NALAIYA THIRAN - IBM PROJECT REPORT

Professional Readiness for Innovation, Employability and Entrepreneurship

ON

PERSONAL EXPENSE TRACKER

Submitted by

TEAM ID: PNT2022TMID23410

Team:

SR RAGHAV K R (113219031145)

AYUSH H N (113219031021)

SIVANESH K (113219031142)

THARUN KUMAR T (113219031701)

CONTENTS

CHAPTER 1
INTRODUCTION

A Personal Expense Tracker Application is a particular form of digital diary that aids in keeping track of all of our cash transitions and moreover offers daily, weekly, monthly, and yearly reports on all financial activities. User receives alerts to keep track of their income and expenses using the system in place. All data is kept in cloud for easy access at any time and from any location, provided the user has network connectivity. The Daily Expense Tracker's user interface is incredibly straightforward and appealing, making it simple to grasp and the finest approach to record our financial data.

1.1 PROJECT OVERVIEW

Simply put, personal finance includes all of the financial decisions and actions that a finance software facilitates by assisting you in effectively managing your finances. A personal finance software will not only assist you with accounting and budgeting, but it will also provide you with valuable advice on money management.

Users of personal finance applications will be prompted to enter their costs, after which their wallet balance will be updated and displayed to them. Users can also receive a graphical analysis of their expenses. They can choose to establish a cap on how much can be used in that month, and if the cap is surpassed, the user will receive an email alert.

1.2 PURPOSE

When you keep track of your spending, you can make sure your money is being utilized wisely and you will know where it goes. You can learn why you're in debt and how you got there by keeping track of your spending. You can then use this information to create a debt relief plan that works for you.

You may plan for both short-term and long-term expenses by using a budget to make sure you're not spending more than you're earning. It's a simple, practical solution for folks with all types of income and expenses to maintain order in their finances.

CHAPTER 2

LITERATURE SURVEY

The “Expense Tracker” is developed using Python flask for front end and IBM db2 for back end. Expense Tracker helps to maintain the record of daily expenses and monthly income of users from anywhere and also generates a monthly report of the expenses in pdf format. The Expense Tracker app tracks all the expenses and helps the user to manage his/her expenses so that the user is on the path of financial stability. The Tracking of expenses is categorized by week, month and year, it helps to see the more expenses made. To use the Expense Tracker the user has to sign up in such as name, phone no., address, email address, username, password and confirm password of the user. The user can get enlisted just a single time, per user can just one record. The remainder is set if the type is future expense. The whole subtleties of the income or expense can be seen or refreshed or can be erased by long pressing the specific rundown thing. The things in the rundown can be separated by month, year and date. When the month’s end is reached at the complete pay, all out past expenses and all-out future expenses are determined and shown for the user.

2.1 EXISTING PROBLEM

In the Existing System, when a user gets his/her salary, he/she can add that into the daily expense manager. Then after adding his/her salary details the user can expense manager stores all the details. If the user wants all the details of credit and debit card, he/she can get it through the Expense Tracker. Notification Manager also reminds about credit and debit details after the salary is updated.

A. User Registration and Creation - This application like the vast majority of the applications will have a user login screen and alternatives for enlistment. The user should enlist in this application when the person in question is using it for the first time. Nonetheless, the client who is now enlisted can login to the application utilizing their login accreditations that are made by the user at the hour of enrolment.

B. Adding Income and Expenses - This application will provide to choose the categories or type of income or expenses. Every user of the application has the option

to add incomes and expenses accordingly. Each record should have details of the date of occurrence of item, details of items etc.

C. Category Master - This module fundamentally relies upon the DB2 for putting away classification details and expense subtleties and income. The class exchange is put away in a DB2 database.

D. Management View - Date Wise The Expenses are recorded dependent on the Predetermined date insightful in this module. By recovering all the income and expense subtleties are seen as a rundown of exchange classes by our various costs. The income and expenses are recovered by utilizing DB2 queries and saw in advanced cell.

E. Management View - Category Wise The Expenses are recorded dependent on the Predetermined classification astute in this module. By retrieving all the income and expense subtleties are seen as a rundown of exchange classifications by our various expenses. The income and expenses are recovered by utilizing queries and saw in advanced cell.

F. Remainder - The Rest of is a caution generator module, for user recognition the alert/ready will review the user to add the Income or Expenses at day by day or Certain Period based on user need.

2.2 REFERENCES

[1] Y. Anitha, R. Ranjini, S. Gomathi, “Easy App for Expenses Manager Using Android '', International Journals of Computer Techniques, Volume: 3 Issue: 2, ISSN: 2394-2231 (MarchApril 2016).

[2] N. ZahiraJahan MCA., M. Phil, K. I. Vinodhini, “Personalized Expense Managing Assistant Using Android'', International Journals of Computer Techniques (IJCT), Volume: 3 Issue: 2, ISSN: 2394-2231 (March-April 2016).

[3] S. Chandini, T. Poojitha, D. Ranjith, V. J. Mohammed Akram, M. S. Vani, V. Rajyalakshmi, “Online Income and Expense Tracker'', International Research Journal

of Engineering and Technology (IRJET), Volume: 06 Issue: 3, e-ISSN: 2395-0056, p-ISSN: 2395- 0072 (March 2019).

[4] P. Thanapal, Mohammed Yaseen Patel, T. P. Lokesh Raj and J. Satheesh Kumar, “Income and Expense Tracker”, Indian Journal of Science and Technology, Vol 8(S2), ISSN: 0974-5645 (January 2014).

[5] Girish Bekaroo and Sameer Sunhaloo, “Intelligent Online Budget Tracker”, Computer Science and IT Education Conference (2014).

