

TeamID	PNT2022TMID31819
Project Name	Personal Expense Tracker Application
Team members	Naveen kumar dugar(Team Lead) Ajay M Selva kumar S Kirubanithi V

CHAPTER 1

INTRODUCTION

1.1 Project Overview

In simple words, personal finance entails all the financial decisions and activities that a Finance app makes your life easier by helping you to manage your finances efficiently. A personal finance app will not only help you with budgeting and accounting but also give you helpful insights about money management.

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 email alert.

1.2 Purpose

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 befitting strategy of getting out of debt.

Budgeting ensures you're not spending more than you're making, allowing you to plan for short- and long-term expenses. It's an easy, helpful way for people with all types of income and expenses to keep their finances in order.

CHAPTER 2 LITERATURE SURVEY

Literature review was carried out to gain knowledge and improve the skills needed to complete this project. This chapter shows the different techniques that have been implemented.

2.1 Existing Problem

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.

2.2 References

- i. <http://expense-manager.com/how-expensesoftware/>
- ii. <https://www.splitwise.com/terms>
- iii. <http://code.google.com/p/socialauthandroid/wiki/Facebook>
- iv. <http://code.google.com/p/socialauth-android>
- v. Developer.android.com

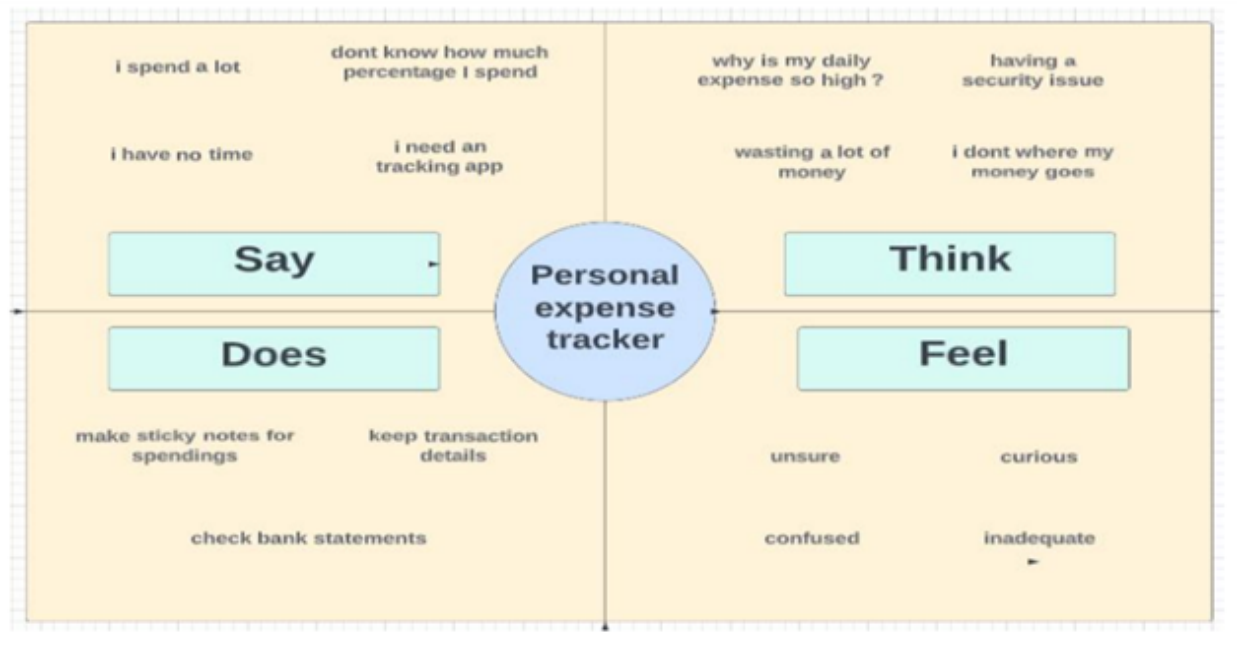
2.3 Problem Statement Definition

It is tough to keep track of all the financial decisions and activities that a person makes. Traditional expense tracking methods are inconvenient and unreliable. In order to get a quick overview about your total incomes and expenses and control spending, it is convenient to digitize the process by having a personal expense tracker.

