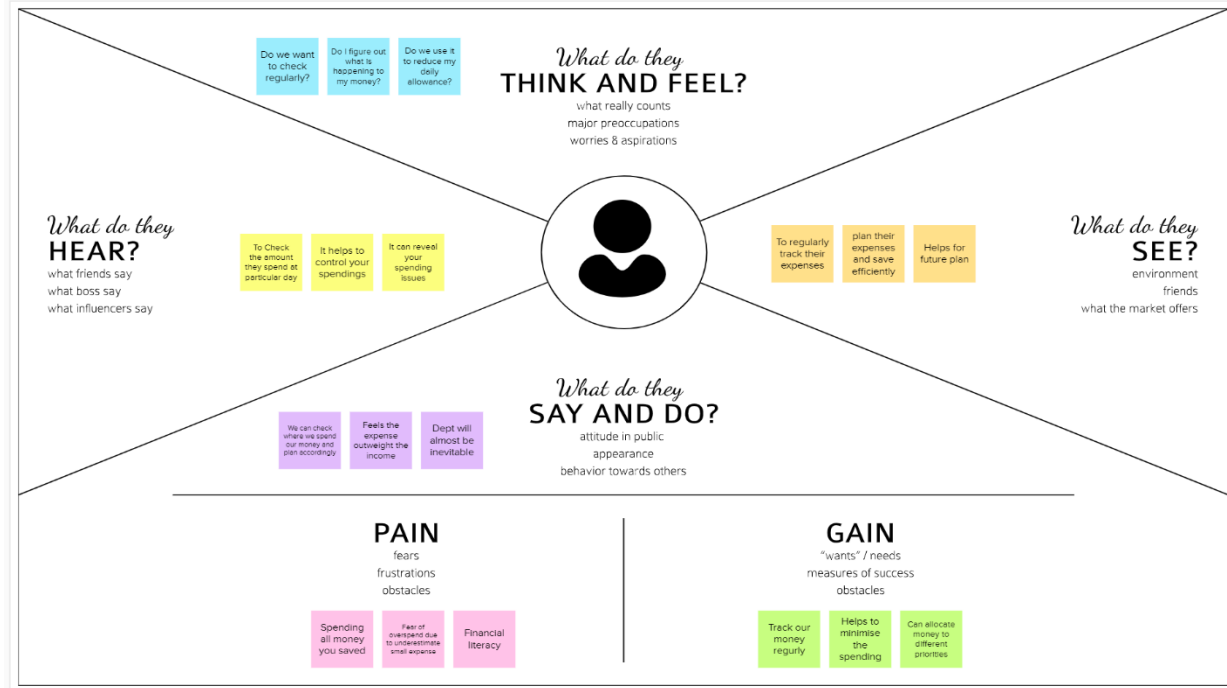
An empathy map canvas helps brands provide a better experience for users by helping teams understand the perspectives and mind set of their customers. Using a template to create an empathy map canvas reduces the preparation time and standardizes the process so you create empathy map canvases of similar quality.

Empathy Map Canvas

Gain insight and understanding on solving customer problems.

1

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



Share your feedback

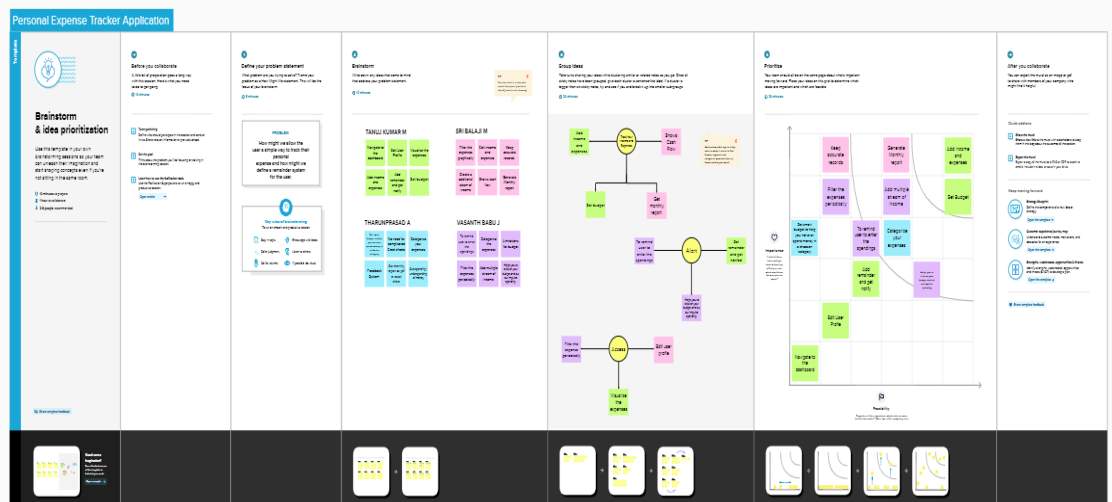
3.2 Ideation and Brainstorming:

Brainstorming is a group activity where everyone comes together to discuss strategies for growth and improvement. You can exchange ideas, share important information and use these meetings as informal catch-up sessions with your co-workers. Brainstorming combines a relaxed, informal approach to problem solving with lateral thinking. It encourages people to come up with thoughts and ideas that can, at first, seem a bit crazy. Some of these ideas can be crafted into original, creative solutions to a problem, while others can spark even more ideas.

Ideation is the process where you generate ideas and solutions through sessions such as Sketching, Prototyping, Brainstorming, Brain writing, Worst Possible Idea, and a wealth of other ideation techniques. Ideation is also the third stage in the Design Thinking process.



As you can see, ideation is not just a one-time idea generation or a brainstorming session. In fact, we can divide ideation in these three stages: generation, selection, and development.



3.3 Proposed Solution:

Expense Tracker is going to be a mobile application so that It can be accessed any time required. This application will have a two-tier architecture: first one is the database tier, where all the data and financial data will be stored. Second it will be the user interface which will support the application user communicate with the system and also store Information in the database. The proposed system should operate offline so it can be accessed at any time without internet availability. The proposed system should provide different categories for the user to select from and they can enter the amount and mode of payment. This system should be able to analyze the information, provide analytics on which category did the user spent most of their money. The proposed system should provide a user interface where the user could store and observe their past expenses. To develop a personal finance application which allows users to add their expenses and based on their expense wallet balance will be updated and displayed. Also, users can get an analysis of their expenditure in graphical forms. They have an option to set a limit for the amount to be used for that particular month and if the limit is exceeded the users will be notified with an email alert.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The problem is that people find it difficult to maintain track of their monthly expenses and to avoid obsessive spending.
2.	Idea / Solution description	The user can enter income and expenses into the personal cost tracker programme. As a result, the expense wallet is updated. A graphical breakdown of the expense could be obtained. The user is also notified if the budget's monthly maximum is surpassed.
3.	Novelty / Uniqueness	The personal expense tracker application helps the user not only in budgeting and accounting; it also provides the insights about money management through the analysis. The user also gets notified if the monthly limit is exceeded.

4.	Social Impact / Customer Satisfaction	The personal spending tracker programme not only assists the user in budgeting and accounting, but it also provides insights into money management through analysis. If the monthly limit is surpassed, the user is also notified.
5.	Business Model (Revenue Model)	The application may have a free and premium version, with the user having the option to upgrade to the premium version to gain access to additional features. Furthermore, the premium version may be ad-free.
6.	Scalability of the Solution	This application is not only for personal use, but it can also be extended to business organisations.

3.4 Problem Fit:

Problem-Solution fit canvas 2.0

PERSONAL EXPENSE TRACKER APPLICATION - TEAM ID: PNT2022TMID04498

Define CS, fit into CC	1. CUSTOMER SEGMENT(S) CS Who is your customer? <i>i.e. working parents of 0-5 y.o. kids</i> <ul style="list-style-type: none"> Customers are people who spend money either carelessly or with difficulty keeping track of it. Provides a whole lot of different categories of expenditure types to avoid mismatch of expenditure. The Need for Financial Management for Common People. 	6. CUSTOMER CONSTRAINTS CC What constraints prevent your customers from taking action or limit their choices of solutions? <i>i.e. spending power, budget, no cash, network connection, available devices.</i> <ul style="list-style-type: none"> The majority of online solutions include numerous ads that restrict their effectiveness. The approach proposed here features a function that allows you to view expenses visually. It also has a functionality that notifies you throughout email if a spending exceeds a predetermined limit. Devices That Are Available. Network Relationship 	5. AVAILABLE SOLUTIONS AS Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? <i>i.e. pen and paper is an alternative to digital notetaking</i> <ul style="list-style-type: none"> Applications that track expenses and are accessible for both iOS and Android. A personal expense tracking tool was created for this project. Calculating the total spendings of the user. Alerting the user nearing the budget. Notifying the user of spending above budget. Providing useful financial tips for better savings. Providing reports for assessments 	Explore AS, differentiate
	2. JOBS-TO-BE-DONE / PROBLEMS J&P Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides. <ul style="list-style-type: none"> This application's goal is to make it possible for users to keep track of their spending. The categories for the expenses are made available to the clients. They also have the choice of viewing the costs as a graphical depiction for the duration of a year, six months, etc. Fixed by establishing a cap on the amount that can be spent in a given month; if the cap is surpassed, the user will be notified through email. 	9. PROBLEM ROOT CAUSE RC What is the real reason that this problem exists? What is the back story behind the need to do this job? <i>i.e. customers have to do it because of the change in regulations.</i> <ul style="list-style-type: none"> Inappropriate expenses result in high taxes. Easy company forecasting; significant cost savings; difficulty in manually tracking expenses due to the abundance of payment options An opportunity lost A reduction in savings A poor investment No comprehensive and simple way to keep track of everyday spending excessive spending without effective management insufficient financial knowledge mistake prone and it takes time. 	7. BEHAVIOUR BE What does your customer do to address the problem and get the job done? <i>i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)</i> <ul style="list-style-type: none"> Start utilising the cost tracker software. Classify expenses as they are incurred to save money. Set a monthly spending cap and maintain separate in-hand wallet and online accounts. Ask your neighborhoods or coworkers for information. Obtain recommendations from professionals who are knowledgeable in the finance sector. 	
Focus on J&P, tap into BE, understand RC	3. TRIGGERS TR What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news. Knowing that these expenditure applications can help clients save a lot of money.	10. YOUR SOLUTION SL If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality. If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour. Create a flask-based personal cost tracker application, use the sendgrid framework to enable email-based expense notifications, and offer a graphical expense display option.	8. CHANNELS of BEHAVIOUR CH 8.1 ONLINE What kind of actions do customers take online? Extract online channels from #7 Virtual budget trackers have numerous advertising that, when clicked, capture information including account numbers if they are provided.	Focus on BE, tap into RC, understand J&P
	4. EMOTIONS: BEFORE / AFTER EM How do customers feel when they face a problem or a job and afterwards? <i>i.e. lost, insecure > confident, in control - use it in your communication strategy & design.</i> Before: Users are in a depressive state prior. After: Users feel ready to handle the cost.		8.2 OFFLINE What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development. <ul style="list-style-type: none"> Access to data that has already been downloaded. Make sure they are familiar with the tax laws by having them read the available books on taxes. 	
Identify strong TR & EM				Extract online & offline CH of BE

