PERSONAL EXPENSE TRACKER

PROJECT REPORT

Submitted by

PRANESH S (19EUCS107)

RAJEEV CHANDRAN S (19EUCS112)

POOVARASAN K (19EUCS106)

PRAVEEN KUMAR S P (19EUCS110)

in partial fulfillment of the requirements for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY COIMBATORE

(An Autonomous Institution)

ANNA UNIVERSITY: CHENNAI

MAY 2022

SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institution)
(Approved by AICTE and Affiliated to Anna University, Chennai)
ACCREDITED BY NAAC WITH "A" GRADE

BONAFIDE CERTIFICATE

Certified that this project report titled **Personal Expense Tracker"** is the bonafide work of **PRANESH S (19EUCS107), RAJEEV CHANDRAN S (19EUCS107), POOVARASAN K (19EUCS106), PRAVEEN KUMAR S P (19EUCS110)** who carried out the project work under my supervision.

SIGNATURE   SIGNATURE

Dr.K. SASI KALA RANI, M.E., Ph.D.,   **MS. POORANAM N**

HEAD OF THE DEPARTMENT   **SUPERVISOR**

Department of Computer Science and Engineering

Sri Krishna College of Engineering and Technology

Kuniamuthur,

Coimbatore

This project report is submitted for the Autonomous Project Viva-Voce examination held on …………......

INTERNAL EXAMINER   **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

TABLE OF CONTENTS

CHAPTER 1

1. INTRODUCTION:

In today's busy and expensive lives we are in a great rush to make money. But at the end of the day we broke off. As we are unknowingly spending money on little and unwanted things. So, we have come over with the idea to track our earnings. Daily Expense Tracker (DET) aims to help everyone who are planning to know their expenses and save from it. DTE is a website in which user can add expenses on daily basis and its table will get generated and at the end based on user expenses report will be generated. User can select date range to calculate his/her expenses come over with the idea to track our earnings. Personal Expense Tracker aims to help everyone who are planning to know their expenses and save from it. Personal Expense Tracker is a website in which user can add expenses on daily basis and at the end, based on user expenses report will be generated. User can select date range to calculate his/her expenses.

1.1 Project Overview :

This website is used to track expenses and control spending beyond limits. while input data of expenses in website, we must select category which spent on and additionally notes can be used to note the details of expenses. By entering those record we can track our expenses. we can generate reports in graphical, pie chat. We can also set limits to particular category which alerts in email when the limits exceed.

1.2 Purpose :

At end of certain period, users does not know where they spent their money and they spend more on needless expenses beyond budgets which leads to financial crisis. To avoid this people needs to track their expenses. While calculating in diary requires lot of manual calculation and lot of time. This is the purpose to go for website application to track expenses.

CHAPTER 2

LITERATURE SURVEY

2.1 Existing problem :

People can't able to track their expenses and spending more on unnecessary expenses which leads to money crisis. Without tracking people can't know whether they exceed the limit of their budget. Diary notes requires lots of manual calculation and It reduces the interest to track expenses. User frustrated about they can't remember where their money goes and can't handle their cash flow. There is no alerting system about exceeding limits.

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

2.2 References:

1. Adamson, I., Chan, K.-M., & Handford, D. (2003). Relationship marketing: Customer commitment and trust as a strategy for the smaller Hong Kong corporate banking. *International Journal of Bank Marketing, 21*(6/7), 347–358.

2. Adiwijaya, K., Wahyuni, S., & Mussry, J. S. (2017). Marketing ambidexterity and marketing performance: Synthesis, a conceptual framework, and research propositions. In *Enhancing Business Stability Through Collaboration* 431–442. CRC Press.

3. Afèche, P., Araghi, M., & Baron, O. (2017). Customer acquisition, retention, and service access quality: Optimal advertising, capacity level, and capacity allocation. *Manufacturing & Service Operations Management, 19*(4), 674–691.

4. Aggarval, C. C., & Yu, P. S. (2002). Finding localized associations in market basket data. *IEEE Transactions on Knowledge and Data Engineering, 14*, 51–62.

5. Alhakimi, W., & Ghaleb, A. (2019). The impact of CRM components system on customer retention in the telecom industry: A case of Y-Telecom in Yemen. *Middle East Journal of Management, 6*(4), 378–409.

6. Al-Omoush, K. S., Simón-Moya, V., Atwah Al-ma'aitah, M., & Sendra-García, J. (2021). The determinants of social CRM entrepreneurship: An institutional perspective. *Journal of Business Research, 132*, 21–31.

7. Canhoto, A. I., Meadows, M., Ball, K., Daniel, E., Dibb, S., & Spiller, K. (2017). The role of customer management capabilities in public–private partnerships. *Journal of Strategic Marketing, 25*(5–6), 384–404.

8. Chang, C. W., & Zhang, J. Z. (2016). The effects of channel experiences and direct marketing on customer retention in multichannel settings. *Journal of Interactive Marketing, 36*, 77–90.

9. Chen, C., Geng, L., & Zhou, S. (2020). Design and implementation of bank CRM system based on decision tree algorithm. *Neural Computing and Applications*, 1–11.

10. Cricelli, L., Famulari, F. M., Greco, M., & Grimaldi, M. (2020). Searching for the one: Customer relationship management software selection. *Journal of Multi-Criteria Decision Analysis*.

11. Drew, J. H., Mani, D. R., Betz, A. L., & Datta, P. (2001). Targeting customers with statistical and data-mining techniques. *Journal of Service Research, 3*, 205–220.

12. Fidel, P., Schlesinger, W., & Cervera, A. (2015). Collaborating to innovate: Effects on customer knowledge management and performance. *Journal of Business Research, 68*(7), 1426–1428.

13. Foltean, F. S., Trif, S. M., & Tuleu, D. L. (2019). Customer relationship management capabilities and social media technology use: Consequences on firm performance. *Journal of Business Research, 104*, 563–575.

14. Fu, H. P., & Chang, T. S. (2016). An analysis of the factors affecting the adoption of cloud consumer relationship management in the machinery industry in Taiwan. *Information Development, 32*(5), 1741–1756.