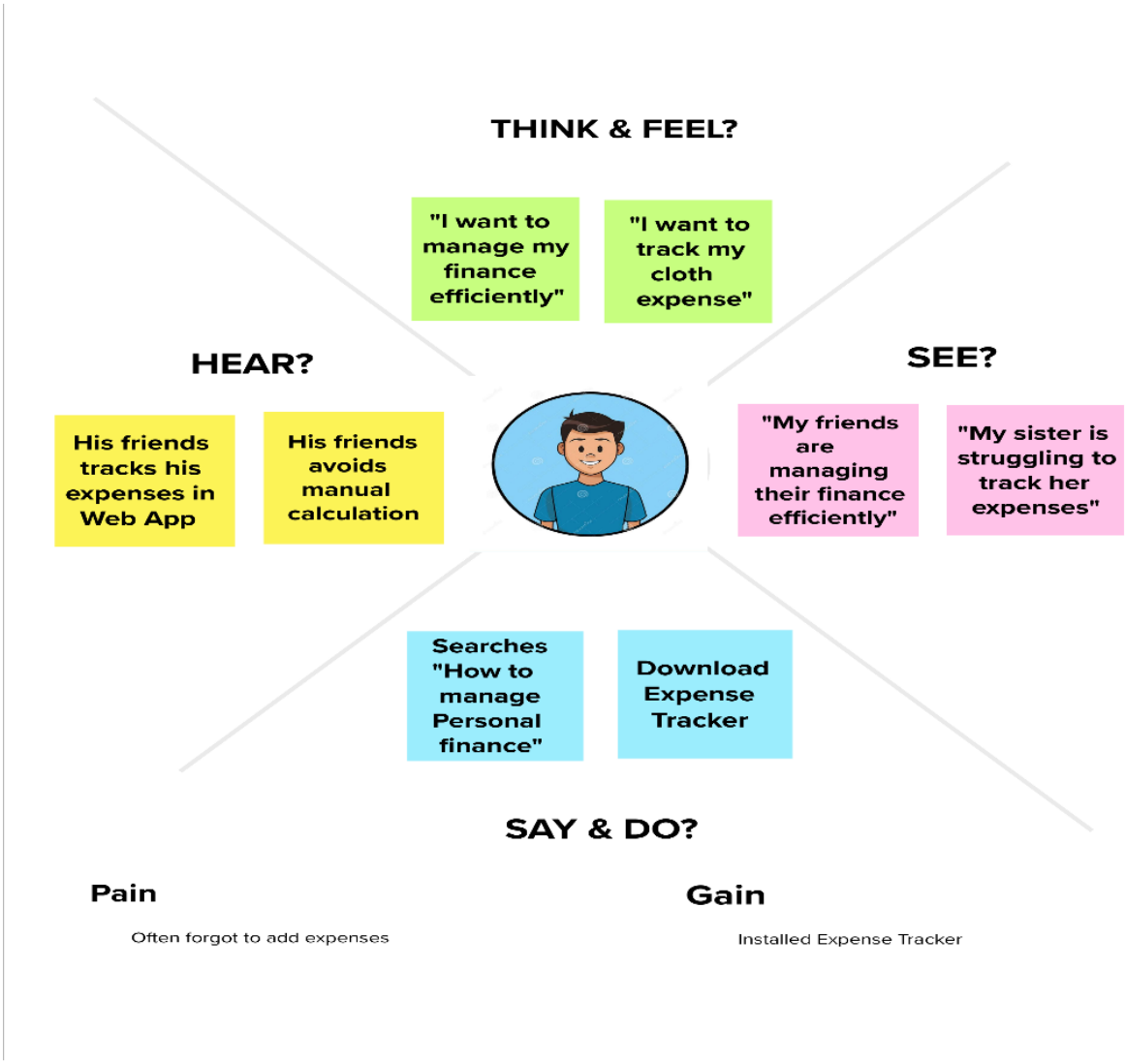
2.3 PROBLEM STATEMENT DEFINITION

In our daily life money is the most important portion and without it we cannot last one day on earth but if we keep on track all financial data then we can overcome this problem. Most of the people cannot track their expenses and income one way they face the money crisis and depression. This situation motivates us to make an android app to track all financial activities. Using the Personal Expense Tracker Application user can be tracking expenses day to day and making life tension free.