4. Requirement Analysis:

4.1 Functional Requirements:

1. Dashboard panel

The system shall authenticate the user and then display panel based on the particular identified user.

2. Add bill

The system shall allow the user to add bill details based on the user's need to track the type of expenses.

3. Expense planner

The system should graphically represent the current month figure based current month expenses and user's own budget share.

4. Expense tracker

The system should graphically represent the yearly expense numbers in form of report

5.Add notes

The system shall allow users to add notes to their expenses.

4.2 Non-Functional Requirements:

1. Usability

There is a consistency in all the modules and webpages. To ease the navigation there is a back tab to provide access to previous page. There is proper instruction on each page.

2. Reliability

Each data record is stored on a well-built efficient database schema. There is no risk of data loss. The internal evaluation of data is well coded.

3. Supportability

The system is well built to support any machine. Maintainability of the system is easy.

4. Performance

In order to ease the accessibility, the types of expenses are categorized along with an option to name on the own. Throughput of the system is increased due to light weight database support.

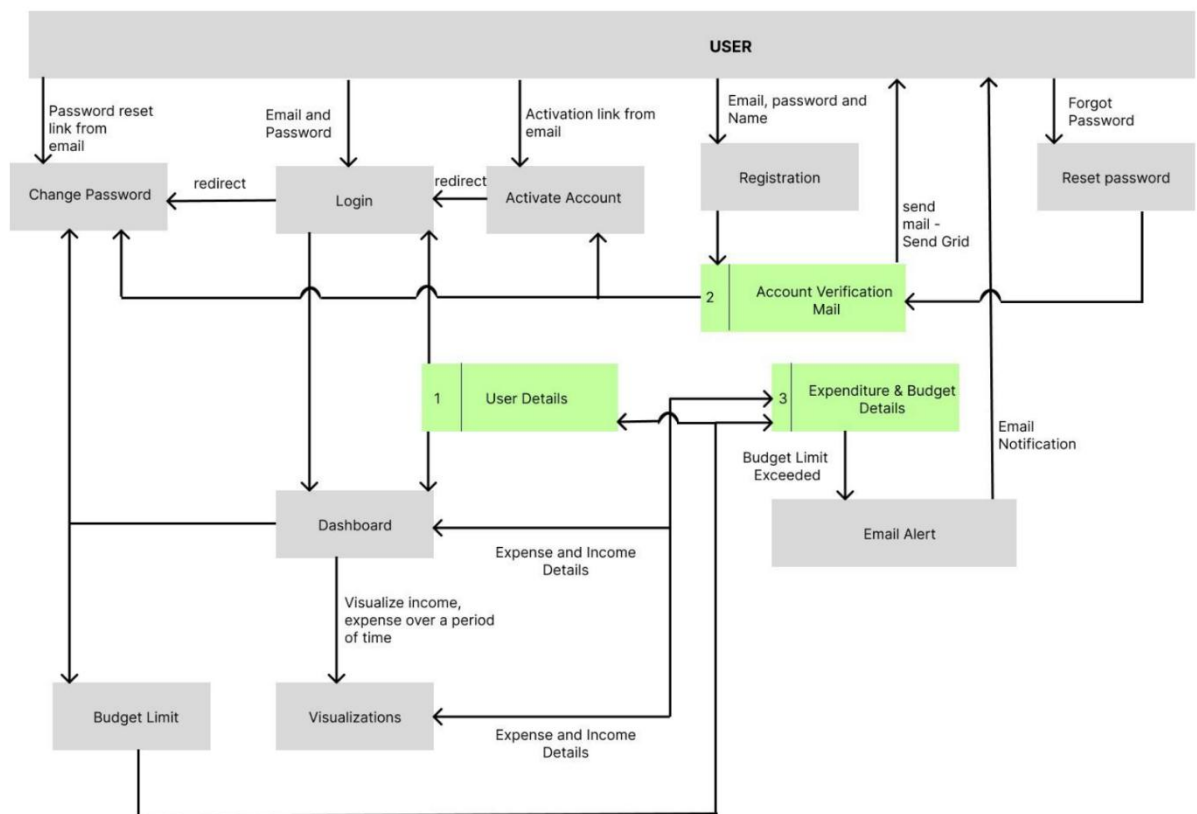
5. Availability

The system is available all the time, no time constraint.

5. Project Design

5.1 Data flow Diagrams:

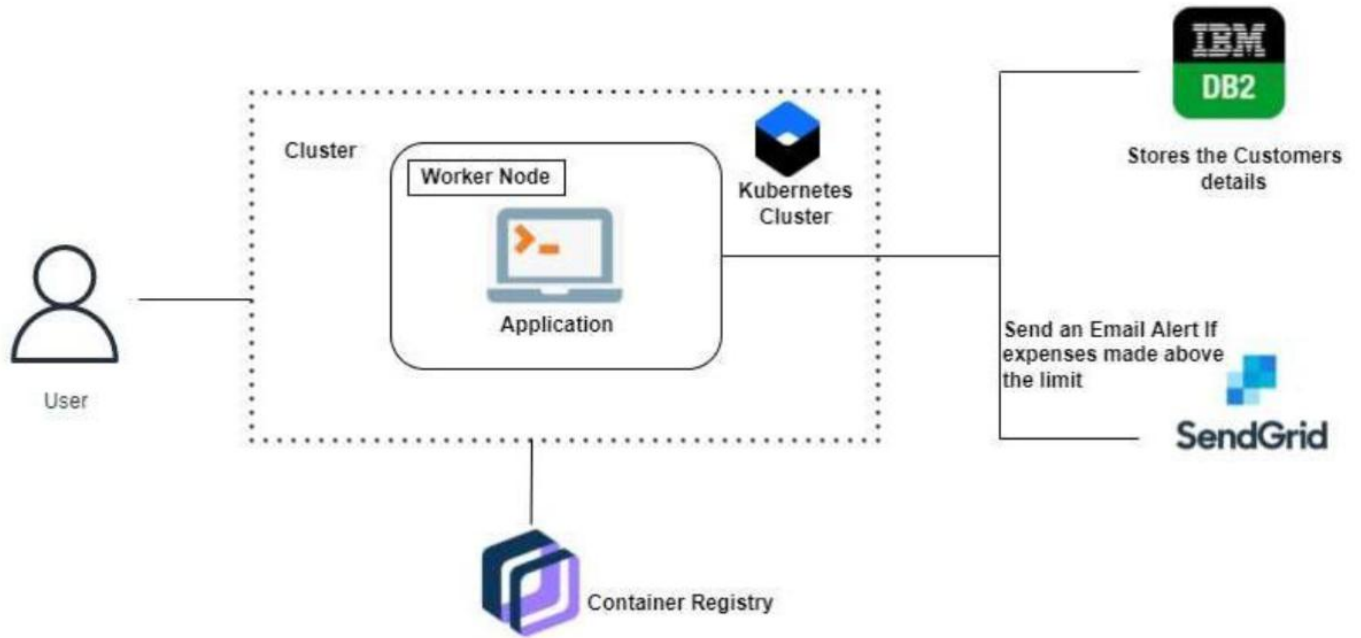
A data flow diagram (DFD) is a graphical or visual representation using a standardized set of symbols and notations to describe a business's operations through data movement. Data flow diagrams provide a straightforward, efficient way for organizations to understand, perfect, and implement new processes or systems. They're visual representations of your process or system, so they make it easy to understand and prune.



5.2 Solution & Technical Architecture:

Technical Architecture (TA) is a form of IT architecture that is used to design computer systems. It involves the development of a technical blueprint with regard to the arrangement, interaction, and interdependence of all elements so that system-relevant requirements are met.