15. Ngai, E. W. (2005). Customer relationship management research (1992–2002) An academic literature review and classification. *Marketing Intelligence & Planning, 23*(6), 582–605.

2.3 Problem Statement Definition :

Our project helps the user to keep track their expenses and determine whether they are spending as per their set budget. Potential users need to input the required data such as the expense amount, merchant, category, and date when the expense was made. Which allows users to track their expenses daily, weekly, monthly, and yearly in terms of summary, bar graphs, and pie-charts. It is like automated diary which requires no burden of manual calculation and enables the user to not just keep the control on the expenses but also to generate and save reports. Users can insert and delete transactions. We can compare with past expenses. Customized email alerts are used alerts user when limit exceeds.

CHAPTER 3

IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

3.2 IDEATION & BRAINSTORMING :

3.3 PROPOSED SOLUTION :

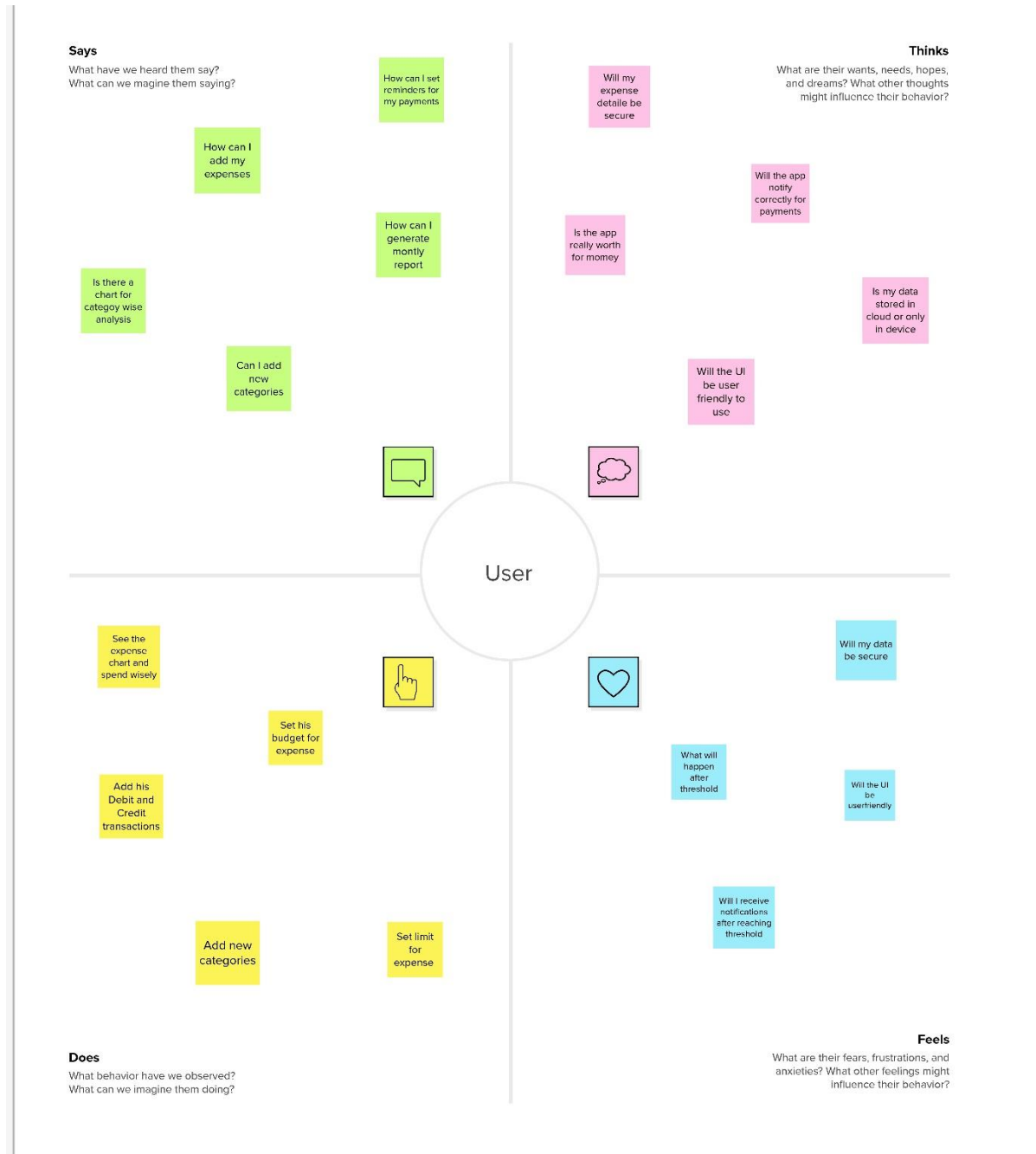| S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	People can't able to track their expenses and spending more on unnecessary expenses which leads to money crisis. People forget to pay dues on time, sometimes this leads to fine. Diary notes requires lots of manual calculation and It reduces the interest to track expenses.
2.	Idea / Solution description	Our project helps the user to keep track their expenses and determine whether they are spending as per their set budget. Potential users need to input the required data such as the expense amount, merchant, category, and date when the expense was made. Which allows users to track their expenses daily, weekly, monthly, and yearly in terms of summary, bar graphs, and pie-charts. User forgotten to input records can be avoided by remainders and alerts are helps to pay dues on time. It is like automated diary which

		requires no burden of manual calculation and enables the user to not just keep the control on the expenses but also to generate and save reports. Users can insert and delete transactions. We can compare with past expenses.
3.	Novelty / Uniqueness	We can set budgets for particular category to track unwanted expenses. we can generate reports as pdf for specific category. Budget setting feature leads people to overconsume some goods, under consume others and control over spending beyond limits.
4.	Social Impact / Customer Satisfaction	This solution controls users on overspending and reduces money crisis due to unwanted expenses. As this tracking expense becomes a habit, people can get a good picture of how much money they need to maintain their lifestyle. Tracking helps people to feel confidence on finance.
5.	Business Model (Revenue Model)	Revenue can be generated by placing advertisement.
6.	Scalability of the Solution	A Future update shall have payment option were we can pay dues and subscription. Linking Bank accounts and also tracking shares. It can be scaled for all types of people from any type of field.