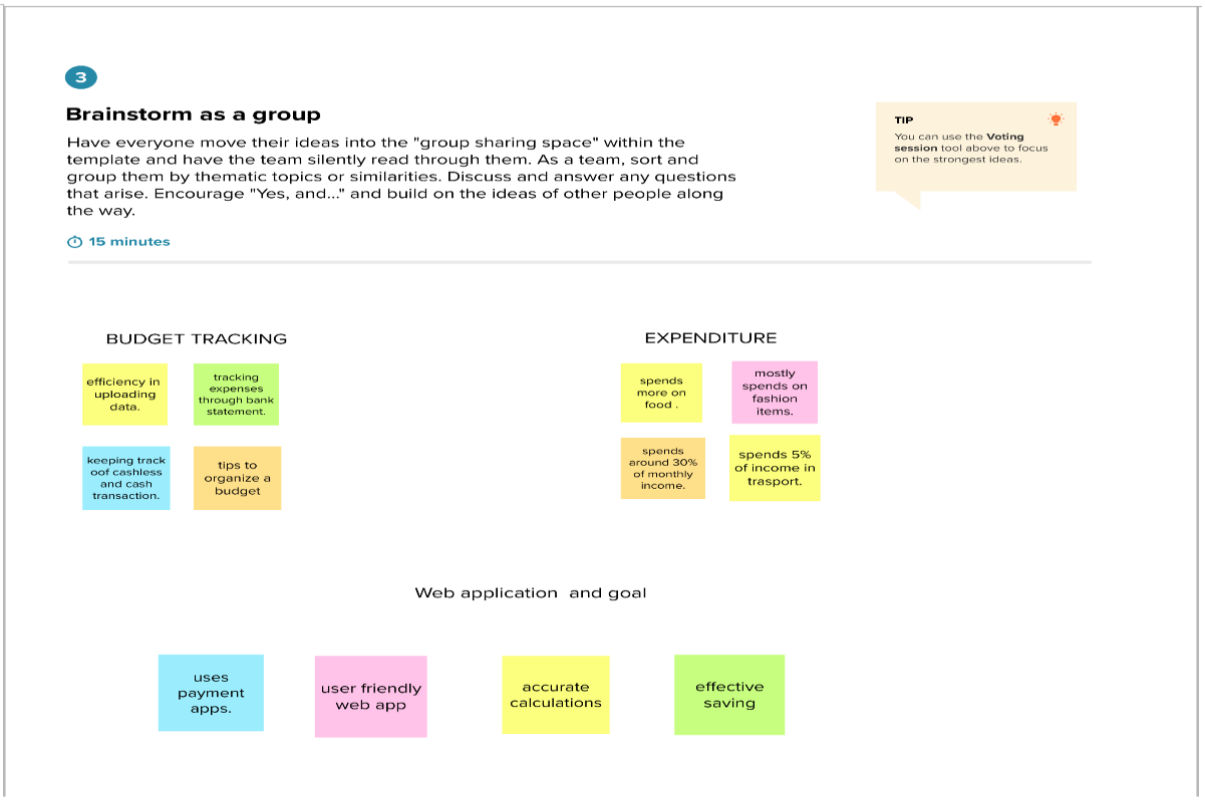
CHAPTER 3

IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



3.2 IDEATION AND BRAINSTORMING



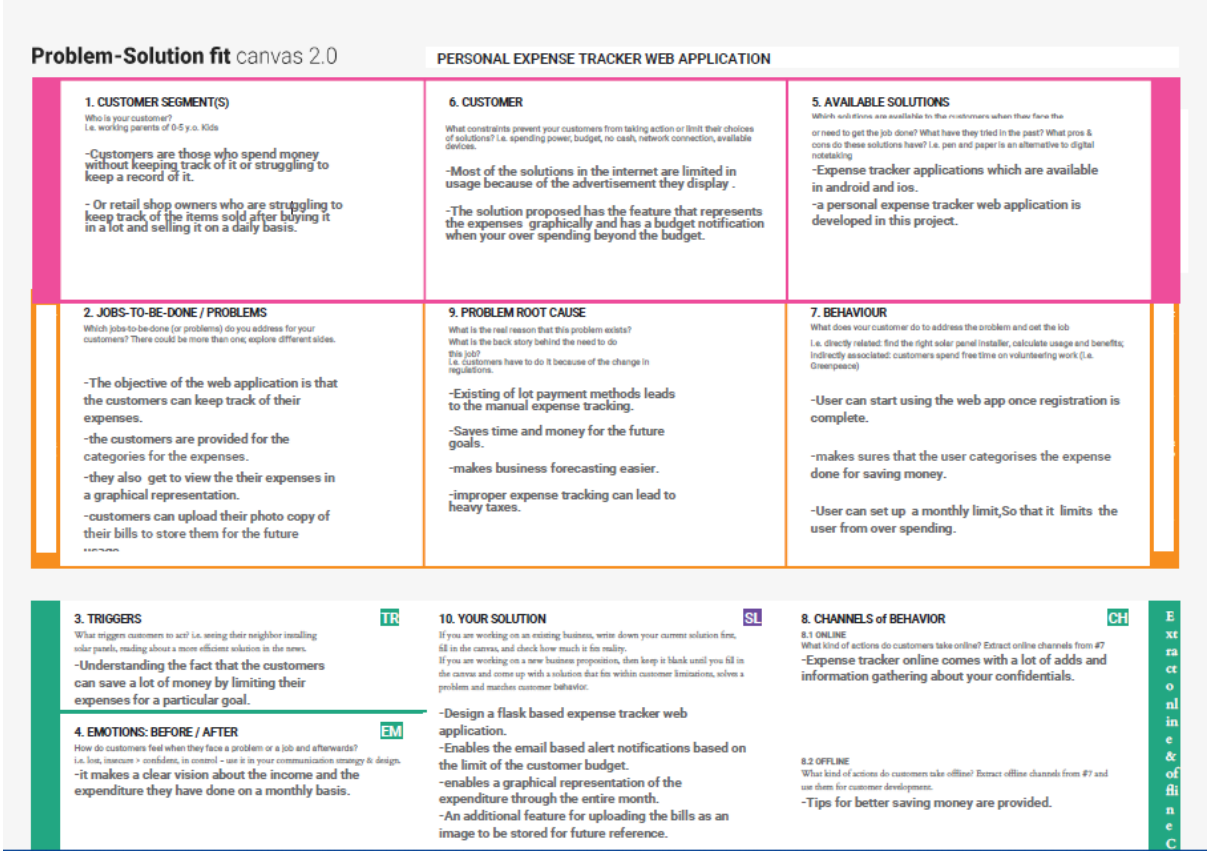
3.3 PROPOSED SOLUTION

S.NO.	Parameter	Description
1.	Problem Statement	In a paper-based expense tracker system it is difficult to track our monthly expenses manually. In a paper-based expense tracker system it is difficult to track our monthly expenses manually. The paper-based expense records may get lost in case of fire accidents, floods etc.
2.	Scalability of the Solution	This application can handle large numbers of users and data with high performance and security. This application can adapt for

		both large-scale and small-scale purposes. Easily available in all kinds of devices.
3.	Idea / Solution description	Daily expense management system which is specially designed for non-salaried and salaried personnel for keeping track of their daily expenditure in an easy and effective way through a computerized system which tends to eliminate manual paperwork. Personal finance applications will ask users to add their expenses and based on their expenses wallet balance will be updated which will be visible to the user. They have an option to set a limit for the amount to be used for that particular month if the limit is exceeded the user will be notified with an email alert.
4.	Novelty / Uniqueness	The user gets notified when their expense exceeds the limit and also it reminds the user when they forgot to make an entry. Tracking expenses through SMS. Data analytics on expenses. Future expense prediction
5.	Social Impact / Customer Satisfaction	The application should be able to generate reports of their spending and notify users if they have exceeded their budget. It is designed to be dynamic to produce the prediction. It also provides users' personal information, their income as well as their expenses. This application can create awareness among common people about finance and stuff. This application also helps users to be financially responsible. It

		Reduces time rather than entering details manually.
6.	Business Model (Revenue Model)	This Application is provided for free of cost. But It will have some advertisements. In premium version there is no advertisement and contains some additional features.

3.4 PROPOSED SOLUTION FIT



CHAPTER 4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

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

4.2 NON-FUNCTIONAL REQUIREMENTS

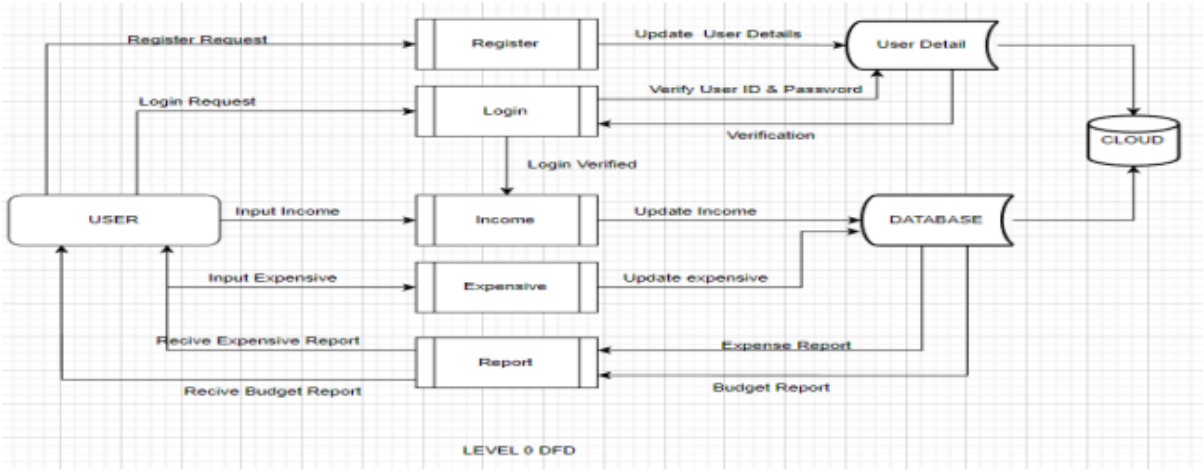
NFR. No.	Non-Functional Requirement	Sub Requirement
NFR-1	Usability	Helps to keep an accurate record of your income and expenses.
NFR-2	Security	The user's data is being stored in a secured way and the credentials are stored as hashed text.
NFR-3	Reliability	Each data record is stored on a well-built efficient database schema. There is no risk of data loss.

NFR-4	Performance	The types of expense are categories along with an option. Throughput of the system is increased due to light weight database support.
NFR-5	Availability	The application must have a 100% up-time.
NFR-6	Scalability	The ability to appropriately handle a user's demand.