Technical Architecture:



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, name and confirming my password.	2	High	Tanuj Kumar M
Sprint-1	Account Activation	USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Tanuj Kumar M
Sprint-1	Login	USN-3	As a user, I can log into the application by entering email & password	2	High	Sri Balaji M
Sprint- 2	Dashboard	USN-4	As a user, I can add expenses and income	2	High	Sri Balaji M
Sprint- 4	Dashboard	USN-5	As a user, I can change my password through reset link from email	1	Low	Tharun Prasad A
Sprint- 2	Dashboard	USN-6	As a user, I can view my past expenses and income over a period of time and visualize them	2	High	Tharun Prasad A
Sprint- 4	Notification	USN-7	As a user, when my monthly limit exceeds, I get a email notification	1	Low	Vasanth Babu J

Sprint- 4	Dashboard	USN-8	As a user, I can set a monthly expense limit	1	Low	Sri Balaji M
Sprint- 3	Forget password	USN-9	As a user, I can get a reset password link if I forget it through mail	2	High	Vasanth Babu J
Sprint- 3	Forget password	USN-10	As a user, I can change my password if I forget it	2	High	Vasanth Babu J

6.2 Sprint Delivery Schedule:

A sprint schedule is a document that outlines sprint planning from end to end. It's one of the first steps in the agile sprint planning process—and something that requires adequate research, planning, and communication.

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

6.3 Reports from JIRA:

JIRA is a software application used for issue tracking and project management. The tool, developed by the Australian software company Atlassian, has become widely used by agile development teams to track bugs, stories, epics, and other tasks

TAN board - Agile board - Jira

tracker-11.atlassian.net/jira/software/projects/TAN/boards/1

Jira Software Your work Projects Filters Dashboards People Apps Create

Search

Does your team need more from Jira? Get a free trial of our Standard plan.

Projects / Tanuj

TAN board

GROUP BY None

TO DO 7 ISSUES

- As a user, I can add expenses and income
TAN-8
- As a user, I can change my password through reset link from email
TAN-9
- As a user, I can view my past expenses and income over a period of time and visualize them
TAN-10

IN PROGRESS 3 ISSUES

- As a user, I can register for the application by entering my email, password, name and confirming my password.
TAN-5
- As a user, I will receive confirmation email once I have registered for the application
TAN-6
- As a user, I can log into the application by entering email & password
TAN-7

DONE

Quickstart

25°C Cloudy 9:17 PM 11/2/2022

Tanuj - Roadmap - Jira

tracker-11.atlassian.net/jira/software/projects/TAN/boards/1/roadmap

Jira Software Your work Projects Filters Dashboards People Apps Create

Search

Give feedback Share Export View settings

Projects / Tanuj

Roadmap

Status category Epic

	T	NOV	DEC	JAN 23
TAN-1 Sprint 1				
TAN-2 Sprint 2				
TAN-3 Sprint 3				
TAN-4 Sprint 4				
+ Create Epic				

Today Weeks Months Quarters Quickstart

25°C Cloudy 9:16 PM 11/2/2022

7. Coding & Solution:

7.1 Feature 1:

The screenshot shows a web browser window with the title "Expense Tracker". The address bar shows "Not secure | 159.122.177.199:31111/signup". The main content is a "Sign Up" form with the following fields:

- Name: Tharunprasad
- Email: tharunprasad.19cse@kongu.edu
- Username: Tharunprasad
- Password: (masked with dots)
- Confirm Password: (masked with dots)
- ☒ Remember me
- Sign Up button (blue)
- Cancel button (white)

The Windows taskbar at the bottom shows the date and time as 11:01 AM, 22-11-2022.

The screenshot shows the "EXPENSE TRACKER" application interface. The top navigation bar includes "Home", "+ Add Transactions", and "Logout". A message at the top states "You are already logged in".

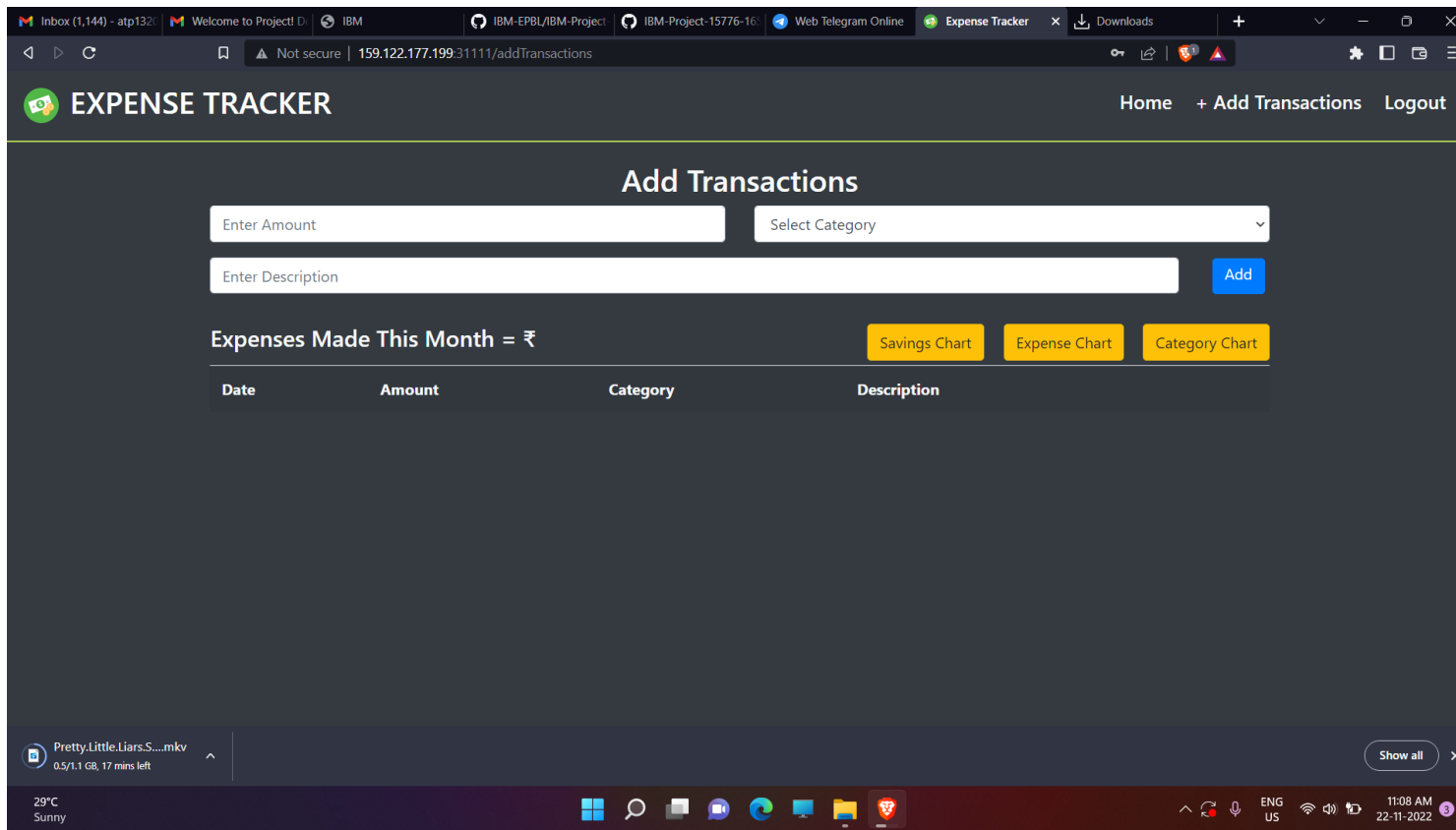
The main section is titled "Add Transactions" and contains the following form:

- Enter Amount: (text input)
- Select Category: (dropdown menu)
- Enter Description: (text input)
- Add button (blue)