3.4 Problem Solution fit :

The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer's problem. It helps entrepreneurs, marketers and corporate innovators identify behavioral patterns and recognize what would work and why

Purpose:

- ❑ Solve complex problems in a way that fits the state of your customers.
- ❑ Succeed faster and increase your solution adoption by tapping into existing mediums and channels of behavior.

- ❑ Sharpen your communication and marketing strategy with the right triggers and messaging.
- ❑ Increase touch-points with your company by finding the right problem-behavior fit and building trust by solving frequent annoyances, or urgent or costly problems.
- ❑ **Understand the existing situation in order to improve it for your target group.**

Template:

Project Title: Personal Expense Tracker Application | Project Design Phase-I - Solution Fit | Team ID: PNT2022TMID02774

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) `CS`
Who is your customer?
i.e. working parents of 0-5 y.o. kids

- a employee
- a student
- function organizer
- a shop keeper

6. CUSTOMER CONSTRAINTS `CC`
What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

Users had no time to record manually and most of the time they forget to note records on time and had burden of remembering those records, which reduces the interest to track.

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.e. pen and paper is an alternative to digital notetaking

- Diary notes ,we can't able to carry all the time and includes lots of calculations.
- Mobile notes, which includes burden of manual calculations.
- Notes using Excel, we can't add descriptions adequate and we can't track category wise.

Explore AS, differentiate

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

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.

Track their expenses, remainder to pay dues on time and avoid spending more on unnecessary expenses and reduce burden on manual calculations.

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.e. customers have to do it because of the change in regulations.

- track expenses to know were money was spent
- control spending more on unwanted expenses
- reduce expenses and improve in finance.
- avoid financial crisis.

7. BEHAVIOUR `BE`
What does your customer do to address the problem and get the job done?
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)

- User records the expenses daily manually
- setting alarms to avoid forgot to record expenses.
- read blogs to learn best expense tracking ways.
- searching ideas in internet to control expenses to maintain their lifestyle.

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

Identify strong TR & EM

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.

At end of certain period, users doesn't know were they spent their money and they spend more on needless expenses beyond budgets which leads to financial crisis.

4. EMOTIONS: BEFORE / AFTER `EM`
How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

Before using the tool, users frustrated about they can't remember where their money goes and can't handle their cash flow. After the tool, this tracking expense becomes a habit, people can get a good picture of how much money they need to maintain their lifestyle.

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.

Our project helps the user to track their expenses and determine whether they are spending as per their set budget. Potential users need to input the required data such as the expense amount, category and date when the expense was made. Which allows users to track their expenses daily, weekly, monthly and yearly in terms of summary, bar graphs, and pie-charts.
User forgotten to input records can be avoided by remainders and alerts are helps to pay dues It is like automated diary which requires no burden of manual calculations and enables the user to not just keep the control on the expenses but also to generate and save reports. We can compare with past expenses.

8. CHANNELS of BEHAVIOUR `CH`
8.1 ONLINE
What kind of actions do customers take online? Extract online channels from #7

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

Online :
- learn best expense tracking ways by reading blogs in internet
- searching ideas in internet to control expenses and setting budgets.

Offline :
- record expenses daily in diary, mobile notes or in excel sheets.
- setting alarms to pay dues and to avoid forgot recording expenses.

Identify strong TR & EM

CHAPTER 4

REQUIREMENT ANALYSIS

4.1Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Be aware of daily expenditures	*Enter amount spent*
FR-2	Generate visually appealing charts	*Notify users periodically to update their expenses*
FR-3	Categorize credit and debit transactions	*Always looks for credit/debit threshold*
FR-4	Prompt to not exceed the threshold amount	*Send email alerts if the user is on the verge of exceeding the threshold*
FR-5	Show ways to minimize expense in the most spent area	*Constantly look for patterns from previous expenses to improve accuracy*

4.2 <u>Non-functional Requirements:</u>

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

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	*The UI/UX must be visually appealing and pleasing to the senses with proper placements of primitive elements.*
NFR-2	**Security**	*Completely safe and private as user's data is neither shared nor utilized for any other secondary purposes.*
NFR-3	**Reliability**	*The application is guaranteed to give non-erroneous results at most instances.*
NFR-4	**Performance**	*The application is entirely robust to handle the incoming traffic even if there occurs an unexpected surge.*
NFR-5	**Availability**	*The application does not fail to keep track of the expenses that have been entered*

CHAPTER 5

PROJECT DESIGN :

5.1 Data Flow Diagrams:

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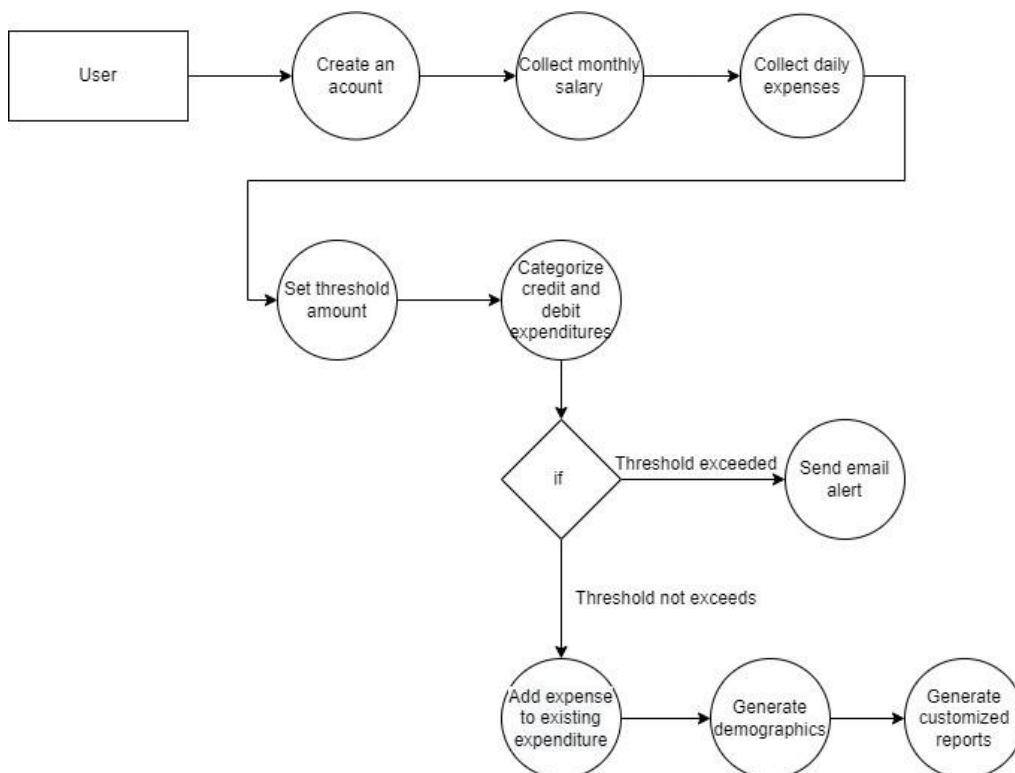
5.2 Solution Architecture and Technical Architecture :