CHAPTER 5
PROJECT DESIGN

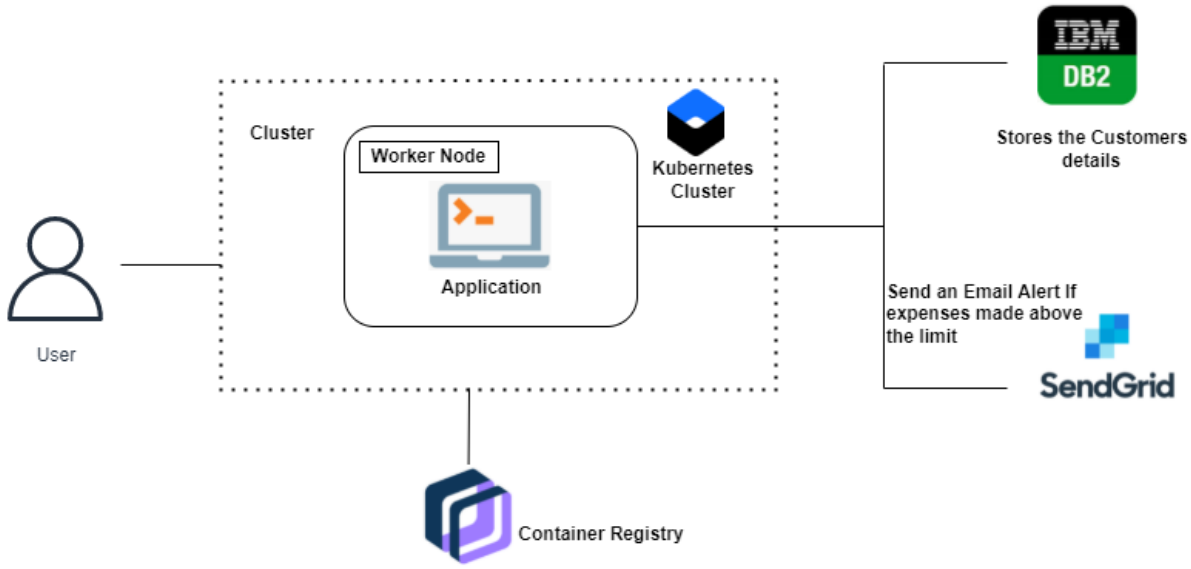
5.1 DATA FLOW DIAGRAMS

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 enters and leaves the system, what changes the information, and where data is stored.



5.2 TECHNICAL ARCHITECTURE

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2.



5.3 USER STORIES

A user story is the smallest unit of work in an agile framework. It's an end goal, not a feature, expressed from the software user's perspective. A user story is an informal, general explanation of a software feature written from the perspective of the end user or customer.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the	I can receive confirmation email &	High	Sprint-1
			application	click confirm		
Customer	Login	USN-1	As a user, I used my Mail id and password for login	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I forget my password. Used forget password ,	I got verification mail and changed my password	High	Sprint-2
Customer	Dashboard	USN-1	As a user, there is profile tab	Where I can update/edit my personal details	High	Sprint-3
		USN-2	As a user, there is budget tab	Where I can update/edit/set budget	High	Sprint-2
Customer	Profile	USN-1	As a user, I can change my phone no, mail, name	It get updated	Low	Sprint-2
	Budget	USN-1	As a user, I create a budget, update the budget.	It get create, update	High	Sprint-4
		USN-2	As a user, I can enter my expense into category	It get update to budget	High	Sprint-4
Customer	Report	USN-1	As a user, I get a expense report anytime I need	It show the report	High	Sprint-5
Customer	Logout	USN-1	As a user, I click on the logout button	It logout the user account	Low	Sprint-5
Customer care	Chat bot	ADMIN-1	As a admin, chat bot help to get familiar with application	It teach the user for the first time	High	Sprint-5

CHAPTER 6

PROJECT PLANNING AND SCHEDULING

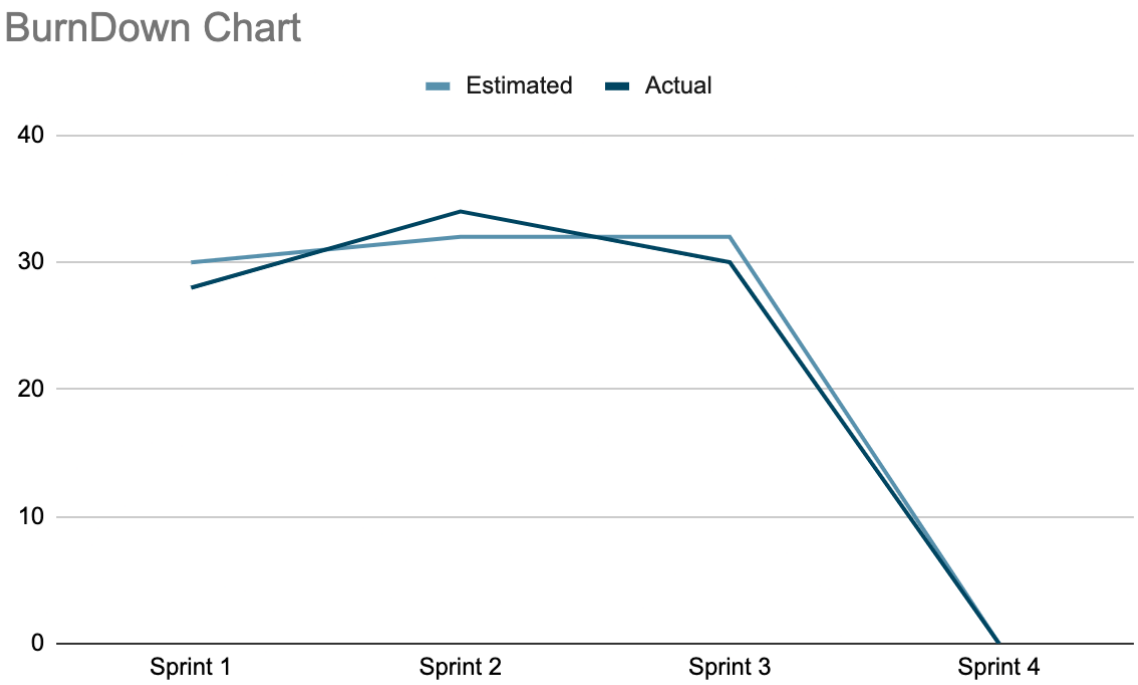
6.1 SPRINT PLANNING AND ESTIMATION

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.	10	High	Sivanesh K
Sprint-1	Login	USN-2	As a user, I can log into the application by entering email & password	10	High	Sri Raghav KR, Sivanesh K
Sprint-2	Add Expenditures	USN-3	As a user, I can add my expenditures.	20	High	Ayush HN
Sprint-3	Wallet	USN-4	As a user, I can check my wallet balance and other related wallet info	8	Medium	Sri Raghav KR
Sprint-3	Dashboard	USN-5	As a user, I can view my account details and other information like total spends, previous month spends, expenditure limit.	8	Medium	Tharun Kumar T, Ayush HN
Sprint-4	Expenditure limit warning email	USN-6	As a user, I receive an email when my expenditure limit has reached or wallet balance is about to be exhausted.	4	Low	Tharun Kumar T

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	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	16	6 Days	07 Nov 2022	12 Nov 2022	16	12 Nov 2022
Sprint-4	4	2 Days	14 Nov 2022	15 Nov 2022	4	15 Nov 2022