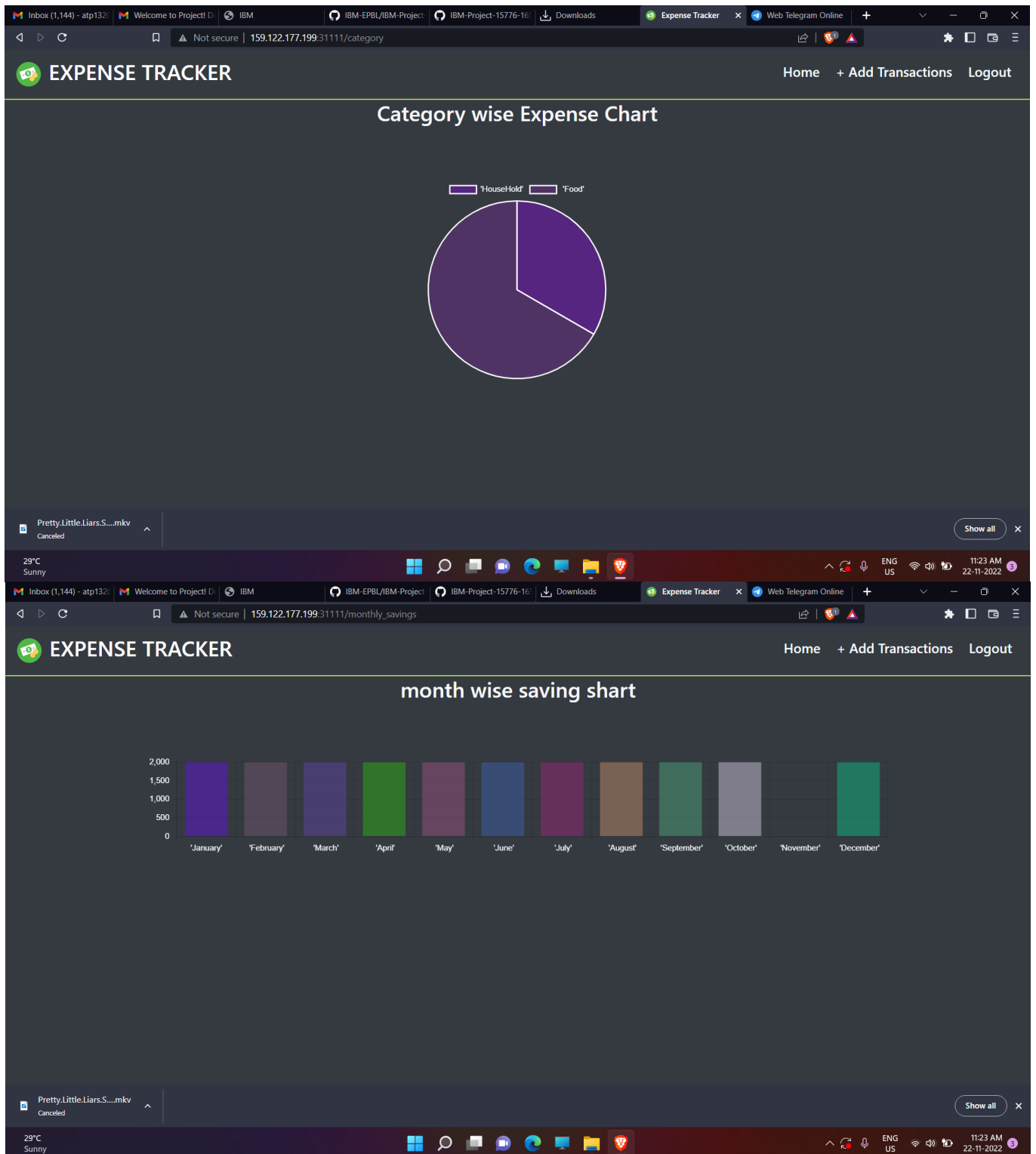
Below the form, it displays "Expenses Made This Month = ₹ 3000" and three buttons: "Savings Chart", "Expense Chart", and "Category Chart".

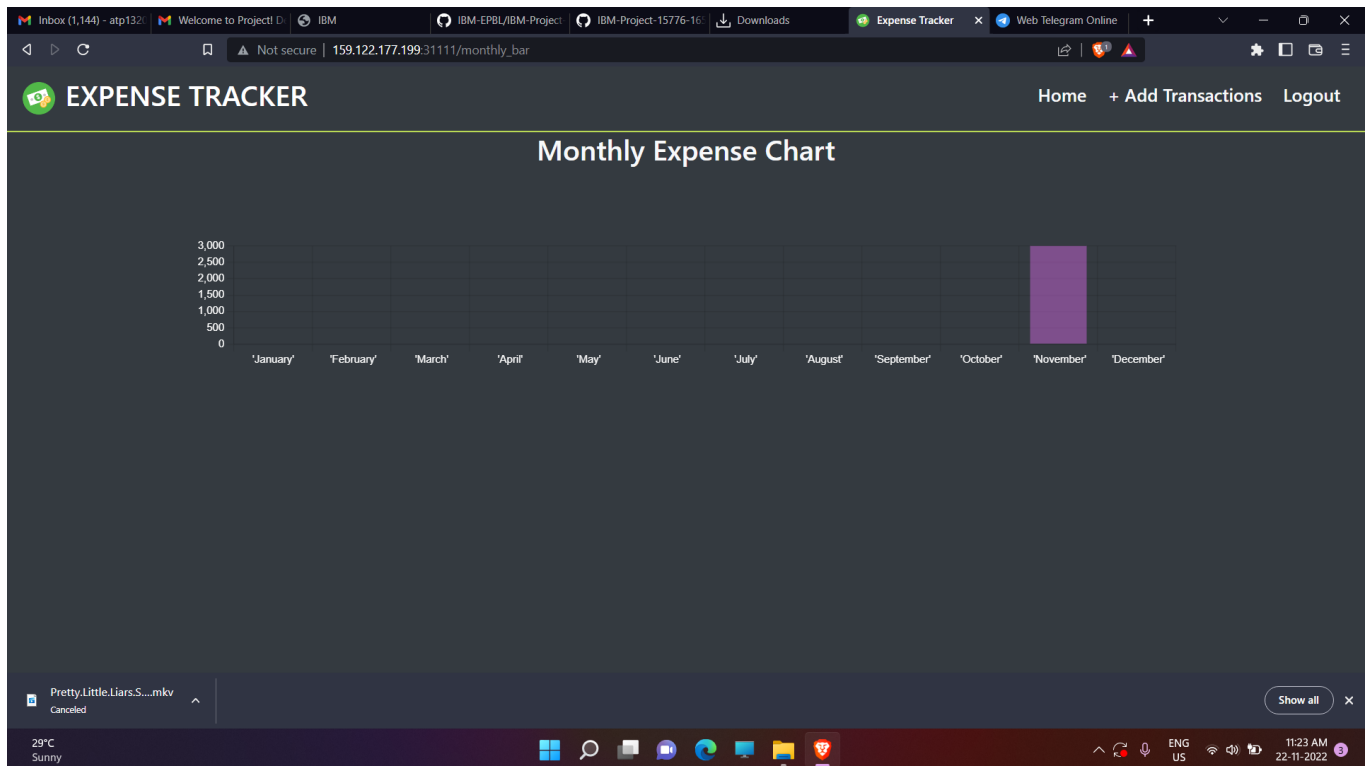
Date	Amount	Category	Description	
2022-11-22	1000	HouseHold	Per day 10	Edit
2022-11-22	2000	Food	Per day 100	Edit

The Windows taskbar at the bottom shows the date and time as 11:22 AM, 22-11-2022.



Feature 2:





Code:

App.py:

```
from functools import wraps

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

import plotly.graph_objects as go

from wtforms import Form, IntegerField, StringField, validators

from wtforms.validators import DataRequired


import connection

from flask_toastr import Toastr

app = Flask(__name__)

toastr = Toastr(app)

app.secret_key = "ibm_team_cloud"
```

```

@app.route('/')

def homepage():

    return render_template("landing.html")

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

def question():

    if request.method == 'POST':

        money = request.form.get('pmoney')

        budget = request.form.get('dbudget')

        goal = request.form.get('mgoal')

        useremail = session.get('usermail',None)

        pswd = session.get('pwd',None)

        print(useremail,pswd)

        conn = connection.establish()

        connection.setuser(conn,money,budget,goal,useremail,pswd)

        flash('Details added successfully', 'success')

        return redirect(url_for('addTransactions'))

    else:

        return render_template('questionnaire.html')

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

def signup():

    if 'logged_in' in session and session['logged_in'] == True:

        flash('You are already logged in', 'info')

        return redirect(url_for('addTransactions'))

    if request.method == 'POST':

        name = request.form.get('name')

        email = request.form.get('email')

```

```
username = request.form.get('username')

passw = request.form.get('psw')

rep_pass = request.form.get('psw-repeat')

if(passw != rep_pass):

    flash('Confirm password doesnot match','error')

    return redirect(url_for('signup'))

else:

    conn = connection.establish()

    if(connection.useremail_check(conn,email)==False):

        flash('User with email already exists, try again', 'warning')

        return redirect(url_for('signup'))

    else:

        session['usermail'] = email

        session['pwd'] = passw

        connection.insertuser(conn,name,email,username,passw)

        flash('You are now registered', 'success')

        return redirect(url_for('question'))

    else:

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

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

```
def login():
```

```
    if 'logged_in' in session and session['logged_in'] == True:
```

```
        flash('You are already logged in', 'info')
```

```
        return redirect(url_for('addTransactions'))
```

```

if request.method == 'POST' :

    email = request.form.get('email')

    password_input = request.form.get('psw')

    conn = connection.establish()

    res = connection.user_check(conn,email,password_input)

    if(res!=False):

        print('Login Success')

        session['logged_in'] = True

        session['userID'] = res['ID']

        flash('Login Successfull','success')

        return redirect(url_for('addTransactions'))

    else:

        print('Login Failure')

        flash('Incorrect Username/Password','error')

        return redirect(url_for('login'))

else:

    return render_template('login.html')

```

```

def is_logged_in(f):

    @wraps(f)

    def wrap(*args, **kwargs):

        if 'logged_in' in session:

            return f(*args, **kwargs)

        else:

            flash('Please login', 'info')

            return redirect(url_for('login'))

```

```

    return wrap

@app.route('/logout')
@is_logged_in
def logout():
    session.clear()

    flash('You are now logged out', 'success')

    return redirect(url_for('login'))


@app.route('/addTransactions', methods=['GET', 'POST'])
def addTransactions():
    if request.method == 'POST':
        amount = request.form['amount']

        description = request.form['description']

        category = request.form['category']

        conn = connection.establish()

        connection.inserttransac(conn,session['userID'],amount,description,category)

        flash('Transaction Successfully Recorded', 'success')

        return redirect(url_for('addTransactions'))
    else:
        conn = connection.establish()

        res = connection.gettotalsum(conn,session['userID'])

        total = res['SUM']

        dict= connection.getalltransac(conn,session['userID'])

        if len(dict)!=0:
            return render_template('addTransactions.html', totalExpenses=total, transactions=dict)
        else:

```

```

        return render_template('addTransactions.html', result=dict)

    return render_template('addTransactions.html')

class TransactionForm(Form):

    amount = IntegerField('Amount', validators=[DataRequired()])

    description = StringField('Description', [validators.Length(min=1)])