Solution Architecture :

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.

- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.

- Define features, development phases, and solution requirements.

- Provide specifications according to which the solution is defined, managed, and delivered.

Example - Solution Architecture Diagram:
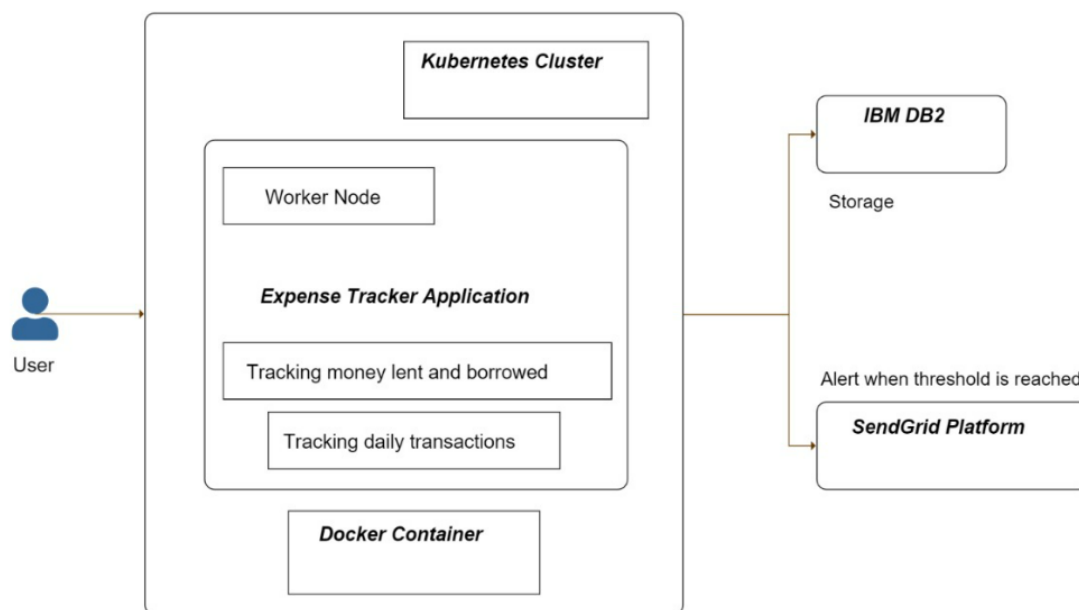
TECHNICAL ARCHITECTURE:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2
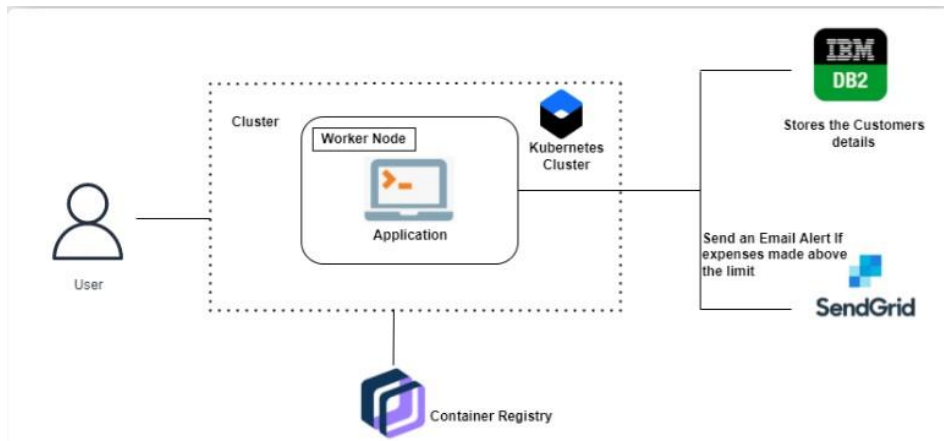
Table-1 : Components & Technologies:

S.No	Component	Technology
1.	User Interface	HTML
2.	Application Logic-1	Python
3.	Application Logic-2	IBM DB2
4.	Microservice	SendGrid

Table-2: Application Characteristics:

S.No	Characteristics	Technology
1.	Open-Source Frameworks	Flask
2.	Performance	It can handle about 100 requests per second

5.3 User Stories :

Use the below template to list all the user stories for the product.