6.3 REPORTS FROM JIRA – BURNDOWN CHART



CHAPTER 7
CODING AND SOLUTION

7.1 FEATURE 1 – EXPENSE TRACKING

This feature focuses on providing the user with the facility to store their expenses in the platform. This way everyone can keep track of their spending and take informed decisions about their expenditure.

```
from flask import session
from modules.db import Db


class Expenses(Db):
    def __init__(self) -> None:
        super().__init__()

    def add(self, form):
        return self.insert(
            "expenses", [
                form['user_id'],
                form['title'],
                form['remarks'],
                form['category'],
                form['amount']
            ], ["user_id",
                "title",
                "remarks",
                "category",
                "amount"]
        )

    def get_expense(self, form):
        return self.get("expenses", "user_id = " + str(form["user_id"]) + " an
d title = '" + form["title"] + "' and amount = " + str(form["amount"]))

    def get_all(self, user_id):
        return self.getall("expenses", "user_id = " + user_id)

    def expense_category(self, id):
        return self.getall("expenses", "user_id = " + str(id) + " GROUP BY cat
egory", "category, Count(user_id)")
```

This is the code fragment handling the expenses. It is used to store and retrieve the expenses from the DB2 database.

7.2 FEATURE 2 – EXPENSE VISUALIZATION

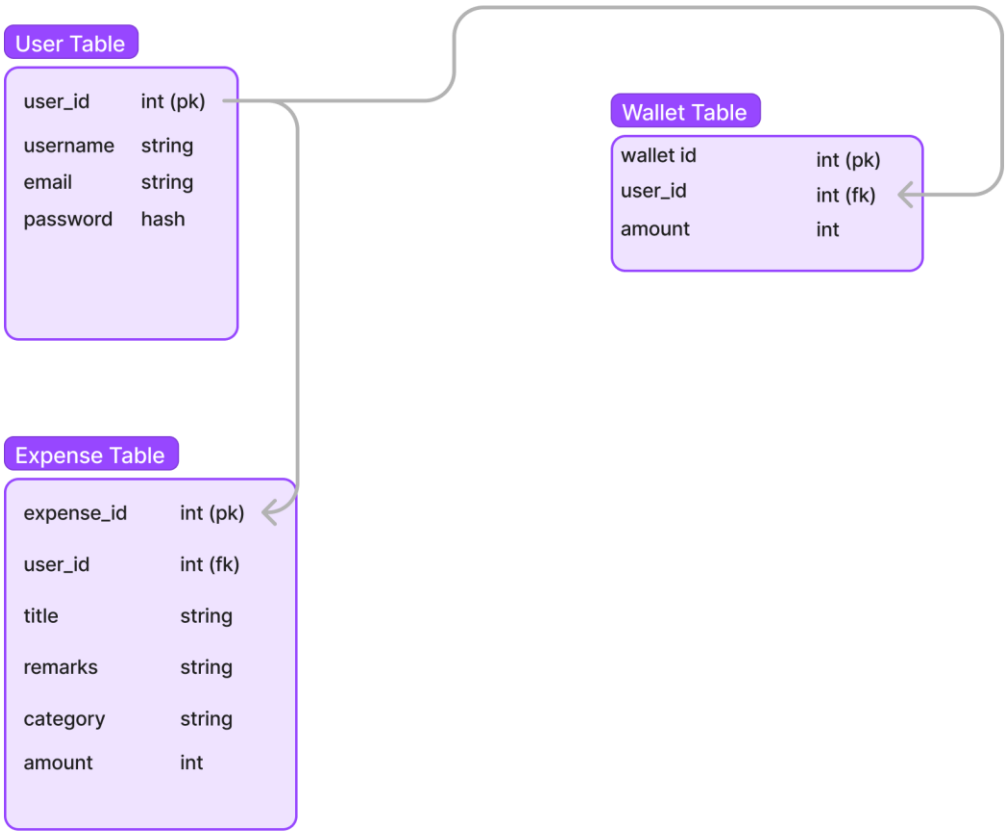
The users can find their expense report, in the form of a pie chart. A pie chart is a circular statistical graphic, which is divided into slices to illustrate numerical proportion. In a pie chart, the arc length of each slice (and consequently its central angle and area) is proportional to the quantity it represents. While it is named for its resemblance to a pie which has been sliced, there are variations on the way it can be presented. The earliest known pie chart is generally credited to William Playfair's Statistical Breviary of 1801.

This will give the users more insight than looking at the tabular data.

```
<PieChart data={expenses.category_spent} />
```

PieChart is component available in “react-minimal-pie-chart”. It builds the pie chart for the data being sent as a prop.

7.3 DATABASE SCHEMA



CHAPTER 8

TESTING

8.1 TEST CASES

Test case ID	Feature Type	Component	Test Scenario
HomePage_TC_001	UI	Login Page	Verify user is able to see the Login/Signup popup when user clicked on login/signup
LoginPage_TC_002	Funtional	login Page	Verify the UI elements in Login/Signup popup
Login Page_TC_003	UI	SignUP Page	Verify the UI elements in login popup
LoginPage_TC_OO4	Functional	Login page	Verify user is able to log into application with Valid credentials
LoginPage_TC_OO5	Functional	Login page	Verify user is able to log into application with InValid credentials
Add Wallet_TC_006	Functional	Wallet page	Verify user is able to add Wallet

Add Budget_TC_007	Funtional	budget page	Verify user is able to set budget
Add Transaction_TC_008	Funtional	Transaction feature	Verify user is able to add transaction

8.2 USER ACCEPTANCE TESTING

The purpose of this document is to briefly explain the test coverage and open issues of the Personal Expense Tracker project at the time of the release to User Acceptance Testing (UAT).

8.2.1 DETECT 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	2	0	1	1	4
Duplicate	1	1	0	0	2
External	0	0	0	0	0
Fixed	2	0	0	0	2
Not Reproduced	1	0	1	0	2
Skipped	0	0	0	1	1
Won't Fix	0	1	0	0	1
Totals	6	2	2	2	12

8.2.2 TEST CASE ANALYSIS

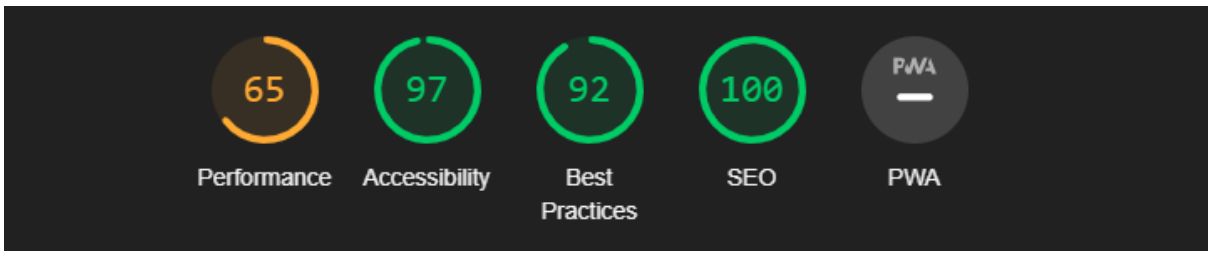
This report shows the number of test cases that have passed, failed, and untested.