CHAPTER 3

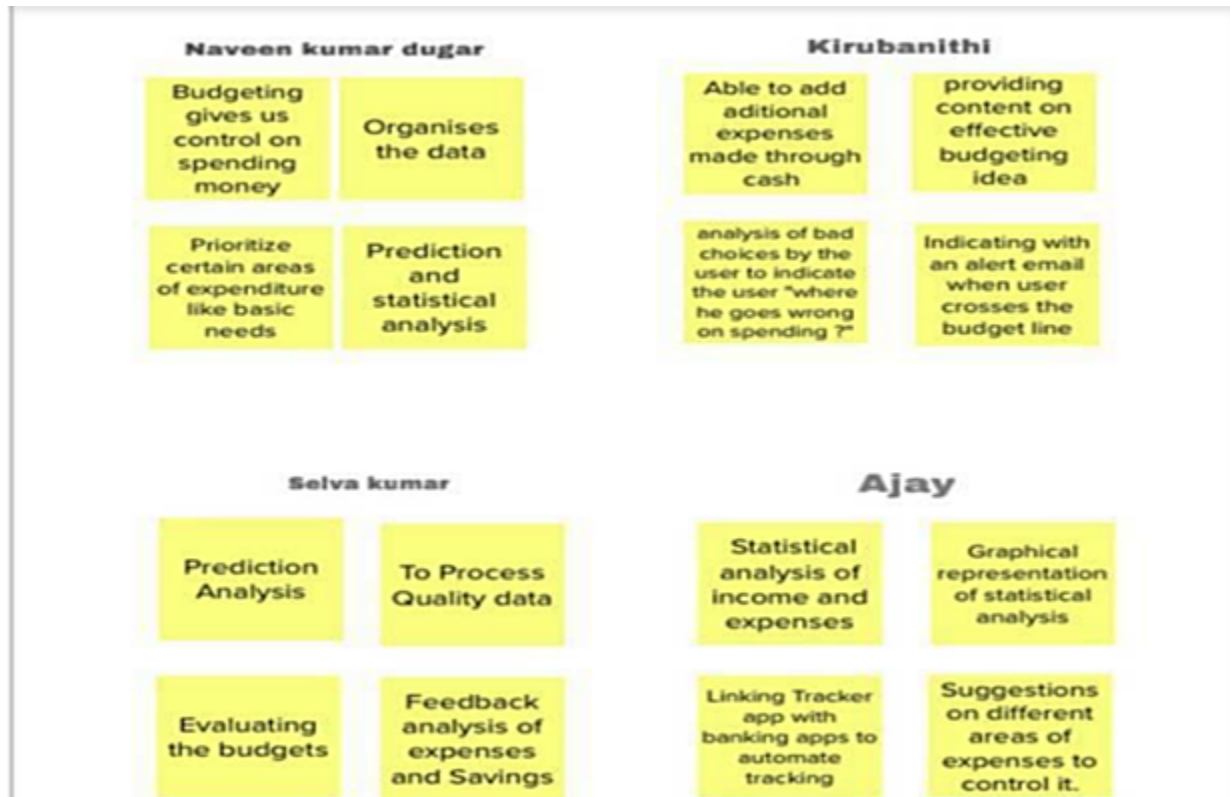
IDEATION & PROPOSEDSOLUTION

3.1 Empathy Map Canvas

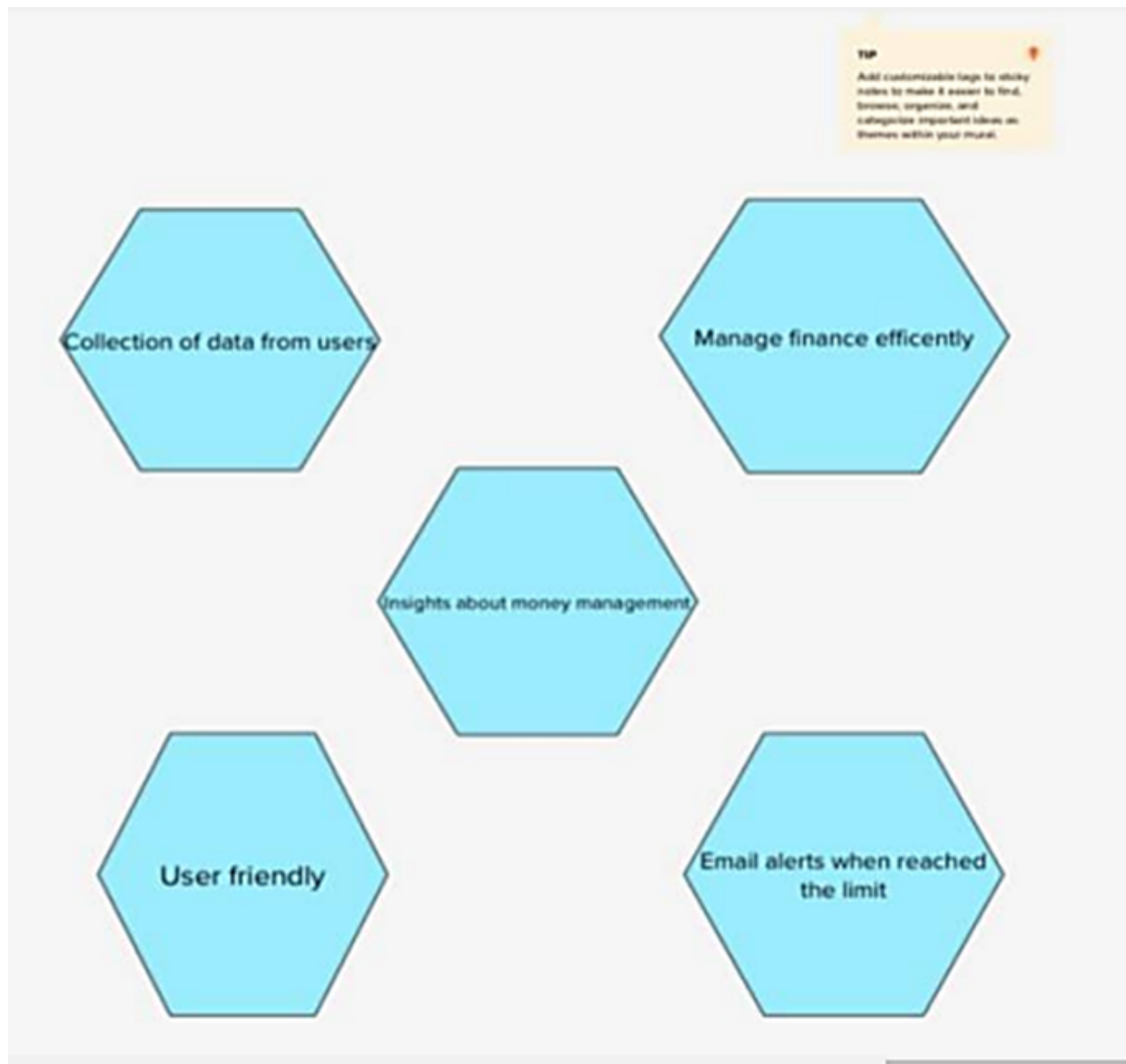


3.2 Ideation & Brainstorming

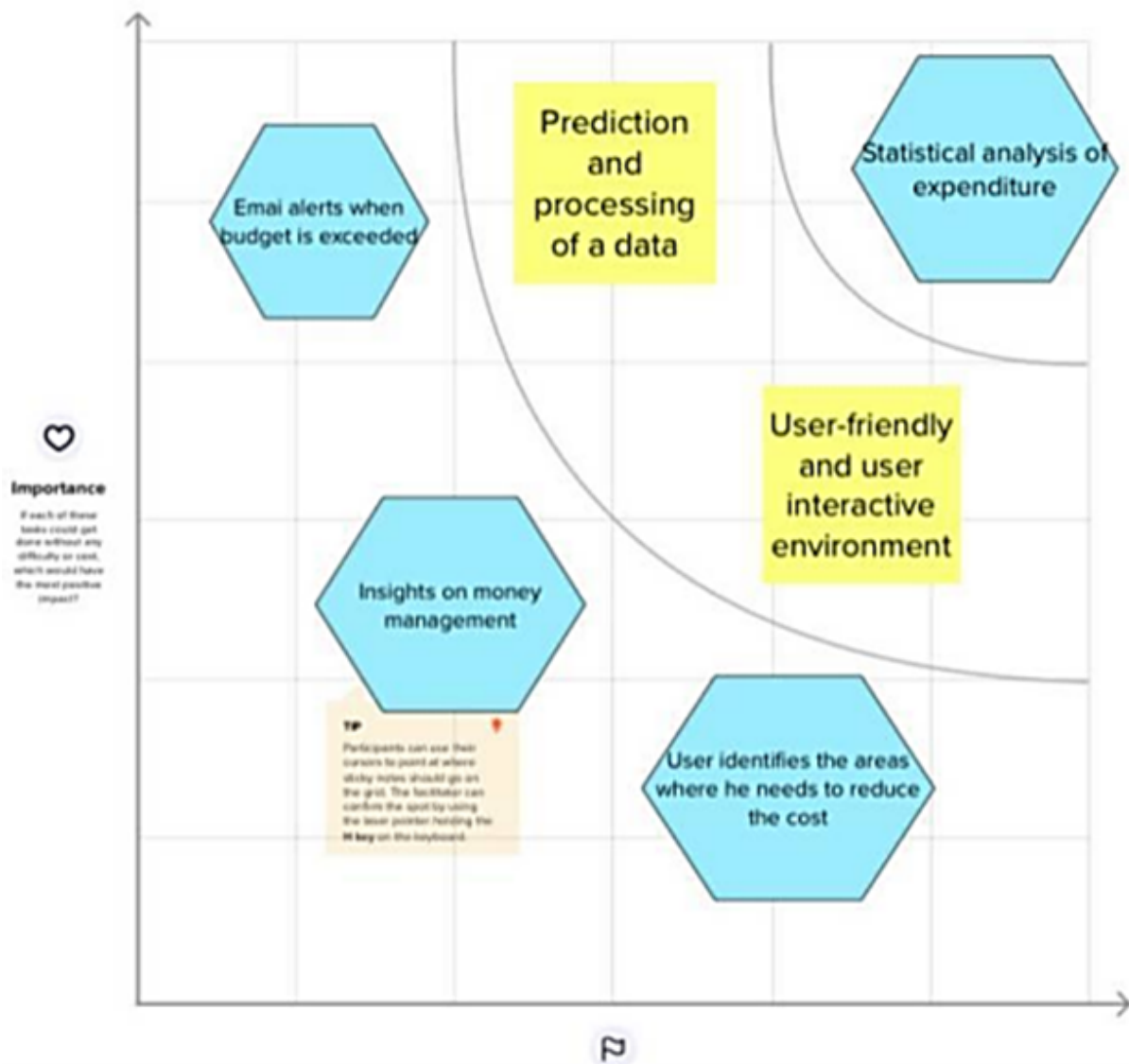
3.2.a Brainstorm



3.2.bGroup Ideas



3.2.c Prioritize



3.3 Proposed Solution

Project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Attempting to manage the expenses of an individual in an efficient and manageable manner, as compared to the traditional way of expense tracking.
2.	Idea / Solution description	The application will be helpful for the individuals in not just managing their expenses, but also in enabling them to improve their investments.
3.	Novelty / Uniqueness	The application gives the user a chance to plan his/her monthly expenses at the start of the month. Besides this, the user gets a notification when he/she exceeds the limit that is set.
4.	Social Impact / Customer Satisfaction	With such applications, the public will start to plan their expenses better leading to their own financial stability. With more users, this application will ensure that financial state of our society improves.
5.	Business Model (Revenue Model)	Free trial for 1 month can be given to the users, so that a significant userbase is created. Following the free trial, the users can be given subscription for 3 months, 6 months or 1 year.
6.	Scalability of the Solution	Since the application takes the same set of input from all the users and does not perform many complex computations, it will be easy for us to scale the application to a larger set of users.