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, mobile number, weekly expense, montly salary	I can access my account / dashboard	High	Sprint-1
	Login	USN-2	As a user, I can log into the application by entering email & password	I can access my account / dashboard	High	Sprint-1
	Landing page		As a user, I can view my entire expenses throughout a particular period of time	I can view my expenses	High	Sprint-1
			As a user, I can generate reports based on my previous expenditures.	Report is successfully generated	Medium	Sprint-2
			As a user, I can logout	Successfully logout	High	Sprint-1
			As a user, I can create expense	Expense is successfully added	High	Sprint-1
			As a user, I can edit ,delete, update expense	The corresponding action is made to the expense	High	Sprint-1
			As a user, I can view credit and	The expenses are filtered accordingly	Medium	Sprint-2

User Type	Functional Requirement (Epic)	User Story Number	User Story /Task	Acceptance criteria	Priority	Release
			debit expenses separately.			
			As a user, I can set a minimum threshold for my total expenditure either each week or month.	Minimum threshold is set successfully	High	Sprint-1
			As a user, I can view graphically interpreted insights of my expenditures.	Demographics of the expenses are generated	High	Sprint-1
			As a user, I can be aware of the expense that I spend the most on	Know my weak points that prevents user from saving more	Low	Sprint-3

CHAPTER 6

PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation :

Product Backlog, Sprint Schedule, and Estimation

Use the below template to create product backlog and sprint schedule

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, mobile number, weekly expense, montly salary	5	High	Poovarasan
Sprint-1	Login	USN-2	As a user, I can log into the application by entering email & password	2	High	Praveen Kumar
Sprint-1	Landing page	USN-3	As a user, I can view my entire expenses throughout a particular period of time	13	High	Pranesh
Sprint-2		USN-4	As a user, I can generate reports based on my previous expenditures	2	Medium	Pranesh
Sprint-4	Logout	USN-5	As a user, I can logout	2	High	Praveen Kuamr
Sprint-2	Dashboard	USN-6	As a user, I can create expense	5	Medium	Rajeev Chandran
Sprint-		USN-7	As a user, I can	13	High	Rajeev

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
2			edit ,delete, update expense			Chandran
Sprint-3		USN-8	As a user, I can view credit and debit expenses separately.	13	High	Pranesh
Sprint-3		USN-9	As a user, I can set a minimum threshold for my total expenditure either each week or month.	2	Low	Rajeev Chandran
Sprint-3		USN-10	As a user, I can view graphically interpreted insights of my expenditures	5	High	Poovarasan
Sprint-4		USN-11	As a user, I can be aware of the expense that I spend the most on	8	High	Poovarasan
Sprint-4		USN-12	As a user, I can be able to update my set monthly limit	5		
Sprint-4		USN-13	As a user, I can able to view my profile	5		

6.2 Sprint Delivery Schedule :

Project Tracker, Velocity & Burndown Chart:

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

Velocity:

Velocity is a metric that predicts how much work an Agile software development team can successfully complete within a two-week sprint (or similar time-boxed period). Velocity is a useful planning tool for estimating how fast work can be completed and how long it will take to complete a project

Average velocity = Total story points/ No. of iterations = 80/4 = 20

Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.
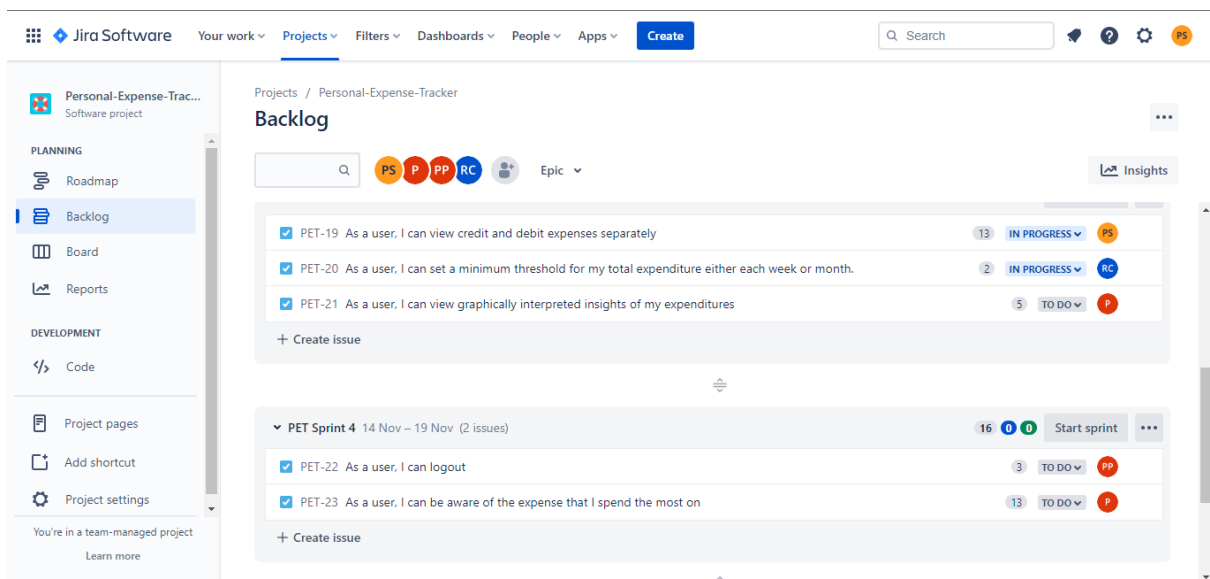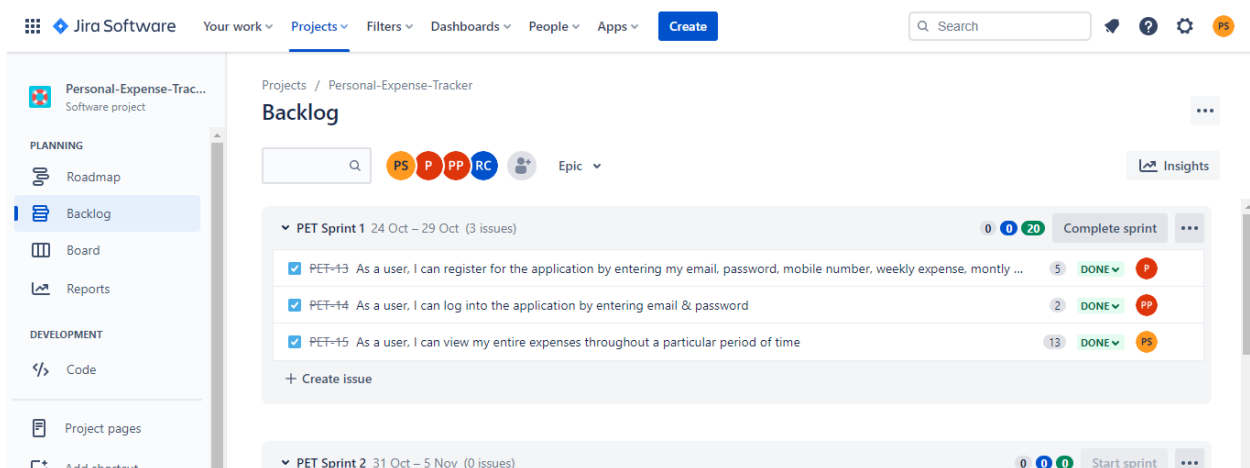
Sprint burndown

