PROJECT REPORT

personal expense tracker application

Submitted by

PNT2022TMID41150

N Madhan
S Sasikala
N Thamiz
S Monisha

TABLE OF CONTENTS

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

7.1 Feature 1

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

Source Code

GitHub & Project Demo Link

NTRODUCTION

1.1 PROJECT OVERVIEW

Expense tracker is an android based application. This application allows the user to maintain a computerized diary. Expense tracker application which will keep a track of Expenses of a user on a day to-day basis. This application keeps a record of your expenses and also will give you a category wise distribution of your expenses. With the help of this application user can track their daily/weekly/monthly expenses. This application will also have a feature which will help you stay on budget because you know your expenses. Expense tracker application will generate report at the end of month to show Expense via a graphical representation. We also have added a special feature which will distributes your expenses in different categories suitable for the user. An expense history will also be provided in application.

Now a day's people are concerned about regularity of their daily expenses. This is done mainly for keep a track of the users' daily expenses to have a control of users' monthly expenses. We have developed an android application named as "Expense Tracker Application" and this application is used to manage the user's daily expenses in a more coherent and manageable way [10]. This application will help us to reduces the manual calculations for their daily expenses and also keep the track of the expenses. With the help of this application, user can calculate his total expenses per day and these results will stored for unique user. As the traditional methods of budgeting, we need to maintain the Excel sheets, Word Documents, notes, and files for the user daily and monthly expenses. There is no as such full-fledged solution to keep a track of our daily expenses easily. Keeping a log in diary is a very monotonous process and also may sometimes lead into problems due to the manual calculations. Looking on all the above given conditions, we are trying to satisfy the user requirements by building a mobile application which will help them reduces their burdens. "Expense Tracker Application" is an application where one can enter their daily expenses and end of the day, they know their expenses in charts.

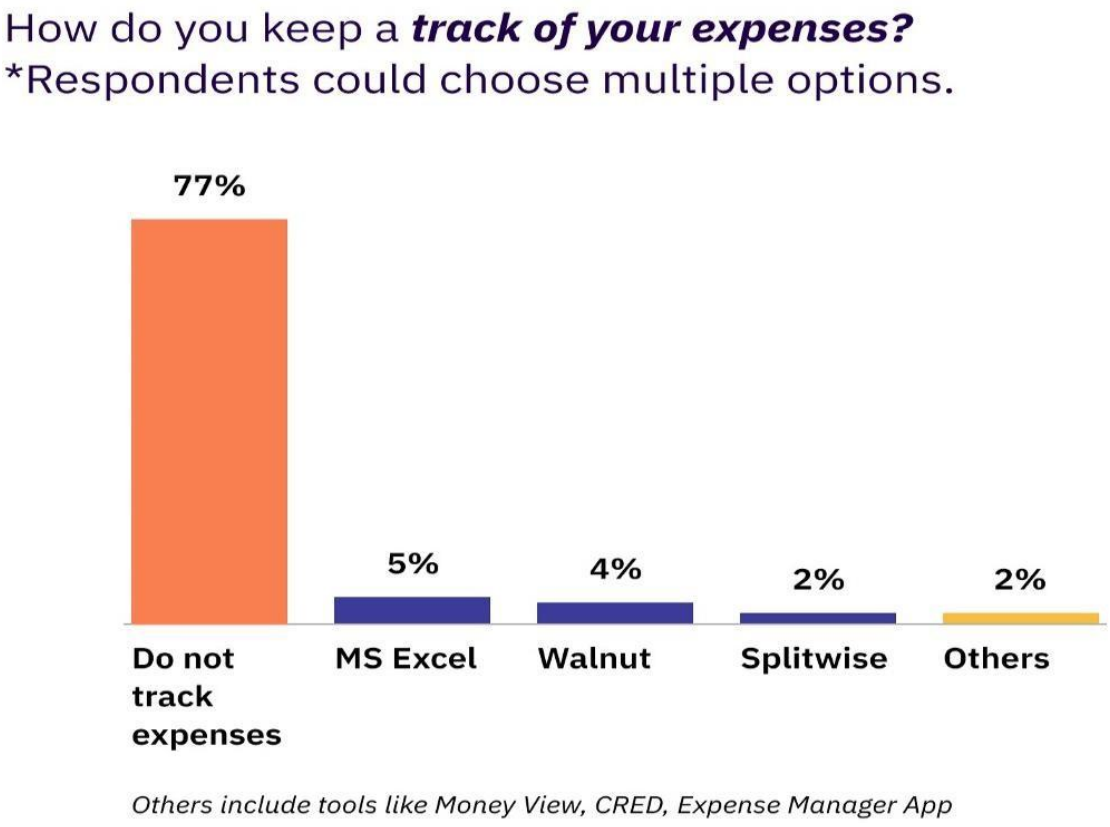
The idea of developing this project in mobile platform for user convenience. Because whenever they make expenses immediately, they add in mobile application. Some of the concerns maintaining a personal expense is a BIG problem, in daily expenses many times we don't know where the money goes. Some of the conventional methods used to tackle this problem in normal circumstances are like making use of a sticky notes by common users, Proficient people deals with this kind of problems by using spreadsheets to record expense and using a ledger to maintains the large amounts data by especially by expert people. As this shows that it is various methods used by different people. This makes using this data contrary. There is still complication in areas like there is no assurance for data compatible, there are chances of crucial inputs can be missed and the manual errors may sneak in. The Data recorders are not always handled, and it could be hectic process to have overall view of those expenses. We believe a handy design and a handy mobile application which handles these troubles. Such that app is capable of recording the expenditure and giving broad view with easy to use the user interface and this application is intelligent enough to shows the history of expenses noted in the app