@app.route('/editCurrentMonthTransaction/<string:id>', methods=['GET', 'POST'])
@is_logged_in
def editCurrentMonthTransaction(id):

    form = TransactionForm(request.form)

    form = TransactionForm(request.form)

    if request.method == 'POST' and form.validate():

        amount = request.form['amount']

        description = request.form['description']

        conn = connection.establish()

        connection.updateTrans(conn,id,amount,description)

        flash('Transaction Updated', 'success')

        return redirect(url_for('addTransactions'))

    return render_template('editTransaction.html', form=form)

@app.route('/category')

def createBarCharts():

    conn = connection.establish()

    res = connection.gettotalsum(conn, session['userID'])

    total = res['SUM']

```



```

dict = connection.getalltransac(conn, session['userID'])

if len(dict) > 0:

    values = []

    labels = []

    print(dict)

    for transaction in dict:

        values.append(transaction['amt'])

        labels.append(transaction['cat'])

    print(labels)

    print(values)

    fig = go.Figure(data=[go.Pie(labels=labels, values=values)])

    fig.update_traces(textinfo='label+value', hoverinfo='percent')

    fig.update_layout(title_text='Category Wise Pie Chart For Current Year')

    #fig.show()

    return render_template('chart.html',context={'labels':labels,'value':values})

```

```

@app.route('/monthly_bar')

```

```

def monthlyBar():

    conn = connection.establish()

    res = connection.gettotalsum(conn, session['userID'])

    total = res['SUM']

    dict = connection.getalltransac(conn, session['userID'])

    if len(dict) > 0:

        year = []

        value = []

        print(dict)

```

```
d={'January':0,'February':0,'March':0,'April':0,'May':0,'June':0,'July':0,'August':0,'September':0,'October':0,'November':0,'December':0}
```

```
for transaction in dict:
```

```
    d[transaction['date'].strftime("%B")] += transaction['amt']
```

```
print(d)
```

```
for t,a in d.items():
```

```
    year.append(t)
```

```
    value.append(a)
```

```
print(year)
```

```
print(value)
```

```
fig = go.Figure([go.Bar(x=year, y=value)])
```

```
fig.update_layout(title_text='Monthly Bar Chart For Current Year')
```

```
#fig.show()
```

```
#cur.close()
```

```
return render_template('chart1.html',context={'labels':year,'value':value})
```

```
@app.route('/monthly_savings')
```

```
def monthllysave():
```

```
    conn = connection.establish()
```

```
    res = connection.gettotalsum(conn, session['userID'])
```

```
    total = res['SUM']
```

```
    dict = connection.getalltransac(conn, session['userID'])
```

```
    r = connection.get_budget(conn, session['userID'])
```

```
    print(r)
```

```
    if len(dict) > 0:
```

```
        year = []
```

```
        value = []
```

```
print(dict)
```

```
d={'January':0,'February':0,'March':0,'April':0,'May':0,'June':0,'July':0,'August':0,'September':0,'October':0,'November':0,'December':0}
```

```
for transaction in dict:
```

```
    d[transaction['date'].strftime("%B")] += transaction['amt']
```

```
print(d)
```

```
for t,a in d.items():
```

```
    x=int(r['POCKETMONEY'])-a
```

```
    if x<0:
```

```
        x=0
```

```
    year.append(t)
```

```
    value.append(x)
```

```
print(year)
```

```
print(value)
```

```
fig = go.Figure([go.Bar(x=year, y=value)])
```

```
fig.update_layout(title_text='Monthly Savings Chart For Current Year')
```

```
#fig.show()
```

```
#cur.close()
```

```
#return redirect(url_for('addTransactions'))
```

```
return render_template('chart2.html',context={'labels':year,'value':value})
```

```
if __name__ == '__main__':
```

```
    app.run(debug=True)
```

AddTransaction:

```
{% extends 'layout.html' %} {% block body %}

<div class="add">

    <h2 class="text-light">Add Transactions</h2>

    {% from "include/formhelpers.html" import render_field %}

    <form class="form" method="POST" action="">

        <div class="form-group row">

            <div class="form-group col-md-6">

                <input

                    type="number"

                    placeholder="Enter Amount"

                    class="form-control"

                    name="amount"

                    value="{ { request.form.amount} }"

                />

            </div>

            <div class="form-group category col-md-6">

                <select name="category" id="category" class="form-control">

                    <option value="Miscellaneous" selected="selected" disabled>Select
Category</option>

                    <option value="Miscellaneous">Miscellaneous</option>

                    <option value="Food">Food</option>

                    <option value="Transportation">Transportation</option>

                    <option value="Groceries">Groceries</option>

                    <option value="Clothing">Clothing</option>

                    <option value="HouseHold">HouseHold</option>
```

```

        <option value="Rent">Rent</option>

        <option value="Bills and Taxes">Bills and Taxes</option>

        <option value="Vacations">Vacations</option>

    </select>

</div>

<div class="form-group col-md-10 col-lg-11">

    <input

        type="text"

        placeholder="Enter Description"

        name="description"

        class="form-control"

        value="{{ request.form.description }}"

    />

</div>

<div class="form-group col-md-2 col-lg-1 btn">

    <button type="submit" class="btn btn-primary">Add</button>

</div>

</div>

</form>

{ % if result != 0% }

<div class="current-month">

    <h4 class="text-light float-left">

        Expenses Made This Month = <span class="green-text expense">₹

{ { totalExpenses } } </span>

    </h4>

    <p class="text-light float-left swipe">Swipe to Edit/Delete</p>

```

[Category Chart](/category)

[Expense Chart](/monthly_bar)

[Savings Chart](/monthly_savings)

{ % for transaction in transactions % }

[Edit](editCurrentMonthTransaction/{ { transaction.id } })

{ % endfor % }

</div>

</div>

{ %endif% } { % endblock % }

Chart:

{ % extends 'layout.html' % } { % block body % }

<h2 style="text-align:center; " class="text-light">Category wise Expense Chart</h2>

<input type="hidden" id="years" labels="{{ context.labels }}">

<input type="hidden" id="values" values="{{ { context.value } }}">

<div style="width:100;height:100%">

<canvas id="myChart"></canvas>

</div>

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

<script>

let years = document.getElementById('years').getAttribute('labels')

years = years.slice(1,years.length-1).split(",")

let values = document.getElementById('values').getAttribute('values')

values = values.slice(1,values.length-1).split(",")

console.log(years)

const ctx = document.getElementById('myChart').getContext('2d');

var barColors = [];

for (let index = 0; index < years.length; index++) {

const r = Math.floor(Math.random()*255);

const g = Math.floor(Math.random()*255);

const b = Math.floor(Math.random()*255);

```
barColors.push('rgba('+r+', '+g+', '+b+', 0.4)')
```

```
}
```

```
const data = {
```

```
  labels: years,
```

```
  datasets: [{
```

```
    label: 'My Expenses',
```

```
    data: values,
```

```
    backgroundColor: barColors,
```

```
    hoverOffset: 4
```

```
  ]]
```

```
};
```

```
Chart.defaults.color = "white";
```

```
var myChart = new Chart(ctx, {
```

```
  type: 'pie',
```

```
  data: data,
```

```
  options: {
```

```
    legend: {
```

```
      display: false
```

```
    },
```

```
    maintainAspectRatio: false,
```

```
  }
```

```
});
```

```
</script>
```

```
{% endblock % }
```


Login:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
```

```
<link rel="shortcut icon" href="../static/images/money-icon.png" type="image/x-icon" />
```

```
<!-- CSS only -->
```

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">
```

```
{ { toastr.include_jquery() } }
```

```
{ { toastr.include_toastr_css() } }
```

```
{ { toastr.message() } }
```

```
</head>
```

```
<title>Expense Tracker</title>
```

```
<style>
```

```
body { font-family: Arial, Helvetica, sans-serif;margin: 0;}
```

```
* { box-sizing: border-box;}
```

```
/* Full-width input fields */
```

```
input[type=text], input[type=password] {
```

```
width: 100%;
```

```
padding: 15px;
```

```
margin: 5px 0 22px 0;
```

```
display: inline-block;

border: none;

background: #f1f1f1;

}
```

```
/* Add a background color when the inputs get focus */

input[type=text]:focus, input[type=password]:focus {

    background-color: #ddd;

    outline: none;

}
```

```
/* Set a style for all buttons */

button {

    background-color: #3A3B3C;

    color: white;

    padding: 7px 22px;

    margin: 8px 2px;

    border: none;

    cursor: pointer;

    width: 100%;

    opacity: 0.9;

}
```

```
button:hover {

    opacity: 1;

}
```

```
/* Extra styles for the cancel button */
```

```
.cancelbtn {  
    padding: 14px 20px;  
    background-color: #ffffff;  
    color: black;  
    border: 1px solid #007BFF;  
}
```

```
/* Float cancel and signup buttons and add an equal width */
```

```
.signupbtn{  
    background-color: #007BFF;  
    color: white;  
}
```

```
.cancelbtn, .signupbtn {  
  
}
```

```
/* Add padding to container elements */
```

```
.container {  
    padding: 16px;  
}
```

```
/* The Modal (background) */
```

```
.modal {  
    display: block; /* Hidden by default */  
    position: fixed; /* Stay in place */  
    z-index: 1; /* Sit on top */  
    left: 0;
```

```
top: 0;

width: 100%; /* Full width */

height: 100%; /* Full height */

overflow: auto; /* Enable scroll if needed */

background-color: #ffffff;

padding-top: 50px;

}

/* Modal Content/Box */

.modal-content {

    background-color: #fefefe;

    margin: 5% auto 15% auto; /* 5% from the top, 15% from the bottom and centered / width: 40%; /
    Could be more or less, depending on screen size */