3.4 Problem Solution fit

Define CS, fit into CL	1. CUSTOMER SEGMENT(S) CS Individuals who want to track their expenses (Working professionals, Students, etc.)	6. CUSTOMER LIMITATIONS CL <small>EG. BUDGET, DEVICES</small> <ul style="list-style-type: none"> • Device with internet connection • Subscription payment 	5. AVAILABLE SOLUTIONS AS <small>PROS & CONS</small> <ul style="list-style-type: none"> • Goodbudget - free but manual entry of transactions • Mint - free & wrong categorization • Spendee - many kind of integration but paid 	Explore AS, differentiate
	2. PROBLEMS / PAINS PR <small>+ ITS FREQUENCY</small> <ul style="list-style-type: none"> • People want to track their expenses everyday and • Some people also want to track it especially on some occasions 	9. PROBLEM ROOT / CAUSE RC <ul style="list-style-type: none"> • Its hard for the people to manage their financial expenses by keep tracking manually. • This makes them to exceed the actual budget that they made. 	7. BEHAVIOR BE <small>+ ITS INTENSITY</small> <ul style="list-style-type: none"> • People tend to avoid tracking the spendings and worry when they come to know about it. • People who try to do it manually will end up leaving some spendings. 	Focus on PR, tap into BE, understand RC
Identify strong TR & EM	3. TRIGGERS TO ACT TR <ul style="list-style-type: none"> • People are not aware about the existing solutions especially in India. • Providing a visualization about how they spend makes the people to decide easily. • Free for sometime and show the results.(retain & premium users) 	10. YOUR SOLUTION SL <ul style="list-style-type: none"> • Build an application to track their expenses seamlessly. • Allow users to access it for free and make them realize how useful to save money. 	8. CHANNELS of BEHAVIOR CH ONLINE <ul style="list-style-type: none"> • Social media advertisements (especially LinkedIn as the users are more of working professionals and students) • Lifestyle influencers would be a great choice OFFLINE <ul style="list-style-type: none"> • Word of mouth 	Extract online & offline CH of BE
	4. EMOTIONS EM <small>BEFORE / AFTER</small> <ul style="list-style-type: none"> • People are worried when the exceed the actual budget and not for saving it. • People will be able to track their expenses and bad situations related to financial management. 			

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 Functional Requirements

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Application Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User monthly expense tentative data	Data to be registered in the app
FR-4	User monthly income data	Data to be registered in the app
FR-5	Alert/ Notification	Alert through E-mail Alert through SMS
FR-6	User Budget Plan	Planning and Tracking of user expense vs budget limit

4.2 Non Functional Requirements

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

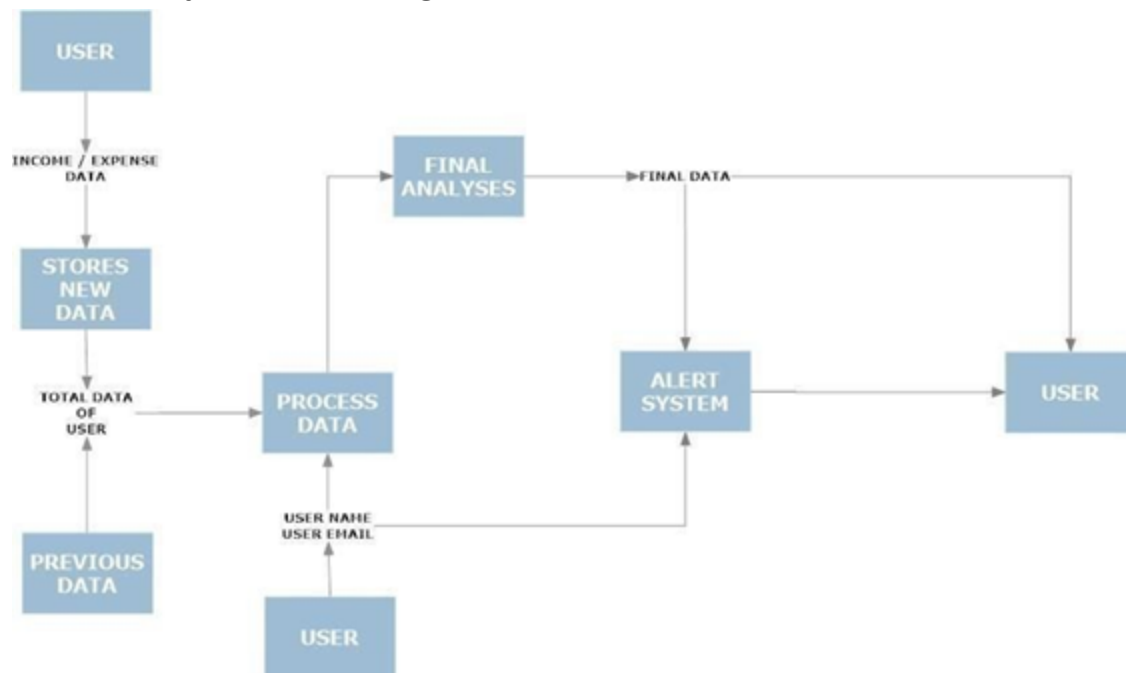
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Effectiveness, efficiency and overall satisfaction of the user while interacting with our application.
NFR-2	Security	Authentication, authorization, encryption of the application.
NFR-3	Reliability	Probability of failure-free operations in a specified environment for a specified time.
NFR-4	Performance	How the application is functioning and how responsive the application is to the end-users.
NFR-5	Availability	Without near 100% availability, application reliability and the user satisfaction will affect the solution.
NFR-6	Scalability	Capacity of the application to handle growth, especially in handling more users.