1.2 Purpose

The mobile applications that are available in the market are very helpful to the smartphone users and make their life accessible. The expense tracker is also one of those applications, which much extent in daily life.[8] As there are many similar apps available today, we added some innovative components to make our application unique, easy to use and coherent. Apart from adding unique features like view analytics and expense history in the application, we also added features like multiple user accounts. We have an idea of making use of application for the purpose of survey in the field of expenses of user. This idea serves as a main objective of research project [1][2]. The research also includes view the reports at the form of charts

2. LITERATURE SURVEY

A **literature survey** is an overview of the previously published works on a specific topic. The term can refer to a full scholarly paper or a section of a scholarly work such as a book, or an article. A good literature review can ensure that a proper research question has been asked and a proper theoretical framework and/or research methodology have been chosen.



2.1 Existing problem

The Expense tracker existing system does not provide the user portable device management level, existing system only used on desktop software so unable to update anywhere expenses done and unable to update the location of the expense details disruptive that the proposed system provides [6]. In existing, we need to maintain the Excel sheets, CSV files for the user daily, weekly and monthly expenses. In existing, 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. But this project will not have any reminder to remain a person in a specific date, so that is the only drawback in which the remainder is not present. This project will be an unpopulated information because it has some disadvantages by not remind a person for each and every month. But it can used to perform calculation on income and expenses to overcome this problem we propose the new project.

2.2 References

Expense Tracker is going to be a mobile application so that It can be accessed any time required [11]. 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 data base. 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 an tics 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 create this system, we will use the android studio and it[13].It will be written in Java, X m l. MySQL will be the database used.