Section	Total Cases	Not Tested	Fail	Pass
Client Application	15	0	0	15
Security	4	0	0	4
Exception Reporting	5	0	0	5
Final Report Output	3	0	0	3
Version Control	1	0	0	1

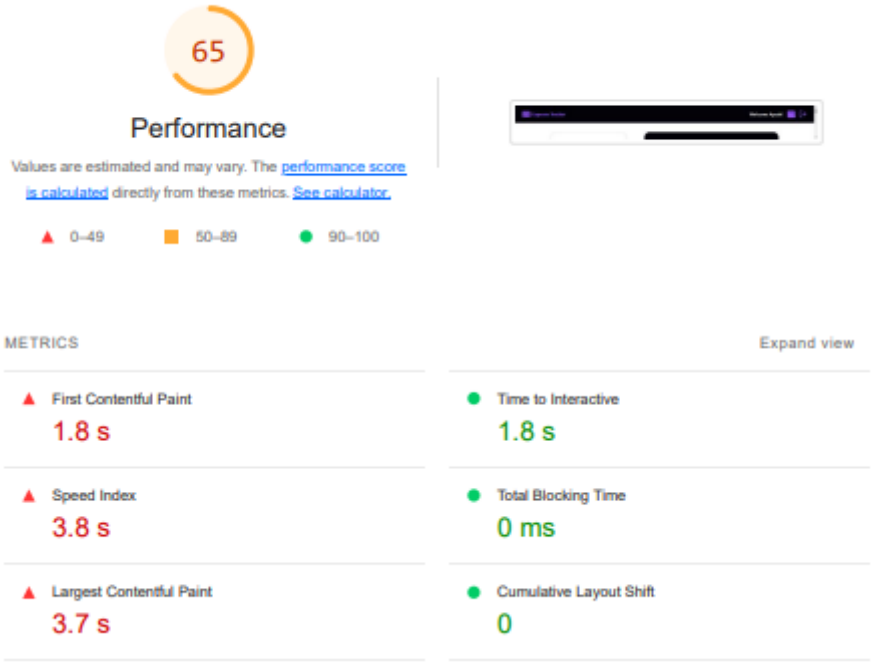
CHAPTER 9
RESULTS

9.1 PERFORMANCE METRICS

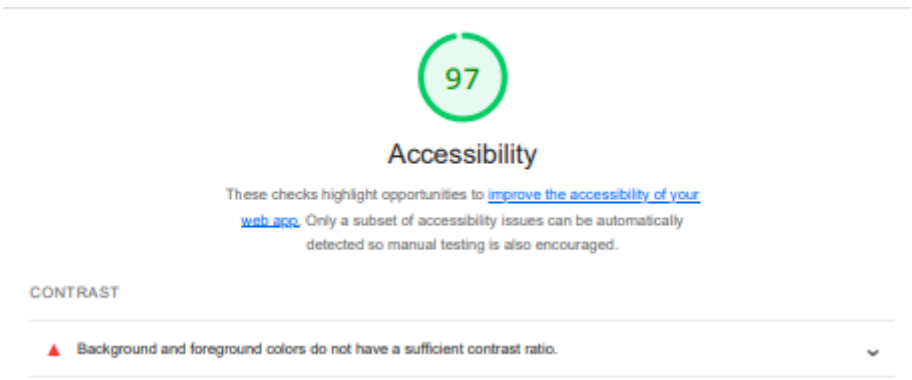
The performance of the client application was test using the Light House analysis. The analysis was done only for the desktop version of the application, since it is not responsive yet.



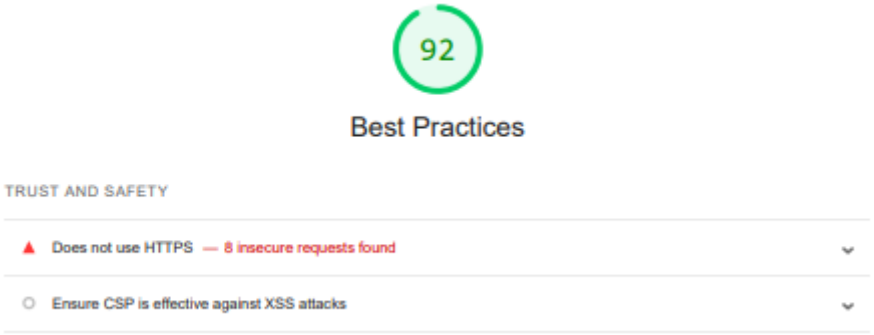
9.1.1 PERFORMANCE



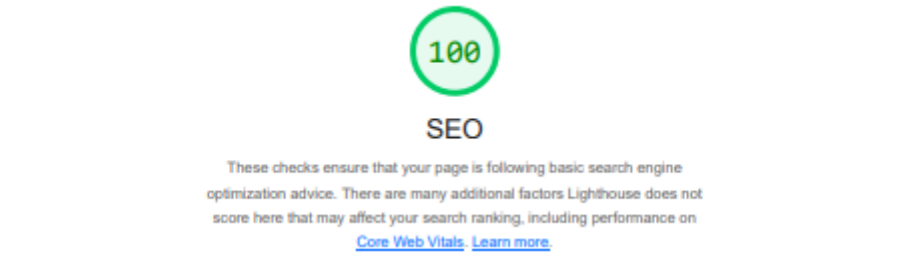
9.1.2 ACCESSIBILITY



9.1.3 BEST PRACTICES



9.1.4 SEARCH ENGINE OPTIMIZATION



CHAPTER 10
ADVANTAGES AND DISADVANTAGES

10.1 ADVANTAGES

- Easy to keep track of expenses
- Accessible on the go
- Simple to use
- Basic level insights available off the bat

10.2 DISADVANTAGES

- Requires manual entry of expenses
- Time consuming (at the moment)
- Lack of automation
- Lack of extensive analysis
- UI isn’t responsive, making it difficult to use in portable devices
- Inability to delete incorrect expenses

CHAPTER 11
CONCLUSION

Money has become a commodity which decides the way we live, in today's world. Hence, it has become important that we keep track of every single rupee we spend. We have created a web application which will allow users to have a clear understanding of their spending and figure where they spend the most. This will educate the user and help them to decide to whether cut down their expenses or not. The application has been deployed in the IBM cloud successfully.

CHAPTER 12
FUTURE SCOPE

At the moment, the approach of the application is grounded. In the future, we could implement some techniques, wherein the user can just scan the bills of their expense, to record their expenses. We could also connect all the UPI and bank accounts of the users to record all the expense to be in sync with all the expenditure.

This application can also be extended as a digital vault to the users, where they can store their bills. For example, store the bill of a TV they bought until the warranty expires. This will save the users from having to maintain the physical copies of the bills, which is quite tiresome and unorganized.