}

/* Style the horizontal ruler */

hr {

    border: 1px solid #f1f1f1;

    margin-bottom: 25px;

}

/* The Close Button (x) */

.close {

    position: absolute;

    right: 35px;

    top: 15px;

    font-size: 40px;

    font-weight: bold;
```

```
    color: #f1f1f1;

}

.close:hover,

.close:focus {

    color: #f44336;

    cursor: pointer;

}

/* Clear floats */

.clearfix::after {

    content: "";

    clear: both;

    display: table;

    display: flex;

    justify-content: space-between ;

}

a.clearfix{

text-decoration: none;

}

/* Change styles for cancel button and signup button on extra small screens*/

@media screen and (max-width: 300px) {

    .cancelbtn, .signupbtn {

        width: 100%;

    }

}

.topnav {
```

```
overflow: hidden;

background-color: #3A3B3C;

color: #ffffff;

height: 50px;

text-align: right;

align: right;

width: 100%;

}
```

```
.topnav .login-container {

float: right;

}
```

```
.topnav .head {

float: left;

text-align: right;

align: right;

padding: 14px 12px;

text-decoration: none;

font-size: 17px;

hover: #32CD32;

}
```

```
.head-img{

float: left;

margin-top: 5px;

margin-left : 7px;

color: #ffffff;

text-align: right;
```

```
    align: right;
}

.myHeading{
    font-weight: bold;
    text-align: center;
    font-size: 30px;
}

.container-car{
    margin-top: 30px;
}

.mySlides w3-display-container w3-center
{
height: 20%;
}

.card {
    box-shadow: 0 4px 8px 0 rgba(0,0,0,0.2);
    transition: 0.3s;
    margin: 5em;
}

.card:hover {
    box-shadow: 0 8px 16px 0 rgba(0,0,0,0.2);
}

img{
    width: 432.8px;
```

```
height: 288.45px;

}

</style>

<body>

<div class="modal">

    <form class="modal-content" action="/login" method="post">

        <div class="container">

            <h1 class="myHeading">Sign In</h1>

            <hr>

            <label ><b>Email</b></label>

            <input type="text" placeholder="Enter Email" name="email" required>

            <label><b>Password</b></label>

            <input type="password" placeholder="Enter Password" name="psw" required>

            <label>

                <input type="checkbox" checked="checked" name="remember" style="margin-bottom:15px">
Remember me

            </label>

            <div class="clearfix">

                <a href="/login" style="text-decoration:none;"></a><button type="submit"
class="signupbtn">Sign In</button></a>

                <a href="/" style="text-decoration:none;" ><button type="button"
class="cancelbtn">Cancel</button></a>

            </div>

        </div>

    </form>
```


</div>

{ { toastr.include_toastr_js() } }

</body>

</html>

Home:

<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">

<link rel="shortcut icon" href="../static/images/money-icon.png" type="image/x-icon" />

<!-- CSS only -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi" crossorigin="anonymous">

{ { toastr.include_jquery() } }

{ { toastr.include_toastr_css() } }

{ { toastr.message() } }

</head>

<title>Expense Tracker App</title>

<style>

body { font-family: Arial, Helvetica, sans-serif;margin: 0;}

* { box-sizing: border-box;}

/* Full-width input fields */

```
input[type=text], input[type=password] {  
    width: 100%;  
    padding: 15px;  
    margin: 5px 0 22px 0;  
    display: inline-block;  
    border: none;  
    background: #f1f1f1;  
}  
  
/* Add a background color when the inputs get focus */  
input[type=text]:focus, input[type=password]:focus {  
    background-color: #ddd;  
    outline: none;  
}  
  
/* Set a style for all buttons */  
button {  
    background-color: #3A3B3C;  
    color: white;  
    padding: 7px 22px;  
    margin: 8px 2px;  
    border: none;  
    cursor: pointer;  
    width: 100%;  
    opacity: 0.9;
```

```
}
```

```
button:hover {  
    opacity: 1;  
}
```

```
/* Extra styles for the cancel button */
```

```
.cancelbtn {  
    padding: 14px 20px;  
    background-color: #f44336;  
}
```

```
/* Float cancel and signup buttons and add an equal width */
```

```
.cancelbtn, .signupbtn {  
    float: left;  
    width: 50%;  
}
```

```
/* Add padding to container elements */
```

```
.container {  
    padding: 16px;  
}
```

```
/* The Modal (background) */
```

```
.modal {  
    display: none; /* Hidden by default */
```

```
position: fixed; /* Stay in place */
z-index: 1; /* Sit on top */
left: 0;
top: 0;
width: 100%; /* Full width */
height: 100%; /* Full height */
overflow: auto; /* Enable scroll if needed */
background-color: #474e5d;
padding-top: 50px;
}
```

```
/* Modal Content/Box */
```

```
.modal-content {
    background-color: #fefefe;
    margin: 5% auto 15% auto; /* 5% from the top, 15% from the bottom and centered */
    border: 1px solid #888;
    width: 40%; /* Could be more or less, depending on screen size */
}
```

```
/* Style the horizontal ruler */
```

```
hr {
    border: 1px solid #f1f1f1;
    margin-bottom: 25px;
}
```

```
/* The Close Button (x) */
```

```
.close {  
  
    position: absolute;  
  
    right: 35px;  
  
    top: 15px;  
  
    font-size: 40px;  
  
    font-weight: bold;  
  
    color: #f1f1f1;  
  
}
```

```
.close:hover,  
  
.close:focus {  
  
    color: #f44336;  
  
    cursor: pointer;  
  
}
```

```
/* Clear floats */
```

```
.clearfix::after {  
  
    content: "";  
  
    clear: both;  
  
    display: table;  
  
}
```

```
/* Change styles for cancel button and signup button on extra small screens */
```

```
@media screen and (max-width: 300px) {  
  
    .cancelbtn, .signupbtn {  
  
        width: 100%;
```

```
}
```

```
}
```

```
.topnav {
```

```
    overflow: hidden;
```

```
    background-color: #1976d2;
```

```
    color: #ffffff;
```

```
    height: 50px;
```

```
    text-align: right;
```

```
    align: right;
```

```
    width: 100%;
```

```
}
```

```
.topnav .login-container {
```

```
    float: right;
```

```
}
```

```
.topnav .head {
```

```
    float: left;
```

```
    text-align: right;
```

```
    align: right;
```

```
    padding: 14px 12px;
```

```
    text-decoration: none;
```

```
    font-size: 17px;
```

```
    hover: #32CD32;
```

```
}
```

```
.head-img{
```

```
    float: left;
```

```
margin-top: 5px;

margin-left : 7px;

color: #ffffff;

text-align: right;

align: right;

}
```

```
.mySlides w3-display-container w3-center

{

height: 20%;

}
```

```
.card {

box-shadow: 0 4px 8px 0 rgba(0,0,0,0.2);

transition: 0.3s;

margin: 5em;

}
```

```
.card:hover {

box-shadow: 0 8px 16px 0 rgba(0,0,0,0.2);h

}
```

```
img{

width: 432.8px;
```

```
height: 288.45px;
```

```
}
```

```
.contentText{
```

```
font-size: 50px;
```

```
font-weight: bold;
```

```
text-align: center;
```

```
margin-top: 50px;
```

```
}
```

```
</style>
```

```
<body>
```

```
<div class="topnav">
```

```
<a class="head" style="color: white">EXPENSE TRACKER APPLICATION</a>
```

```
<div class="login-container">
```

```
<a href="/login" style=" text-decoration:none;"><button style="width:auto;">SIGN IN</button></a>
```

```
<a href="/signup" style=" text-decoration:none;"><button style="width:auto;">SIGN UP</button></a>
```

```
</div>
```

```
</div>
```

```
<div class="mainContent">o
```

```
<div class="contentText">TRACK YOUR EXPENCE WITH OUR APPLICATION </div>
```

```
</div>
```

```
<!-- <div style="display: inline-flex">
```

```
<div class="card">
```

```

```



```
<div class="container">
```

```
  <h4 style="text-align: center;"><b>User Friendly Design</b></h4>
```

```
  <p style="text-align: center;">Inspirational designs, illustrations, graphic elements and effort-less navigation to enhance user experience.</p>
```

```
</div>
```

```
</div>
```

```
<div class="card">
```

```
  
```

```
  <div class="container">
```

```
    <h4 style="text-align: center;"><b>Live Automatic category-wise tracking</b></h4>
```

```
    <p style="text-align: center;">Track expenses category-wise from an in-built set or create your own custom categories.</p>
```

```
  </div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div style="display: inline-flex">
```

```
  <div class="card">
```

```
    
```

```
    <div class="container">
```

```
      <h4 style="text-align: center;"><b>Useful visualizations</b></h4>
```

```
      <p style="text-align: center;">
```

```
        Visualize information monthly, weekly or daily to analyze spending patterns.</p>
```

```
    </div>
```

```
  </div>
```

```
<div class="card">
```

```

```

```
<div class="container">m
```

```
<h4 style="text-align: center;"><b>Smart assistant</b></h4>
```

```
<p style="text-align: center;">AI bot to assist you in queries and enhance the app's overall  
experience.</p>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div> -->
```

```
<div class="row row-cols-1 row-cols-md-2">
```

```
<div class="col mb-4">
```

```
<div class="card">
```

```

```

```
<div class="card-body">
```

```
<h5 class="card-title">User Friendly Design</h5>
```

```
<p class="card-text">Inspirational designs, illustrations, graphic elements and effort-less navigation  
to enhance user experience.</p>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class="col mb-4">
```