2.3 Problem Statement Definition

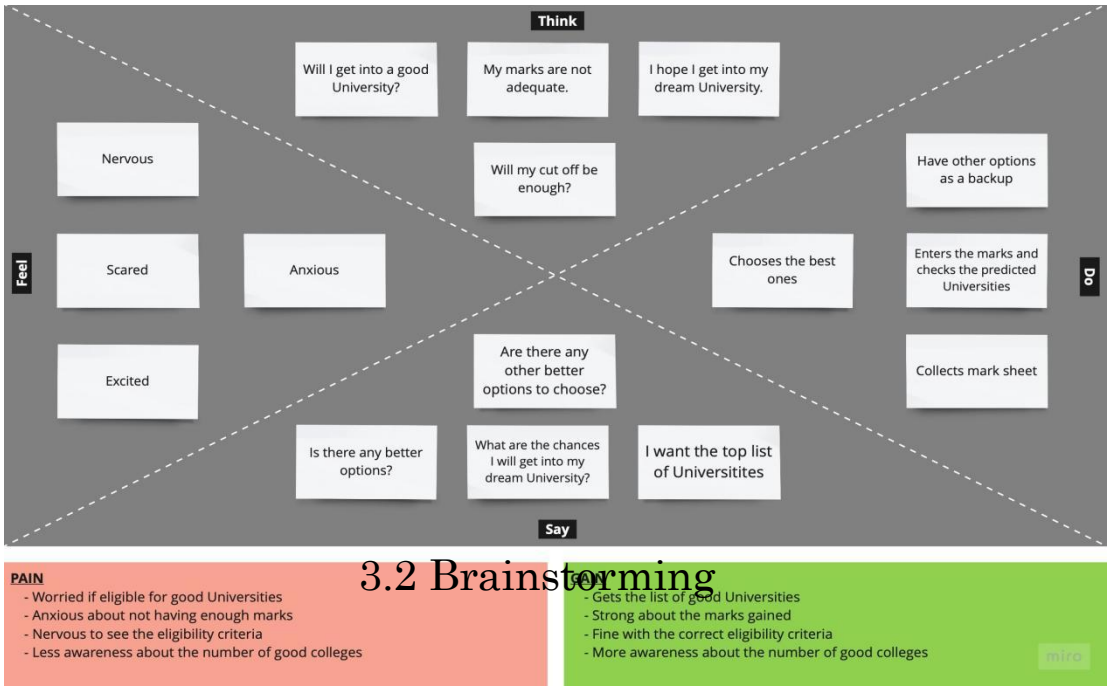
At the instant, there is no as such complete solution present easily or we should say free of cost which enables a person to keep a track of its daily expenditure easily. To do so a person has to keep a log in a diary or in a computer, also all the calculations needs to be done by the user which may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

As the name itself suggests, this project is an attempt to manage our daily expenses in a more efficient and manageable way. The system attempts to free the user with as much as possible the burden of manual calculation and to keep the track of the expenditure. Instead of keeping a dairy or a log of the expenses on the smartphones or laptops, this system.

3. IDEATION & PROPOSED SOLUTION

Also known as expense manager and money manager, an expense tracker is a software or application that helps to keep an accurate record of your money inflow and outflow. 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.

3.1 Empathy Map Canvas



3.2 Brainstorming

BRAINSTORMING

3.2 Proposed Solution

1. The problem statement is to design a college prediction/ prediction system and to provide a probabilistic insight into college administration for overall rating, cut-offs of the colleges, admission intake and preferences of students

2. It has always been a troublesome process for students in finding the perfect university and course for their further studies.

3. At times they do know which stream they want to get into, but it is not easy for them to find colleges based on their academic marks and other performances.

4. We aim to develop and provide a place which would give a probabilistic output of how likely it is to get into a university given their details

Abstract:

● Students are often worried about their chances of admission to University.

● The aim of this project is to help students in shortlisting universities with their profiles

● The predicted output gives them a fair idea about their admission chances to a particular university.

● This analysis should also help students who are currently preparing or will be preparing to get a better idea.