CHAPTER 13

APPENDIX

13.1 SOURCE CODE

app.py:

```
from flask import Flask, session, send_from_directory
from flask_session import Session
from flask_cors import CORS, cross_origin

from routes.users import userBp as user_blueprint
from routes.wallets import wallet_blueprint
from routes.expenses import expenses_blueprint

from dotenv import load_dotenv

import os

load_dotenv()
app = Flask(__name__, static_folder="./client/build")
app.secret_key = os.environ["APP_SECRET_KEY"]
app.config["SESSION_PERMANENT"] = False
app.config["SESSION_TYPE"] = "filesystem"

Session(app)

cors = CORS(app)
app.config['CORS_HEADERS'] = 'Content-Type'

app.register_blueprint(user_blueprint)
app.register_blueprint(wallet_blueprint)
app.register_blueprint(expenses_blueprint)


@app.route("/api/session")
def flask_session():
    if session.get("active") == None:
        return {"success": True}
    else:
        return {"success": False}


@app.route("/api/logout")
def logout():
    session.clear()
```

```
    return {"success": True, "message": "Logged out successfully"}


@app.route('/', defaults={'path': ''})
@app.route('/<path:path>')
def serve(path):
    if path != "" and os.path.exists(app.static_folder + '/' + path):
        return send_from_directory(app.static_folder, path)
    else:
        return send_from_directory(app.static_folder, 'index.html')


if __name__ == '__main__':
    # app.run("127.0.0.1", 8000, debug=True)
    app.run("0.0.0.0")
```

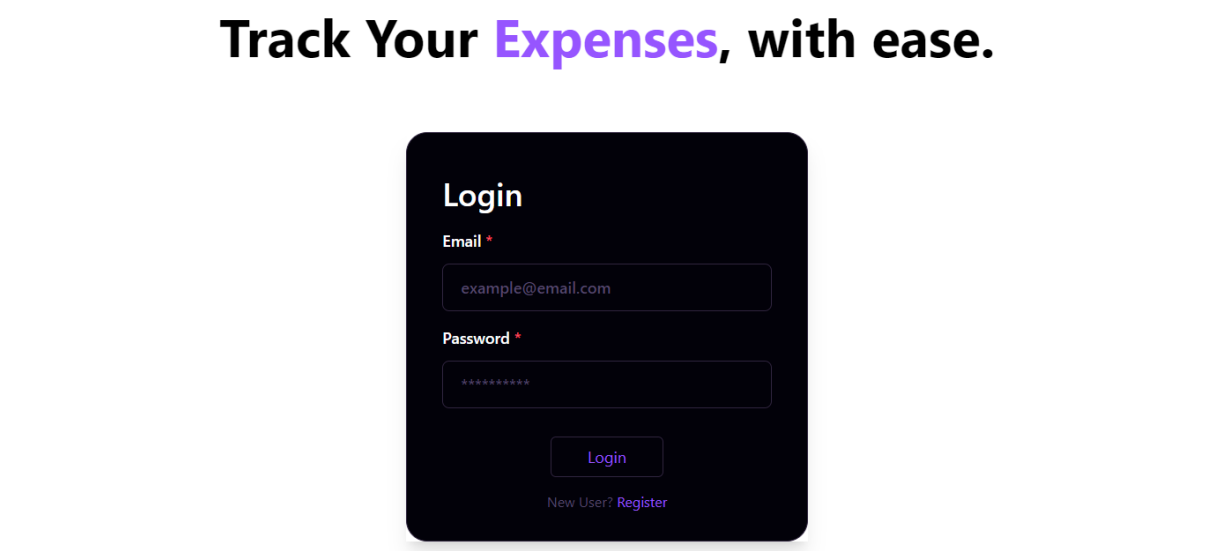
routes/expenses.py:

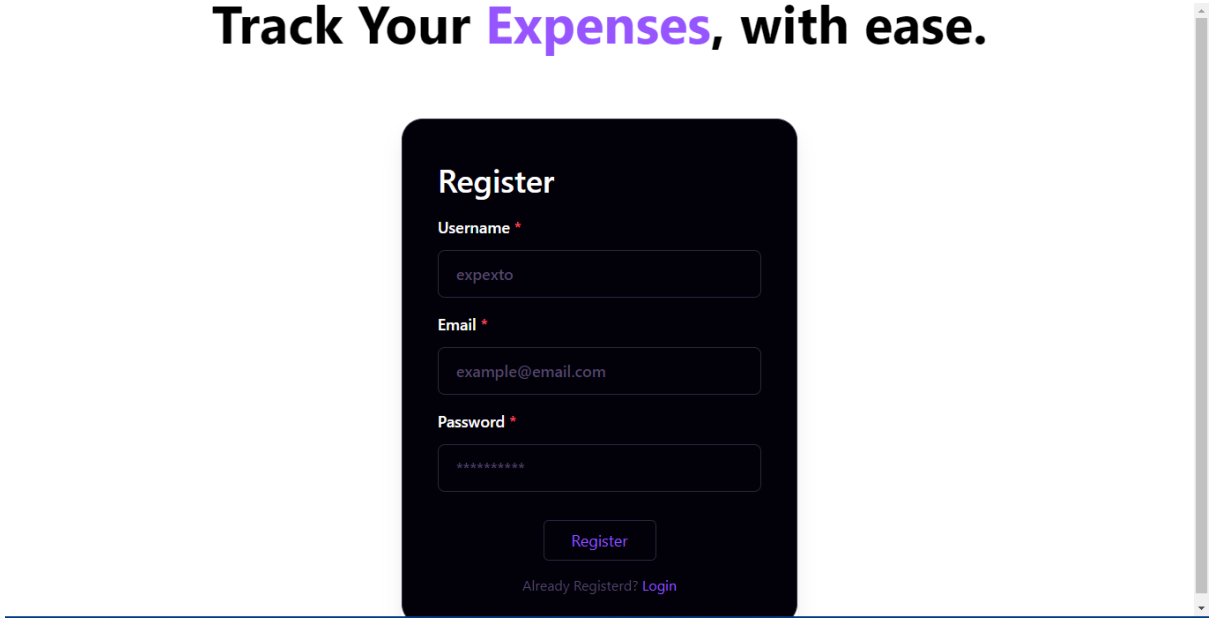
```
from flask import Blueprint, request
from modules.expenses import Expenses
from modules.user import User
from modules.wallets import Wallets

expenses_blueprint = Blueprint('expenses_blueprint', __name__)


@expenses_blueprint.route('/api/expenses', methods=['POST'])
def add():
    user = User().getUser("id", request.json['user_id'])
    if user:
        wallet = Wallets().getWallet(request.json["user_id"])
        Wallets().update_threshold(
            {"user_id": request.json["user_id"], "threshold": str(int(wallet[2
]) - int(request.json["amount"])), "amount": str(wallet[3])})
        Expenses().add(request.json)
        expense = Expenses().get_expense(request.json)
        print(expense)
        return {
            "success": True,
            "message": "Expense added successfully",
            "data": {"expenses": {"id": expense[0], "user_id": expense[1], "ti
tle": expense[2], "remarks": expense[3], "category": expense[4], "amount": exp
ense[5]}}
        }
    else:
        return {"success": False, "message": "Invalid User"}
```