CHAPTER 5

PRODUCT DESIGN

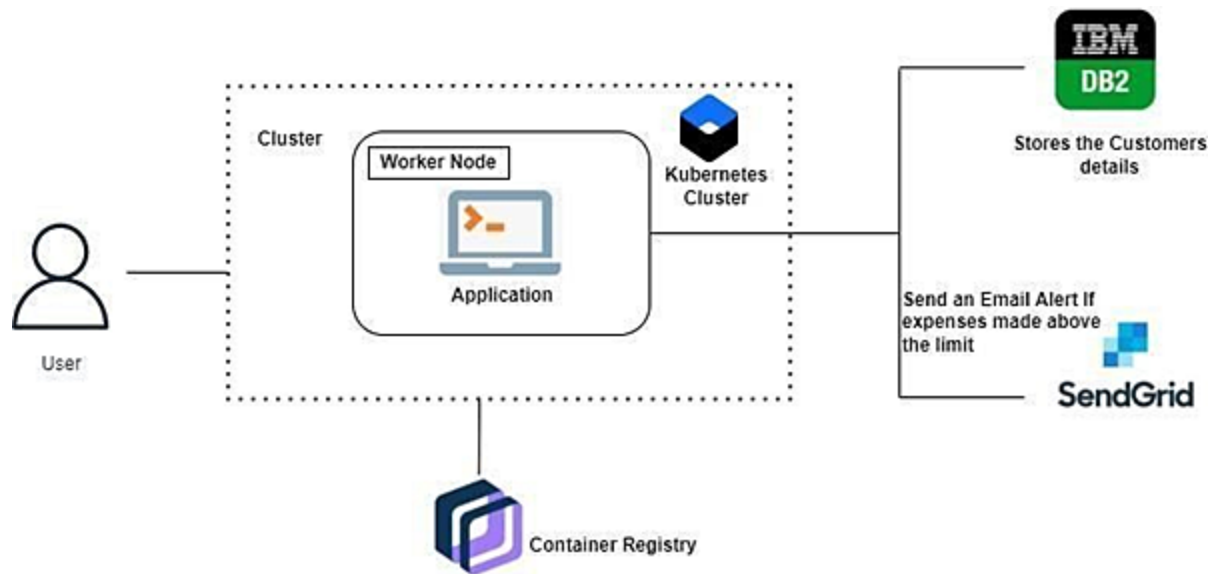
5.1 Data Flow Diagrams

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 Technical Architecture

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



5.3 User Stories

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

User Type	Functional Requirement (Epic)	User Story Number	User Story/ Task	Acceptance criteria	Priority	Release
Customer (Mobile user & web user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account /dashboard	High	

		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	
		USN- 3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	
	Login	USN -4	As a user, I can log into the application by entering email & password	I can access the application	High	
	Dashboard	USN -5	As a user I can enter my income and expenditure details.	I can view my daily expenses	High	
Customer Care Executive		USN 6	As a customer care executive I can solve the log in issues and other issues of the application.	I can provide support or solution at any time 24*7	Medium	