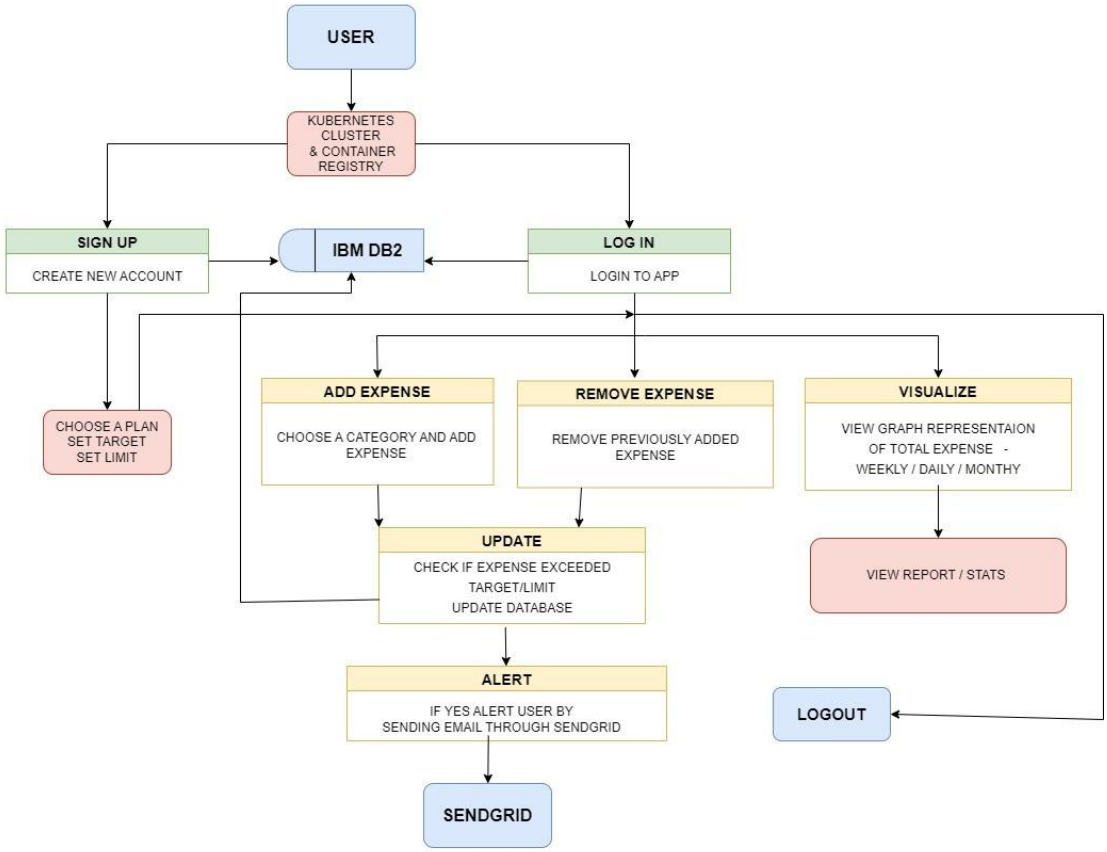
4. REQUIREMENT ANALYSIS

4.1 Functional requirement

As defined in requirements engineering, functional requirements specify particular results of a system. This should be contrasted with non-functional requirements, which specify overall characteristics such as cost and reliability.

5. PROJECT DESIGN

5.1 Data Flow Diagrams



5. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

```
const balance = document.getElementById('balance');
const money_plus = document.getElementById('money-plus');
const money_minus = document.getElementById('money-minus');
const list = document.getElementById('list');
const form = document.getElementById('form');
const text = document.getElementById('text');
const amount = document.getElementById('amount');

// const dummyTransactions = [
//   { id: 1, text: 'Flower', amount: -20 },
```

```
//   { id: 2, text: 'Salary', amount: 300 },
//   { id: 3, text: 'Book', amount: -10 },
//   { id: 4, text: 'Camera', amount: 150 }
// ];
```

7. CODING & SOLUTIONING

7.1 Feature

```
const balance = document.getElementById('balance');
const money_plus = document.getElementById('money-plus');
const money_minus = document.getElementById('money-minus');
const list = document.getElementById('list');
const form = document.getElementById('form');
const text = document.getElementById('text');
const amount = document.getElementById('amount');

// const dummyTransactions = [
//   { id: 1, text: 'Flower', amount: -20 },
//   { id: 2, text: 'Salary', amount: 300 },
//   { id: 3, text: 'Book', amount: -10 },
//   { id: 4, text: 'Camera', amount: 150 }
// ];

const localStorageTransactions = JSON.parse(
  localStorage.getItem('transactions')
);

let transactions =
  localStorage.getItem('transactions') !== null ? localStorageTransactions :
[];

// Add transaction
function addTransaction(e) {
  e.preventDefault();

  if (text.value.trim() === '' || amount.value.trim() === '') {
    alert('Please add a text and amount');
  } else {
    const transaction = {
      id: generateID(),
      text: text.value,
      amount: +amount.value
    };
```

```
    transactions.push(transaction);

    addTransactionDOM(transaction);

    updateValues();

    updateLocalStorage();

    text.value = '';
    amount.value = '';
  }
}

// Generate random ID
function generateID() {
  return Math.floor(Math.random() * 100000000);
}

// Add transactions to DOM list
function addTransactionDOM(transaction) {
  // Get sign
  const sign = transaction.amount < 0 ? '-' : '+';

  const item = document.createElement('li');

  // Add class based on value
  item.classList.add(transaction.amount < 0 ? 'minus' : 'plus');

  item.innerHTML = `
    ${transaction.text} <span>${sign}${Math.abs(
    transaction.amount
  )}</span> <button class="delete-btn" onclick="removeTransaction(${
    transaction.id
  })">x</button>
  `;

  list.appendChild(item);
}

// Update the balance, income and expense
function updateValues() {
  const amounts = transactions.map(transaction => transaction.amount);

  const total = amounts.reduce((acc, item) => (acc += item), 0).toFixed(2);
```

```
  const income = amounts
    .filter(item => item > 0)
    .reduce((acc, item) => (acc += item), 0)
    .toFixed(2);

  const expense = (
    amounts.filter(item => item < 0).reduce((acc, item) => (acc += item), 0)
*
    -1
  ).toFixed(2);

  balance.innerText = `$${total}`;
  money_plus.innerText = `$${income}`;
  money_minus.innerText = `$${expense}`;
}

// Remove transaction by ID
function removeTransaction(id) {
  transactions = transactions.filter(transaction => transaction.id !== id);

  updateLocalStorage();

  init();
}

// Update local storage transactions
function updateLocalStorage() {
  localStorage.setItem('transactions', JSON.stringify(transactions));
}

// Init app
function init() {
  list.innerHTML = '';

  transactions.forEach(addTransactionDOM);
  updateValues();
}

init();

form.addEventListener('submit', addTransaction);
```