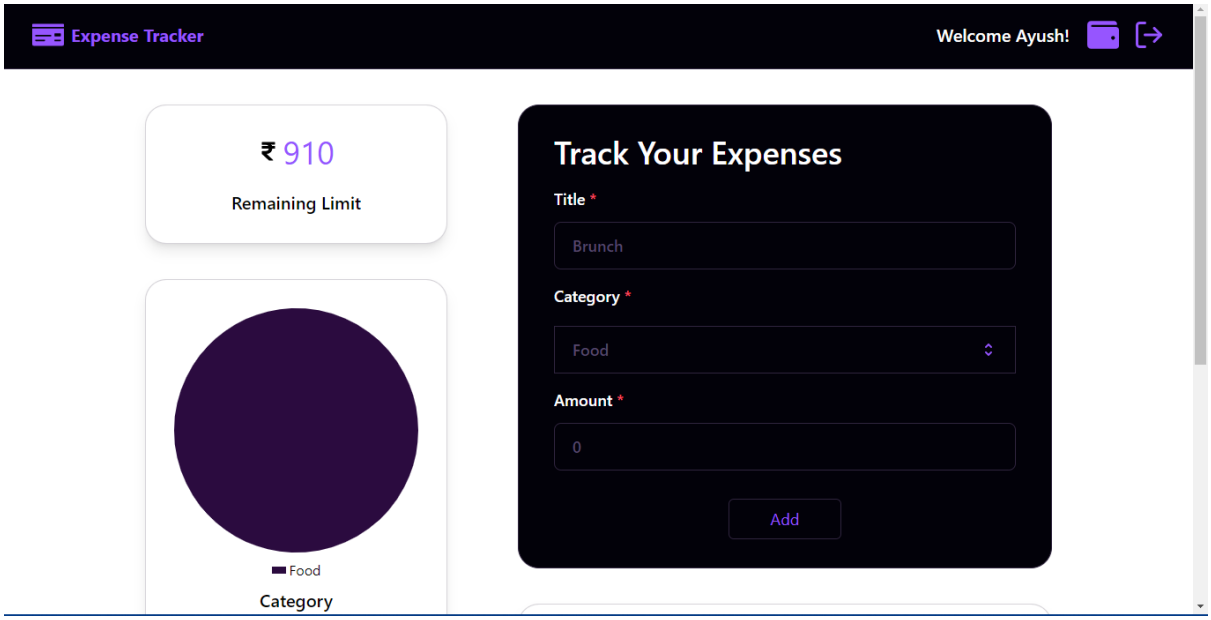
```python
@expenses_blueprint.route('/api/expenses/<user_id>', methods=['GET'])
def get_all(user_id):
    user = User().getUser("id", user_id)
    if user:
        expenses = Expenses().get_all(user_id)
        return {
            "success": True,
            "message": "Expense fetched successfully",
            "data": {"expenses": expenses}
        }
    else:
        return {"success": False, "message": "Invalid User"}


@expenses_blueprint.route('/api/expenses/category/<user_id>', methods=['GET'])
def get_category(user_id):
    user = User().getUser("id", user_id)
    if user:
        expenses = Expenses().expense_category(user_id)
        return {
            "success": True,
            "message": "Expense fetched successfully",
            "data": {"expenses": expenses}
        }
    else:
        return {"success": False, "message": "Invalid User"}
```

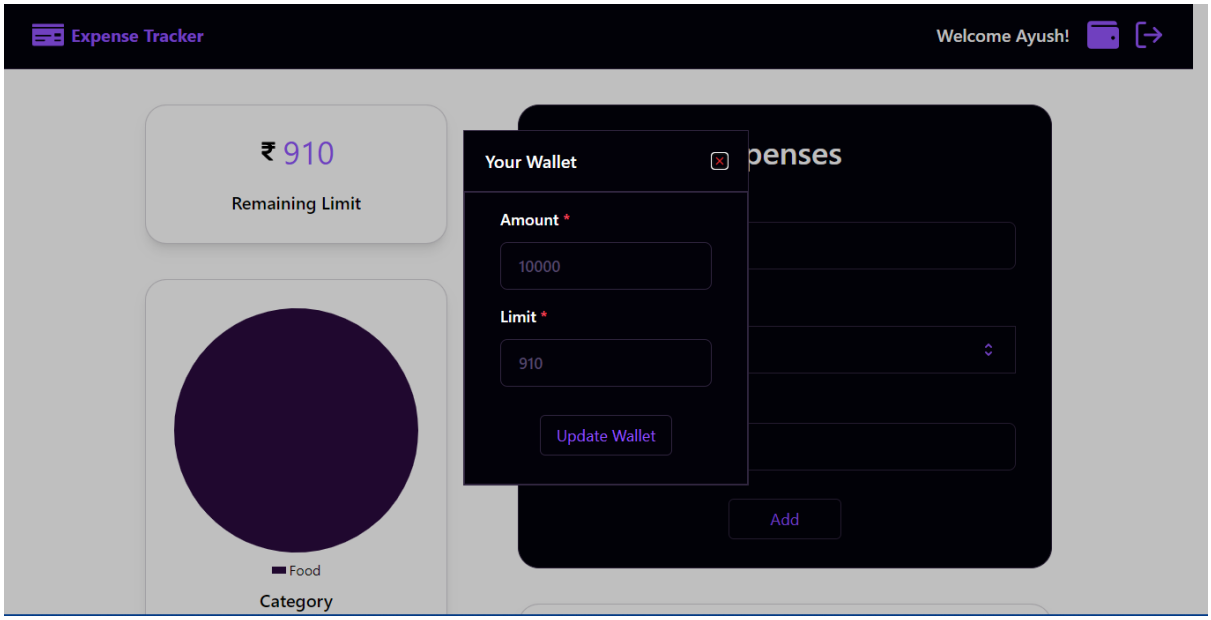
Complete Code available at: https://github.com/IBM-EPBL/IBM-Project-23371-1659881012/tree/main/Final%20Deliverables/Final%20Code

13.2 OUTPUT


Track Your Expenses, with ease.
Login
Email *
example@email.com
Password *
****
Login
New User? Register


Track Your Expenses, with ease.
Register
Username *
expexto
Email *
example@email.com
Password *
****
Register
Already Registerd? Login


Expense Tracker
Welcome Ayush!
₹ 910
Remaining Limit
Food
Category
Track Your Expenses
Title *
Brunch
Category *
Food
Amount *
0
Add


Expense Tracker
Welcome Ayush!
₹ 910
Remaining Limit
Food
Category
Your Wallet
Amount *
10000
Limit *
910
Update Wallet
penses
Add

13.3 LINKS

- GitHub Repository: https://github.com/IBM-EPBL/IBM-Project-23371-1659881012
- Demo Video: https://bit.ly/xpense-tracker-demo-video-1