8 points done, 12 points to go

⚠ Heads up

6.3 Reports from JIRA :

CHAPTER 7

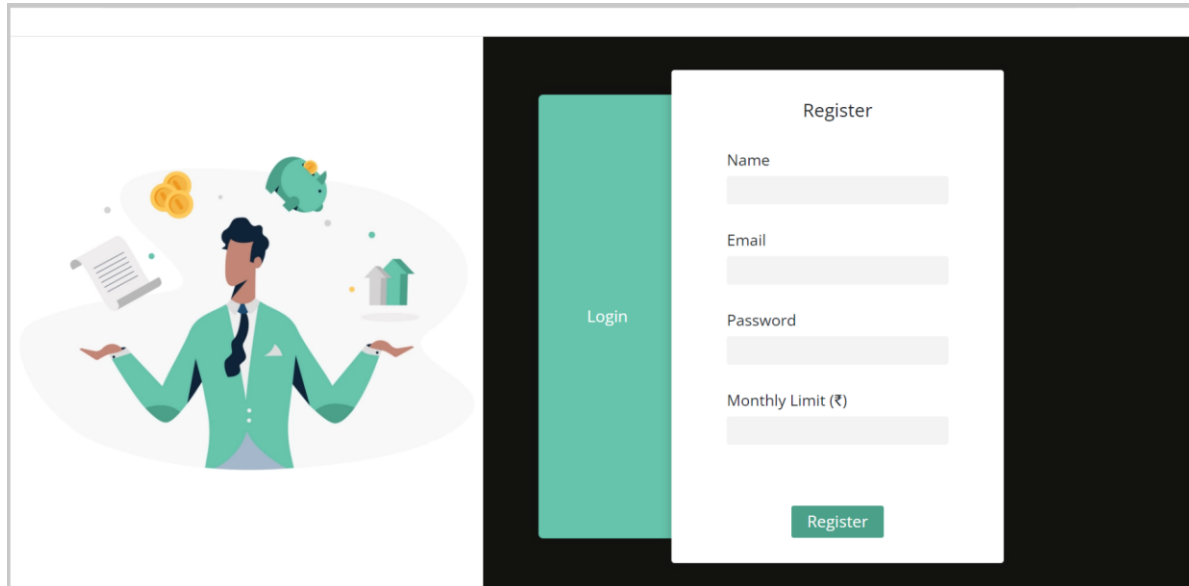# CODING AND SOLUTIONING
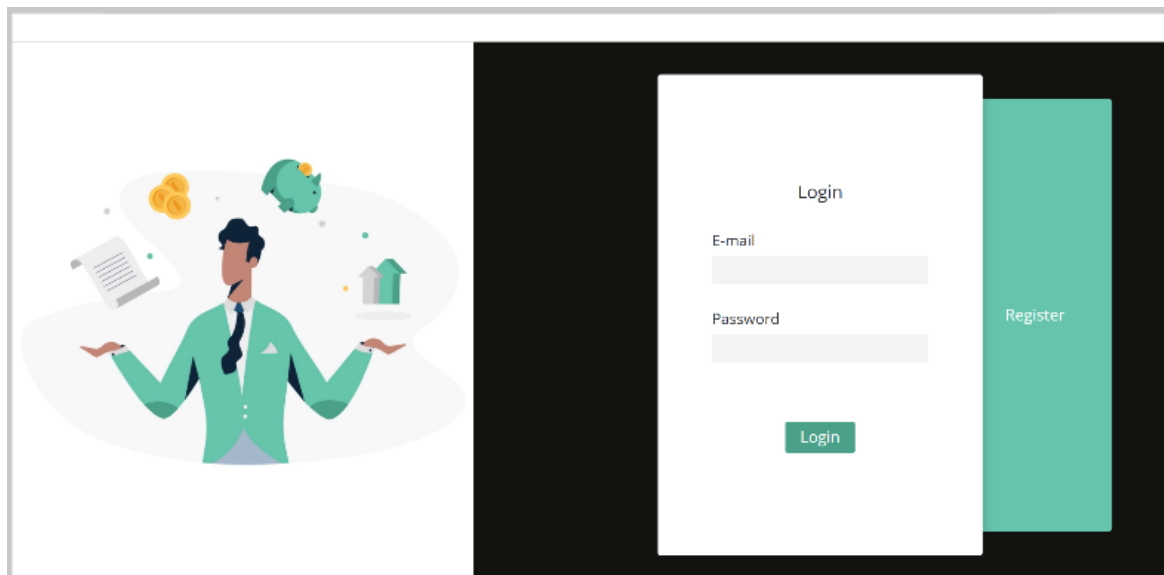
7.1. Feature 1:

Figure 7.1 – Register page

7.2 Feature 2:
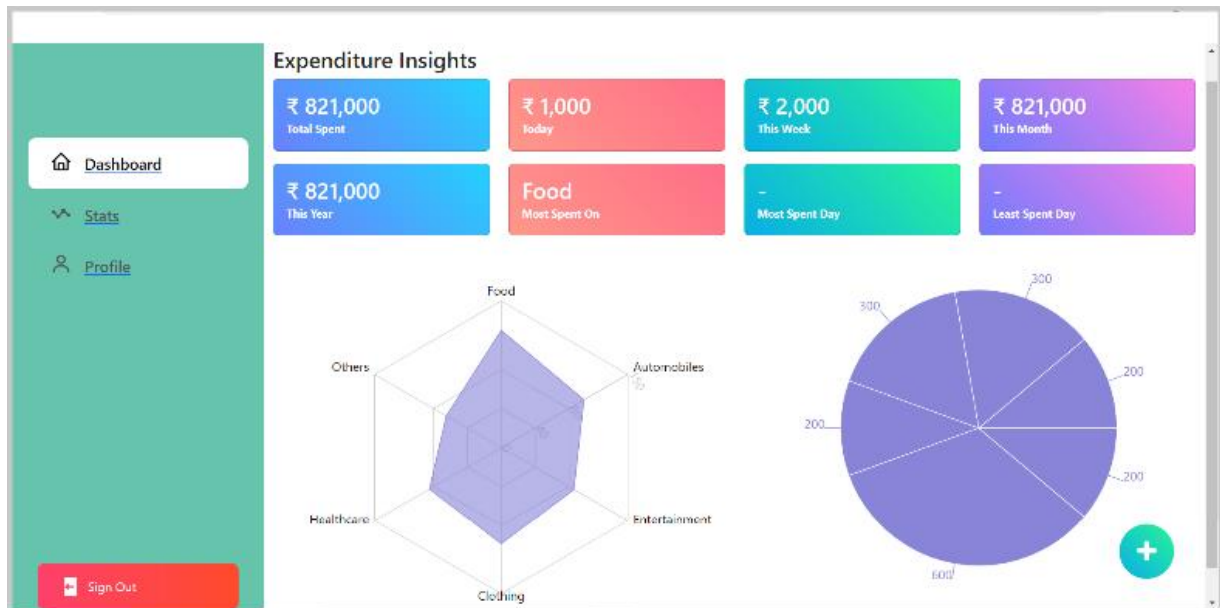

Figure 7.2 – login page

7.1 Feature 3:

Figure 7.3 – Dashboard

7.1 Feature 4:

Figure 7.4 – Stats page

CHAPTER 8

TESTING

8.1 TEST CASES :

1. Login button click with wrong credentials entered.

2. Signup with already registered mail ID.

3. Signup with wrong form data entered.

4. Entering home page with logged out session.

5. delete expense triggers change in graph.

6. Add expense without choosing category.

8.2 USER ACCEPTANCE TESTING

s. n o	Test Cas e id	Featur e Type	compone nt	Test description	Input test Data	Actual output	Expected output	remark s
1	TC – RG 01	Func- tional	Register page	register for the application by entering my name, email, password, monthly limit	User1 User1@gmail.co m ***** 10000	Registratio n successful	Registratio n successful	pass
2	TC – SI 01	Func- tional	Login page	log into the application by entering email & password	User1@gmail.co m *****	Login successful	Login sucessfull	pass
3	TC – ST 01	UI	Stats page	view my entire expenses throughout a particular period of time		Expenses are displayed For particular time	Expenses are displayed For particular time	pass

4	TC – DB 01	UI	Dash-board	Display graph in dashboard		Graph is displayed	Graph is displayed	pass
5	TC – ST 02	Func-tional	Stats page	generate reports based on my previous expenditures		Reports generated in graphical form	Reports generated in graphical form	pass
6	TC – SI 02	Func-tional	Dash-board	can logout		Go to sign page	Sign in page displayed	pass
7	TC – ST 03	Func-tional	Stats page	create expense	14-11-2022 100 Food Debit Night food	Expenses created	Expenses created	pass
8	TC – ST 04	Func-tional	Stats page	can edit ,delete, update expense		Expenses updated	Updated of expenses	pass
9	TC – ST 05	UI	Stats page	can view credit and debit expenses separately.		Expenses are listed separately	Expenses are listed separately	pass
1 0	TC – ST 06	UI	Stats page	aware of the expense that I spend the most on		Expenses are listed for particular category	Expenses are listed for particular category	pass
1 1	TC – PG 01	Func-tional	Profile page	able to update my set monthly limit		Monthly limit updated	Monthly limit updated	pass
1 2	TC – PG 01	UI	Profile page	able to view my profile		Profile details displayed	Profile details displayed	pass

CHAPTER 9

9.1 PERFORMANCE METRICS

1. Hours worked : 50 hours
2. Stick to Timelines : 100%
3. Consistency of the product : 75%
4. Efficiency of the product : 80%
5. Quality of the product : 85%

CHAPTER 10

ADVANTAGES AND DISADVANTAGES

Advantages :

- Which allows users to track their expenses daily, weekly, monthly, and yearly in terms of summary, bar graphs, and pie-charts.

- Separate view for credit and debit transactions

- no burden of manual calculations

- generate and save reports.

- You can insert, delete records

- You can track expenses by categories like food, automobile, entertainment, education etc..

- You can track expenses by time, weekly, month, year etc..

- Setting monthly limits and we can update it later

- Customized email alerts when limit exceeds.

Disadvantages :

- User have entry every records manually
- The category divided may be blunder or messy
- Can't able to customized user defined categories

CHAPTER 11

CONCLUSION :

In this paper, 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 make them aware about their daily expenses. It will prove to be helpful for the people who are frustrated with their daily budget management, irritated because of the amount of expenses and wish 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.

CHAPTER 12

FUTURE SCOPE :

- ➢ In further days, there will be mails and payment embedded with the app. Also, backup details will be recorded on cloud.
- ➢ Here user can define their own categories for expense type like food, clothing, rent and bills where they have to enter the money that has been spend .
- ➢ Alerts for paying dues and remainders to record input at particular user defined time.

CHAPTER 13

APPENDIX :

add_expense_model.jsx :

```jsx
import "./add_expense_modal.css";
import React, { useState, useEffect } from "react";
import { ToastContainer, toast } from "react-toastify";
import "react-toastify/dist/ReactToastify.css";
import useStore from "../../state";
const AddExpenseModal = ({ handleClose, show }) => {
    const showHideClassName = show ? "modal display-block" : "modal display-none";

    const addExpense = useStore(state => state.addExpense)
    const fetchExpenditureBreakdown = useStore(state => state.fetchExpenditureBreakdown)

    const userId = useStore(state => state.userId)

    const [formData, setFormData] = useState({
        date: "",
        amount: "",
        category: "Food",
        description: "",
        expenseType: "debit",
    });

    let categories = [
        "Food",
        "Automobiles",
        "Entertainment",
        "Clothing",
        "Healthcare",
        "Others",
    ];
    let expenseTypes = ["debit", "credit"];
```