8. TESTING

8.1 Test Cases

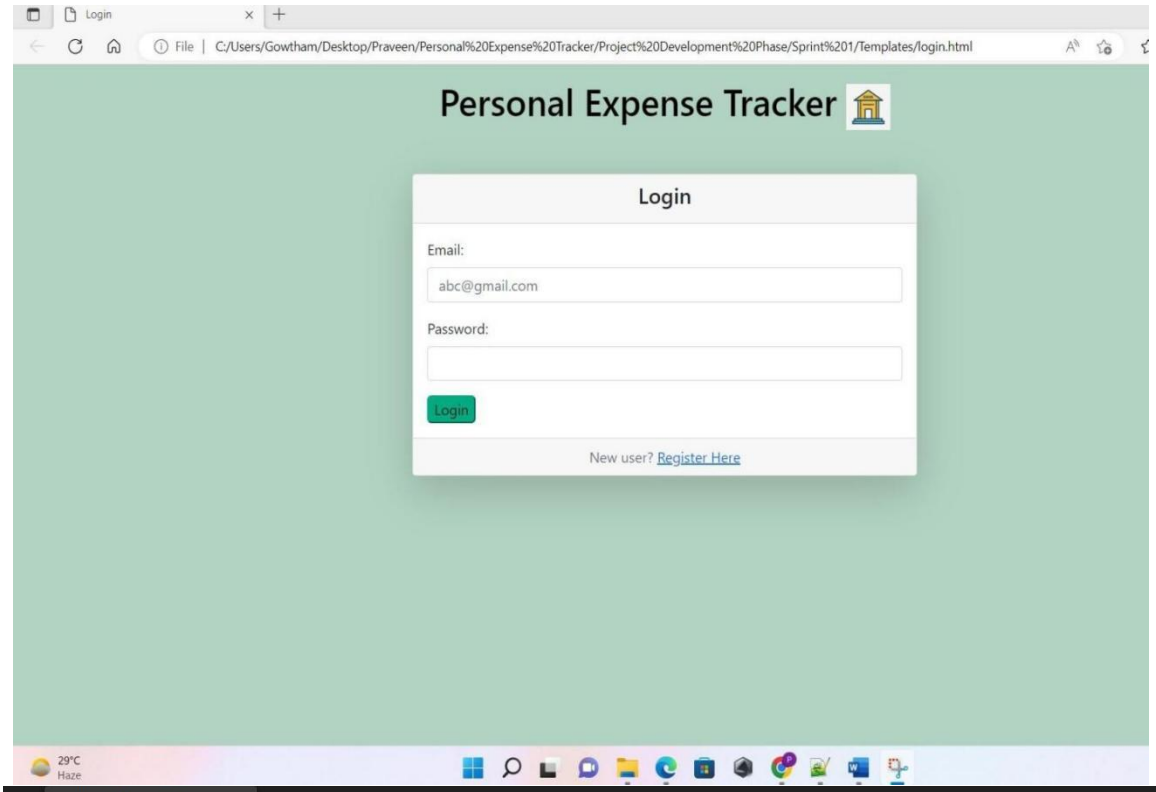
```
@importurl('https://fonts.googleapis.com/css?family=Lato&display=swap
');

:root {
  --box-shadow: 0 1px 3px rgba(0, 0, 0, 0.12), 0 1px 2px rgba(0, 0, 0, 0.24);
}

* {
  box-sizing: border-box;
}

body {
  background-color: #f7f7f7;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  min-height: 100vh;
  margin: 0;
  font-family: 'Lato', sans-serif;
}

.container {
  margin: 30px auto;
  width: 350px;
}

h1 {
  letter-spacing: 1px;
  margin: 0;
}

h3 {
  border-bottom: 1px solid #bbb;
  padding-bottom: 10px;
  margin: 40px 0 10px;
}

h4 {
  margin: 0;
  text-transform: uppercase;
}

.inc-exp-container {
  background-color: #fff;
```