Administrator	Application	USN -7	As a administrat or I can upgrade or update the application.	I can fix the bug which arises for the customers and users of the application	Med iu m	
---------------	-------------	--------	--	--	-------------	--

CHAPTER 6

PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & 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	Register for the application by entering email, password, and confirming password.	2	High	Naveen kumar dugar
Sprint-1		USN-2	As a user will receive confirmation email once have registered for the application	1	High	Ajay.m
Sprint-2		USN-3	Register for the application through Facebook	2	Low	Selva kumar.s
Sprint-1		USN-4	User register for the application through Gmail	2	Medium	Kirubanithi .v
Sprint-1	Login	USN-5	User log into the application by entering	1	High	Ajay.m

			email & password			
Sprint-3	Dashboard	USN-6	Expenditure detailson the application	3	High	Selvakumar .s
Sprint-3	Limits	USN-6	User can set monthly expense limit so that receive a mail on exceeding	4	High	Naveen kumar dugar
Sprint-4	Reports	USN-6	View thegraphical form of expenses category wise	5	Medium	Kirubanithi .v

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	23 Oct 2022	28 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	30 Oct 2022	04 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	06 Nov 2022	11 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	13 Nov 2022	18 Nov 2022	20	19 Nov 2022

6.3 Reports from JIRA

 PETA-1 register for the application by entering email, password, and confirming password	IN PROGRESS ▾
 PETA-3 Register for the application through Facebook	TO DO ▾
 PETA-8 Expenditure details on the application	TO DO ▾
 PETA-9 View the graphical form of expenses category wise	TO DO ▾

CHAPTER 7

7.1 Feature 1

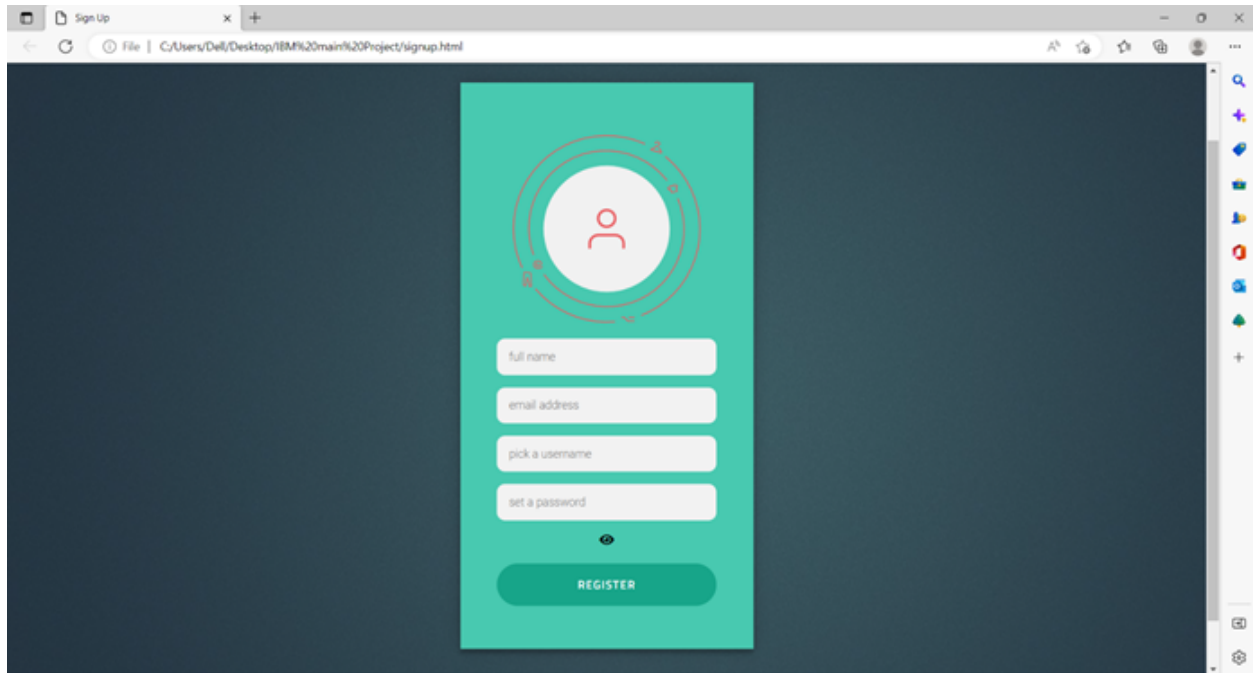
- i. Expense and revenue tracking.
- ii. Managing transaction receipts and records.
- iii. Paying taxes in time.
- iv. Processing payment and invoices.
- v. Create in-depth reports.