```
<div class="card">
```

```

```

```
<div class="card-body">
```

```
<h5 class="card-title">Live Automatic category-wise tracking</h5>
```

```
<p class="card-text">Track expenses category-wise from an in-built set or create your own custom  
categories.</p>
```

</div>

</div>

</div>

<div class="col mb-4">

<div class="card">

<div class="card-body">

<h5 class="card-title">Useful visualizations</h5>

<p class="card-text">Visualize information monthly, weekly or daily to analyze spending patterns.</p>

</div>

</div>

</div>

<div class="col mb-4">

<div class="card">

<div class="card-body">

<h5 class="card-title">Smart assistant</h5>

<p class="card-text">AI bot to assist you in queries and enhance the app's overall experience.</p>

</div>

</div>

</div>

</div>

<script>

// Get the modal

var modal = document.getElementById('id01');

```
// When the user clicks anywhere outside of the modal, close it
```

```
window.onclick = function(event) {
```

```
  if (event.target == modal) {
```

```
    modal.style.display = "none";
```

```
  }
```

```
}
```

```
var myIndex = 0;
```

```
carousel();
```

```
function carousel() {
```

```
  var i;
```

```
  var x = document.getElementsByClassName("mySlides");
```

```
  for (i = 0; i < x.length; i++) {
```

```
    x[i].style.display = "none";
```

```
  }
```

```
  myIndex++;
```

```
  if (myIndex > x.length) { myIndex = 1 }
```

```
  x[myIndex-1].style.display = "block";
```

```
  setTimeout(carousel, 4000);
```

```
}
```

```
</script>
```

```
{ { toastr.include_toastr_js() } }
```

```
</body>
```

```
</html>
```

SignUp:

```
/* The Modal (background) */
.modal {
  display: block; /* Hidden by default */
  position: fixed; /* Stay in place */
  z-index: 1; /* Sit on top */
  left: 0;
  top: 0;
  width: 100%; /* Full width */
  height: 100%; /* Full height */
  overflow: auto; /* Enable scroll if needed */
  background-color: #ffffff;
  padding-top: 50px;
}

.cancelbtn {
  padding: 14px 20px;
  background-color: #ffffff;
  color: black;
  border: 1px solid #007BFF;
}

.signupbtn{
  background-color: #007BFF;
  color: white;
}
```

```
/* Modal Content/Box */
```

```
.modal-content {  
  
    background-color: #fefefe;  
  
    margin: 5% auto 15% auto; /* 5% from the top, 15% from the bottom and centered */  
  
    width: 40%; /* Could be more or less, depending on screen size */  
  
}
```

```
/* Style the horizontal ruler */
```

```
hr {  
  
    border: 1px solid #f1f1f1;  
  
    margin-bottom: 25px;  
  
}
```

```
/* The Close Button (x) */
```

```
.close {  
  
    position: absolute;  
  
    right: 35px;  
  
    top: 15px;  
  
    font-size: 40px;  
  
    font-weight: bold;  
  
    color: #f1f1f1;  
  
}
```

```
.close:hover,
```

```
.close:focus {  
  
    color: #f44336;
```

```
    cursor: pointer;
}
```

```
/* Clear floats */
```

```
.clearfix::after {
    content: "";
    clear: both;
    display: table;
}
```

```
a.clearfix{
text-decoration: none;
}
```

```
/* Change styles for cancel button and signup button on extra small screens */
```

```
@media screen and (max-width: 300px) {
    .cancelbtn, .signupbtn {
        width: 100%;
    }
}
```

```
.topnav {
    overflow: hidden;
    background-color: #3A3B3C;
    color: #ffffff;
    height: 50px;
    text-align: right;
    align: right;
```

```
width: 100%;  
  
}  
  
.topnav .login-container {  
  
    float: right;  
  
}
```

```
.topnav .head {  
  
    float: left;  
  
    text-align: right;  
  
    align: right;  
  
    padding: 14px 12px;  
  
    text-decoration: none;  
  
    font-size: 17px;  
  
    hover: #32CD32;  
  
}
```

```
.head-img{  
  
    float: left;  
  
    margin-top: 5px;  
  
    margin-left : 7px;  
  
    color: #ffffff;  
  
    text-align: right;  
  
    align: right;  
  
}
```

```
.container-car{  
  
    margin-top: 30px;
```



```
}
```

```
.mySlides w3-display-container w3-center
```

```
{
```

```
height: 20%;
```

```
}
```

```
.card {
```

```
box-shadow: 0 4px 8px 0 rgba(0,0,0,0.2);
```

```
transition: 0.3s;
```

```
margin: 5em;
```

```
}
```

```
.myHeading{
```

```
font-weight: bold;
```

```
text-align: center;
```

```
font-size: 30px;
```

```
}
```

```
.card:hover {
```

```
box-shadow: 0 8px 16px 0 rgba(0,0,0,0.2);
```

```
}
```

```
img{
```

```
width: 432.8px;
```

```
height: 288.45px;
```

```
}
```

</style>

<body>

<div class="container bg-light">

{% include 'include/messages.html' %} {% block body %} {% endblock %}

</div>

<div class="modal">

<form class="modal-content" action="/signup" method="post">

<div class="container">

<h1 class="myHeading">Sign Up</h1>

<hr>

<label>Name</label>

<input type="text" placeholder="Enter Name" name="name" required>

<label>Email</label>

<input type="text" placeholder="Enter Email" name="email" required>

<label>Username</label>

<input type="text" placeholder="Enter Username" name="username" required>

<label>Password</label>

<input type="password" placeholder="Enter Password" name="psw" required>

<label>Confirm Password</label>

```
<input type="password" placeholder="Confirm Password" name="psw-repeat" required>
```

```
<label>
```

```
<input type="checkbox" checked="checked" name="remember" style="margin-bottom:15px">
```

Remember me

```
</label>
```

```
<div class="clearfix">
```

```
<a href="/signup" style=" text-decoration: none;" ><button type="submit" class="signuptbtn">Sign  
Up</button></a>
```

```
<a href="/" style=" text-decoration: none;" ><button type="button"  
class="cancelbtn">Cancel</button></a>
```