expenditure_breakdown.jsx

```jsx
import React, { useEffect, useState } from "react";
import "bootstrap/dist/css/bootstrap.css";
import "../../pages/home/home.css";
import LoadingDots from "../loader/loading_dots";
import useStore from "../../state";

const ExpenditureBreakdown = () => {
    const userId = useStore(state => state.userId)
    const expenditureBreakdown = useStore(state => state.expenditureBreakdown)
    const fetchingExpenditureBreakdown = useStore(state=>state.fetchingExpenditureBreakdown)
    const fetchExpenditureBreakdown = useStore(state => state.fetchExpenditureBreakdown)
    const gradients = [
        "card-purple-blue",
        "card-salmon-pink",
        "card-blue-green",
        "card-purple-pink",
    ];

    useEffect( () => {
        async function fetch(){
            if(!fetchingExpenditureBreakdown && Object.keys(expenditureBreakdown).length  === 0){
                await fetchExpenditureBreakdown(fetchingExpenditureBreakdown,userId)
            }
        }
        fetch();
    });

    return (
        <div className="container-fluid">
            <div className="row row-cols-4">
                {Object.entries(expenditureBreakdown)
                    .slice(0, 4)
```

Expense_table.jsx

```jsx
import React, { useEffect, useState } from "react";
import "./expense_table.css";
import dayjs from "dayjs";
import Modal from "react-modal";
import useStore from "../../state";
import Loader from "../../components/loader/loader"

const ExpenseTable = () => {
    let expenseTypes = ["All Expenses", "Credit", "Debit"];
    const [expenseType, setExpenseType] = useState("All Expenses");
    const [showDeleteExpensePopup, setShowDeleteExpensePopup] = useState(false);
    const [loading, setLoading] = useState(false);
    const allExpenses = useStore((state) => state.allExpenses);
    const creditExpenses = useStore((state) => state.creditExpenses);
    const debitExpenses = useStore((state) => state.debitExpenses);
    const [expenseTable,setExpenseTable] = useState(allExpenses)

    const fetchExpensesTable = useStore((state) => state.fetchExpensesTable);

    const userId = useStore((state) => state.userId);

    const onExpenseTypeChange = (e) => {
        setExpenseType(e.target.value);
        setExpenseTable(expenseType == "All Expenses" ? allExpenses : (expenseType == "Credit" ? creditExpenses : debitExpe
        fetch();
    };

    const onDeleteExpenseClick = () => {
        setShowDeleteExpensePopup(!showDeleteExpensePopup);
    };
```

Expense_charts.jsx

```jsx
import React, { PureComponent } from "react";
import {
    Radar,
    RadarChart,
    PolarGrid,
    Legend,
    PolarAngleAxis,
    PolarRadiusAxis,
    ResponsiveContainer,
    PieChart,
    Pie,
    Sector,
    Cell,
} from "recharts";

const data = [
    {
        subject: "Food",
        A: 120,
        B: 110,
        fullMark: 3000,
    },
    {
        subject: "Automobiles",
        A: 98,
        B: 130,
        fullMark: 3000,
    },
    {
        subject: "Entertainment",
        A: 86,
        B: 130,
```

Home.js

```
JS home.js    ✕

personal-expense-tracker > src > pages > home > JS home.js > [∅] Home > [∅] onFabClick
  1   import React, { useEffect, useRef, useState } from "react";
  2   import "./home.css";
  3   import ExpenditureBreakdown from "../../components/expenditure_breakdown/expenditure_breakdown";
  4   import "font-awesome/css/font-awesome.min.css";
  5   import { useNavigate } from "react-router-dom";
  6   import AddExpenseModal from "../../components/add_expense_modal/add_expense_modal";
  7   import ExpenseCharts from "../../components/expense_charts/expense_charts";
  8   import { useCookies } from 'react-cookie';
  9
 10   const Home = () => {
 11       const navigate = useNavigate();
 12       const [showPopup, setShowPopup] = useState(false);
 13       const [cookies,setCookie] = useCookies(['userId'])
 14
 15       const onFabClick = () => {
 16           setShowPopup(!showPopup);
 17       };
 18       return (
 19           <div className="main-container">
 20               <div className="main-area">
 21                   <h3 className="hello-text">Expenditure Insights</h3>
 22                   <ExpenditureBreakdown />
 23                   {/* <h1 className='hello-text'>Most Recent Expenses</h1> */}
 24                   {/* <ExpenditureTable/> */}
 25                   {/* <h1 className='hello-text'>Expenditure Insights</h1> */}
 26
 27                   <ExpenseCharts />
 28                   {showPopup ? (
 29                       <AddExpenseModal
 30                           show={showPopup}
 31                           handleClose={onFabClick}
```

app.py

```
app.py    ✕    add_expense_modal.jsx    expenditure_breakdown.jsx    expense_table.jsx    app_layout.jsx    expense_charts.jsx

app.py > ...
                                          >   add                          Aa ab .*  ? of 3        ↑  ↓  ≡  ✕
  5   import uuid
  6   import datetime
  7   from datetime import datetime, timedelta, date
  8   import calendar
  9   # import os
 10   from sendgrid import SendGridAPIClient
 11   # from sendgrid.helpers.mail import Mail
 12
 13
 14   app = Flask(__name__)
 15   cors = CORS(app)
 16
 17   # API = "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"
 18
 19   try:
 20       print("Connecting")
 21       conn=ibm_db.connect('DATABASE=bludb;HOSTNAME=6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdom
 22       print("Successfully connected")
 23   except Exception as e:
 24       print(ibm_db.conn_errormsg())
 25   @app.route('/')
 26   @cross_origin()
 27   def hello():
 28       return 'hello'
 29
 30   @app.route('/login', methods = ['POST'])
 31   @cross_origin()
 32   def login():
 33       email = request.form['email']
 34       password = request.form['password']
 35       try:
 36           stmt = ibm_db.exec_immediate(conn, "select * from users where email = '%s' and password = '%s'" % (email,password))
```

Project Demonstration Link : https://youtu.be/IWXs5q-sPBI

Source code link : https://github.com/IBM-EPBL/IBM-Project-16706-1659620894