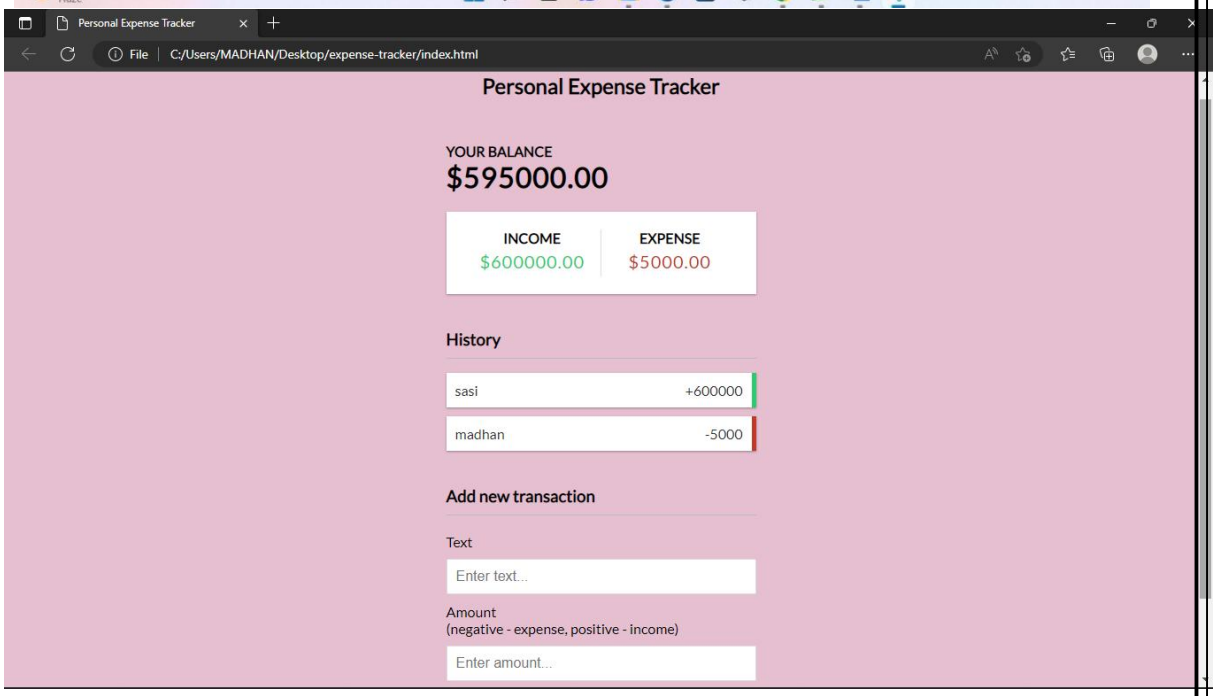
```
  box-shadow: var(--box-shadow);
  padding: 20px;
  display: flex;
  justify-content: space-between;
  margin: 20px 0;
}

.inc-exp-container > div {
  flex: 1;
  text-align: center;
}

.inc-exp-container > div:first-of-type {
  border-right: 1px solid #dedede;
}

.money {
  font-size: 20px;
  letter-spacing: 1px;
  margin: 5px 0;
}

.money.plus {
  color: #2ecc71;
}

.money.minus {
  color: #c0392b;
}

label {
  display: inline-block;
  margin: 10px 0;
}

input[type='text'],
input[type='number'] {
  border: 1px solid #dedede;
  border-radius: 2px;
  display: block;
  font-size: 16px;
  padding: 10px;
  width: 100%;
}

.btn {
```

```
  cursor: pointer;
  background-color: #9c88ff;
  box-shadow: var(--box-shadow);
  color: #fff;
  border: 0;
  display: block;
  font-size: 16px;
  margin: 10px 0 30px;
  padding: 10px;
  width: 100%;
}

.btn:focus,
.delete-btn:focus {
  outline: 0;
}

.list {
  list-style-type: none;
  padding: 0;
  margin-bottom: 40px;
}

.list li {
  background-color: #fff;
  box-shadow: var(--box-shadow);
  color: #333;
  display: flex;
  justify-content: space-between;
  position: relative;
  padding: 10px;
  margin: 10px 0;
}

.list li.plus {
  border-right: 5px solid #2ecc71;
}

.list li.minus {
  border-right: 5px solid #c0392b;
}

.delete-btn {
  cursor: pointer;
  background-color: #e74c3c;
  border: 0;
```

```
  color: #fff;
  font-size: 20px;
  line-height: 20px;
  padding: 2px 5px;
  position: absolute;
  top: 50%;
  left: 0;
  transform: translate(-100%, -50%);
  opacity: 0;
  transition: opacity 0.3s ease;
}

.list li:hover .delete-btn {
  opacity: 1;
}
```