```
</div>
```

```
</div>
```

```
</form>
```

```
</div>
```

```
{ { toastr.include_toastr_js() } }
```

```
</body>
```

```
</html>
```

7.3 Database Schema:

IBM Db2 on Cloud

Load DataLoad HistoryTablesViewsIndexesAliasesMQTsSequencesApplication objects

Find schemas or tables

Refresh

Schemas

Name	Type	Tables
SPQ76013	User	8

Total: 1, selected: 1

Tables

New table

Name	Schema	Properties
EXPENSES	SPQ76013	...
LIMITS	SPQ76013	...
PETA	SPQ76013	...
PETACATEGORY	SPQ76013	...
PETA_EXPENSES	SPQ76013	...
PETA_GROUPS	SPQ76013	...
REGISTER	SPQ76013	...
STUDENTS	SPQ76013	...

Total: 8, selected: 0

IBM Db2 on Cloud

Error: Please check network connectivity then try again.

Show logs

Find schemas or tables

Refresh

Tables

New table

Name	Schema	Properties
EXPENSES	SPQ76013	...
LIMITS	SPQ76013	...
PETA	SPQ76013	...
PETACATEGORY	SPQ76013	...
PETA_EXPENSES	SPQ76013	...
PETA_GROUPS	SPQ76013	...
REGISTER	SPQ76013	...
STUDENTS	SPQ76013	...

Total: 8, selected: 0

Table definition

EXPENSES

No statistics available.

Name	Data type	Nullable	Length	Scale
ID	VARCHAR	Y	255	0
DATES	TIMESTAMP	Y	10	6
EXPENSENAME	VARCHAR	Y	255	0
AMOUNT	INTEGER	Y		0
PAYMODE	VARCHAR	Y	255	0
CATEGORY	VARCHAR	Y	255	0

View data

8. Testing

8.1 Test Cases

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
LoginPage_TC_OO 1	Functional	Home Page	Verify user is able to see the Login/Signup popup when user clicked on My account button		1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Signup popup displayed or not	http://cloud-object-storage-ek-cos-static-web-hosting-8-n.s3-web.jp-tok.cloud-objects-storage.apdomain.cloud/authenticate	Login/Signup popup should display	Working as expected	Pass	initial load is very less			Mohammed Kaif K
LoginPage_TC_OO 2	UI	Home Page	Verify the UI elements in Login/Signup popup		1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Signup popup with below UI elements: a.email text box b.password text box c.Login button d.New customer? Create account link e.Last password? Recovery password link	http://cloud-object-storage-ek-cos-static-web-hosting-8-n.s3-web.jp-tok.cloud-objects-storage.apdomain.cloud/authenticate	Application should show below UI elements: a.email text box b.password text box c.Login button with orange colour d.New customer? Create account link e.Last password? Recovery password link	Working as expected	Pass	user friendly and attractive			Karthick V
LoginPage_TC_OO 3	Functional	Home page	Verify user is able to log into application with Valid credentials		1.Enter URL(https://shopnizer.com/) and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button	Username: chalam@gmail.com password: Testing123	User should navigate to user account homepage		Pass	components are routed			Harish M
LoginPage_TC_OO 4	Functional	Login page	Verify user is able to log into application with Invalid credentials		1.Enter URL(https://shopnizer.com/) and click go 2.Click on My Account dropdown button 3.Enter Invalid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button	Username: chalam@gmail.com password: Testing123	Application should show 'Incorrect email or password' validation message.		Pass	components are routed			Lokesh V

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
LoginPage_TC_OO 4	Functional	Login page	Verify user is able to log into application with Invalid credentials		1.Enter URL(https://shopnizer.com/) and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter Invalid password in password text box 5.Click on login button	Username: chalam@gmail.com password: Testing123678686786876876	Application should show 'Incorrect email or password' validation message.		Pass	comfortable to use			Sethu S
LoginPage_TC_OO 5	Functional	Login page	Verify user is able to log into application with Invalid credentials		1.Enter URL(https://shopnizer.com/) and click go 2.Click on My Account dropdown button 3.Enter Invalid username/email in Email text box 4.Enter Invalid password in password text box 5.Click on login button	Username: chalam password: Testing123678686786876876	Application should show 'Incorrect email or password' validation message.		Pass	user friendly and attractive			Vibin T
LoginPage_TC_OO 6	Functional	Dashboard	Add income and expense and check whether the insights are changing		1.After login 2.Add income and expense 3.View charts	Income:Rs INR Expense:Rs INR	View charts		Pass	Every events are changing accurately			Sudhan A
LoginPage_TC_OO 7	Functional	Dashboard	Email insights		insights are viewed in mail		Mail insights		Pass	insights are reached on time			Srinath S

8.2 User Acceptance Testing

Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	5	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	0	0	0
Skipped	0	0	0	0	0
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	75

Test Case Analysis

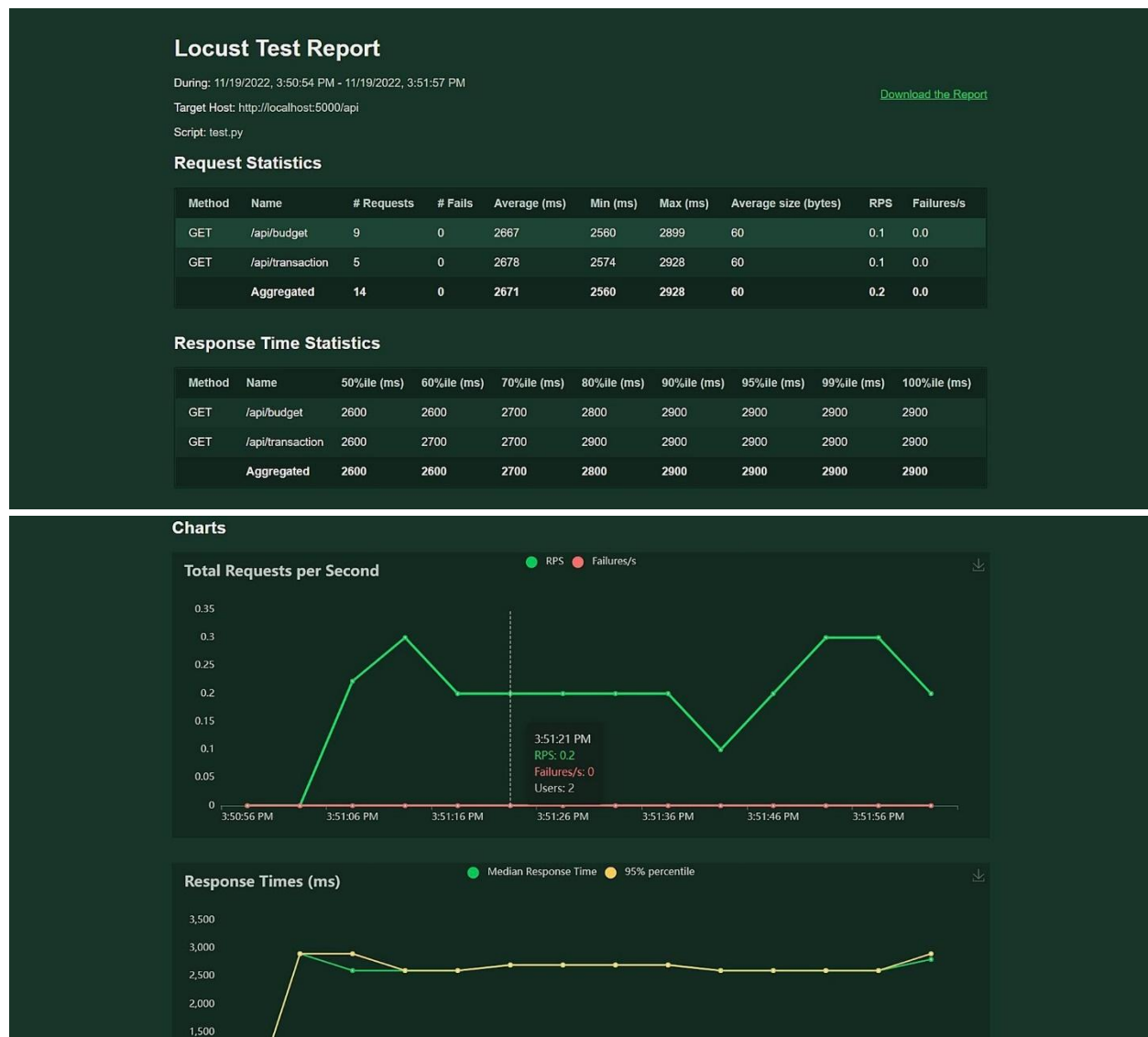
This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	29	0	0	29
Security	4	0	0	4
Outsource Shipping	6	1	0	5
Exception Reporting	7	0	0	7
Final Report Output	4	0	0	4
Version Control	8	0	0	8

9. Results

9.1 Performance Metrics Locust

Locust is a Python-based testing tool used for load testing and user behavior simulation. Load testing is the practice of testing a software application with the primary purpose of stressing the application's capabilities. Locust is a tool that creates a set of testing functions that simulate a heavy number of users.



The screenshot shows an Excel spreadsheet with the following data:

Type	Name	Request C	Failure C	Median Re	Average R	Min Respo	Max Respo	Average C	Requests/	Failures/s	50%	66%	75%	80%	90%	95%	98%	99%	99.90%	99.99%	100%
GET	/api/budge	9	0	2600	2667.621	2560.681	2899.862	60	0.141371	0	2600	2600	2700	2800	2900	2900	2900	2900	2900	2900	2900
GET	/api/trans	5	0	2600	2678.676	2574.646	2928.723	60	0.07854	0	2600	2700	2700	2900	2900	2900	2900	2900	2900	2900	2900
	Aggregator	14	0	2600	2671.569	2560.681	2928.723	60	0.219911	0	2600	2700	2700	2800	2900	2900	2900	2900	2900	2900	2900

Chapter-10:

10.1 Advantages:

1. **You have no control over your money** If you don't check your spending and create a budget, you will have no control whatsoever on your money. Instead, money will control you, and you will either have perpetual lack of funds or you will end up steeped in debt. A money manager app helps you decide between short-term and long-term spending.
2. **You have no financial goals:** If you are spending money frivolously, you will not have money to set financial goals. However, when you have a daily expense manager, you will be able to work with limited resources and use your money in a wise manner so that you can create financial goals and ensure you meet them.
3. **You are unaware what is happening with your money:** If you are clueless about how much is your inflow and how much you are spending; you will not know at the end of the month what happened to your money. An expense tracker helps you figure out what is happening to your money, and whether you can afford something you want.
4. **You spend and save in a haphazard manner:** If you don't have great financial management skills, you will not know how to categorize your expenses. However, tracking your expenses and budgeting them will help you become aware of how much you have to allocate to each expense category, and if you are short, you will be able to make adjustments with ease.
5. **You have no clue about making your money work for you:** In this day and age, when expenses are going through the roof, it has become crucial that you learn to make your money work for you so that you can create a nest egg for the future.

6. **You don't have funds for emergencies** Remember, emergencies come when you least expect. Hence, if you don't have money stashed away for a rainy day, you will end up borrowing from family and friends. This way you could get into debt that will be difficult to pay back due to your poor money management skills

10.2 Disadvantages:

Your information is less secure, and probably being used and sold. If the service is free, then the product is you. Mint.com, like other financial apps, is a free service. They have to pay their bills somehow, so regardless of what their privacy policy may or may not say, just assume that your spending history and trends are going to be recorded and analysed, by someone, somewhere. Now, you shouldn't have to worry about credit card fraud or identity theft, these companies are large enough and secure enough that you'll never have to worry about something like that. Just recognize that your information, most likely anonymous, will be used and potentially even sold. Personally, I have no problem with that, but if you do, then make sure you avoid these types of services. Automating everything to do with your finances can make you financially lazy. If your bills are paid automatically and your finances are track automatically, then what is there left for you to do? Not a lot, to be honest. So you might stop caring about what you're spending and where your money is going. Eventually you may look at your Mint data and realize that you've blown your budget over the last two months, but by then it is too late. So if you do choose to use this program, ensure that you are also being diligent in checking in on your finances. Set up a weekly or biweekly check for yourself to go through your finances and hit on all the important points.

11. Conclusion:

After making this application we assure that this application will help its users to manage the cost of their daily expenditure. It will guide them and aware them about their daily expenses. It will prove to be helpful for the people who are frustrated with their daily budget management, irritated because of amount of expenses and wishes to manage money and to preserve the record of their daily cost which may be useful to change their way of spending money. In short, this application will help its users to overcome the wastage of money

GitHub Link:

<https://github.com/IBM-EPBL/IBM-Project-15776-1659604472.git>

Demonstartion Link:

https://drive.google.com/file/d/1nN1tJHw9emqShEPDW4XtjvvnIoWJ0Hk0/view?usp=share_link

Hosted Link :

<http://159.122.177.199:31111/>