9. RESULST


Login
File | C:/Users/Gowtham/Desktop/Praveen/Personal%20Expense%20Tracker/Project%20Development%20Phase/Sprint%201/Templates/login.html
Personal Expense Tracker
Login
Email:
abc@gmail.com
Password:
Login
New user? Register Here
29°C
Haze


Personal Expense Tracker
File | C:/Users/MADHAN/Desktop/expense-tracker/index.html
Personal Expense Tracker
YOUR BALANCE
$595000.00
INCOME
$600000.00
EXPENSE
$5000.00
History
sasi +600000
madhan -5000
Add new transaction
Text
Enter text...
Amount
(negative - expense, positive - income)
Enter amount...

Source Code

```
const localStorageTransactions = JSON.parse(
  localStorage.getItem('transactions')
);

let transactions =
  localStorage.getItem('transactions') !== null ? localStorageTransactions : [];

// Add transaction
function addTransaction(e) {
  e.preventDefault();

  if (text.value.trim() === '' || amount.value.trim() === '') {
    alert('Please add a text and amount');
  } else {
    const transaction = {
      id: generateID(),
      text: text.value,
      amount: +amount.value
    };

    transactions.push(transaction);

    addTransactionDOM(transaction);

    updateValues();

    updateLocalStorage();

    text.value = '';
    amount.value = '';
  }
}

// Generate random ID
function generateID() {
  return Math.floor(Math.random() * 100000000);
}

// Add transactions to DOM list
function addTransactionDOM(transaction) {
  // Get sign
  const sign = transaction.amount < 0 ? '-' : '+';

  const item = document.createElement('li');

  // Add class based on value
  item.classList.add(transaction.amount < 0 ? 'minus' : 'plus');

  item.innerHTML = `
    ${transaction.text} <span>${sign}${Math.abs(
    transaction.amount
  )}</span> <button class="delete-btn" onclick="removeTransaction(${
    transaction.id
```

```
  })">x</button>
  `;

  list.appendChild(item);
}

// Update the balance, income and expense
function updateValues() {
  const amounts = transactions.map(transaction => transaction.amount);

  const total = amounts.reduce((acc, item) => (acc += item), 0).toFixed(2);

  const income = amounts
    .filter(item => item > 0)
    .reduce((acc, item) => (acc += item), 0)
    .toFixed(2);

  const expense = (
    amounts.filter(item => item < 0).reduce((acc, item) => (acc += item), 0) *
    -1
  ).toFixed(2);

  balance.innerText = `$${total}`;
  money_plus.innerText = `$${income}`;
  money_minus.innerText = `$${expense}`;
}

// Remove transaction by ID
function removeTransaction(id) {
  transactions = transactions.filter(transaction => transaction.id !== id);

  updateLocalStorage();

  init();
}

// Update local storage transactions
function updateLocalStorage() {
  localStorage.setItem('transactions', JSON.stringify(transactions));
}

// Init app
function init() {
  list.innerHTML = '';

  transactions.forEach(addTransactionDOM);
  updateValues();
}

init();

form.addEventListener('submit', addTransaction);
```

```
@import url('https://fonts.googleapis.com/css?family=Lato&display=swap');

:root {
  --box-shadow: 0 1px 3px rgba(0, 0, 0, 0.12), 0 1px 2px rgba(0, 0, 0, 0.24);
}

* {
  box-sizing: border-box;
}

body {
  background-color: #f7f7f7;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  min-height: 100vh;
  margin: 0;
  font-family: 'Lato', sans-serif;
}

.container {
  margin: 30px auto;
  width: 350px;
}

h1 {
  letter-spacing: 1px;
  margin: 0;
}

h3 {
  border-bottom: 1px solid #bbb;
  padding-bottom: 10px;
  margin: 40px 0 10px;
}

h4 {
  margin: 0;
  text-transform: uppercase;
}

.inc-exp-container {
  background-color: #fff;
  box-shadow: var(--box-shadow);
  padding: 20px;
  display: flex;
  justify-content: space-between;
  margin: 20px 0;
}

.inc-exp-container > div {
  flex: 1;
  text-align: center;
}
```

```
.inc-exp-container > div:first-of-type {
  border-right: 1px solid #dedede;
}

.money {
  font-size: 20px;
  letter-spacing: 1px;
  margin: 5px 0;
}

.money.plus {
  color: #2ecc71;
}

.money.minus {
  color: #c0392b;
}

label {
  display: inline-block;
  margin: 10px 0;
}

input[type='text'],
input[type='number'] {
  border: 1px solid #dedede;
  border-radius: 2px;
  display: block;
  font-size: 16px;
  padding: 10px;
  width: 100%;
}

.btn {
  cursor: pointer;
  background-color: #9c88ff;
  box-shadow: var(--box-shadow);
  color: #fff;
  border: 0;
  display: block;
  font-size: 16px;
  margin: 10px 0 30px;
  padding: 10px;
  width: 100%;
}

.btn:focus,
.delete-btn:focus {
  outline: 0;
}

.list {
  list-style-type: none;
  padding: 0;
  margin-bottom: 40px;
}
```

```
.list li {
  background-color: #fff;
  box-shadow: var(--box-shadow);
  color: #333;
  display: flex;
  justify-content: space-between;
  position: relative;
  padding: 10px;
  margin: 10px 0;
}

.list li.plus {
  border-right: 5px solid #2ecc71;
}

.list li.minus {
  border-right: 5px solid #c0392b;
}

.delete-btn {
  cursor: pointer;
  background-color: #e74c3c;
  border: 0;
  color: #fff;
  font-size: 20px;
  line-height: 20px;
  padding: 2px 5px;
  position: absolute;
  top: 50%;
  left: 0;
  transform: translate(-100%, -50%);
  opacity: 0;
  transition: opacity 0.3s ease;
}

.list li:hover .delete-btn {
  opacity: 1;


import smtplib
import sendgrid as sg
import os
from sendgrid.helpers.mail import Mail, Email, To, Content
SUBJECT = "expense tracker"
s = smtplib.SMTP('smtp.gmail.com', 587)

def sendmail(TEXT,email):
    print("sorry we cant process your candidature")
    s = smtplib.SMTP('smtp.gmail.com', 587)
    s.starttls()
    # s.login("il.madhankumar31049@gmail.com", "oms@1Ram")
    s.login("madhankumar31049@gmail", "lxixbmpnexbkiemh")
    message  = 'Subject: {}\n\n{}'.format(SUBJECT, TEXT)
    # s.sendmail("il.madhankumar31049@gmail.com", email, message)
    s.sendmail("il.madhankumar31049@gmail", email, message)
    s.quit()
def sendgridmail(user,TEXT):
```

```
# from_email = Email("madhankumar31049@gmail.com")
from_email = Email("madhankumar31049@gmail.com")
to_email = To(user)
subject = "Sending with SendGrid is Fun"
content = Content("text/plain",TEXT)
mail = Mail(from_email, to_email, subject, content)

# Get a JSON-ready representation of the Mail object
mail_json = mail.get()
# Send an HTTP POST request to /mail/send
response = sg.client.mail.send.post(request_body=mail_json)
print(response.status_code)
print(response.headers)
```

GitHub & Project Demo Link

https://github.com/IBM-EPBL/IBM-Project-47422-1660799200