CHAPTER 8

TESTING

8.1 Test Cases

Registration Page



Sign Up

File | C:/Users/Dell/Desktop/18M%20main%20Project/signup.html

full name

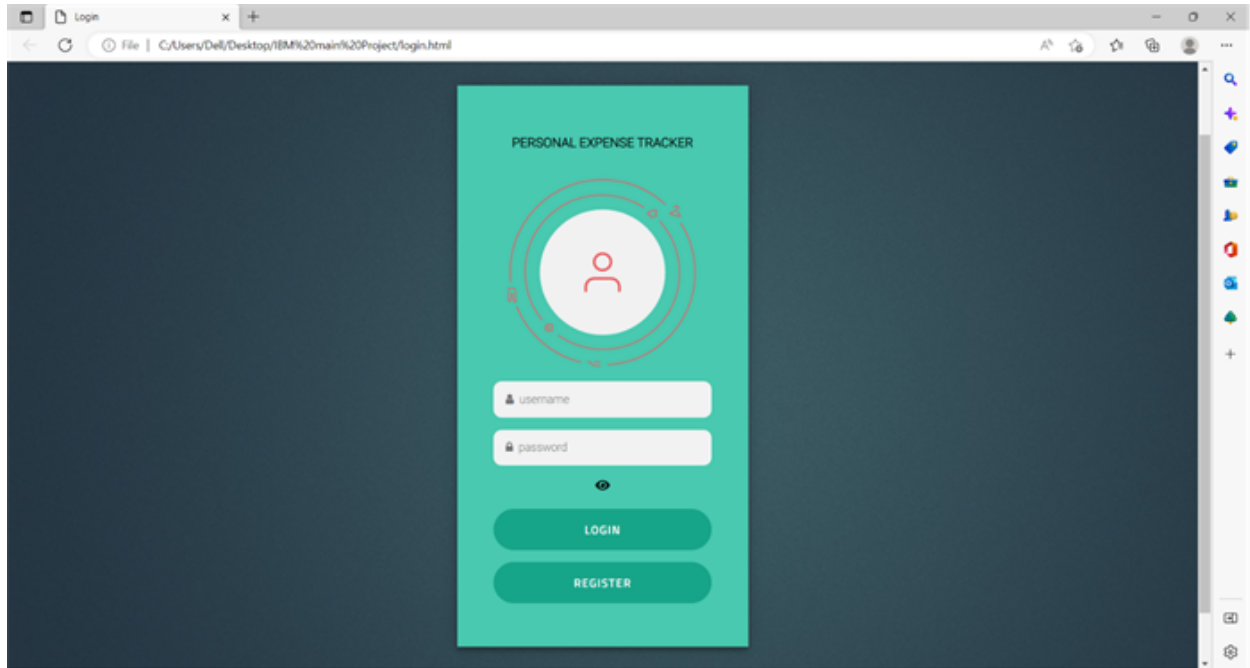
email address

pick a username

set a password

REGISTER

LOGIN



The image shows a web browser window with a single tab titled 'Login'. The address bar displays the file path 'C:/Users/Dell/Desktop/18M%20main%20Project/login.html'. The main content area features a dark blue background with a central teal-colored login form. At the top of the form, the text 'PERSONAL EXPENSE TRACKER' is displayed. Below this is a circular graphic with a red outline and a white center containing a red person icon. The form includes two input fields: 'username' and 'password', both with placeholder text. A small eye icon is positioned below the password field to toggle visibility. At the bottom of the form are two green buttons labeled 'LOGIN' and 'REGISTER'.

PERSONAL EXPENSE TRACKER

username

password

LOGIN

REGISTER

ADD EXPENSE

Expense Tracker App

File | C:/Users/Dell/Desktop/18M%20main%20Project/dashboard.html

Personal Expense Tracker

Add a new item:

Type: Choose one...

Place:

Date: dd-mm-yyyy

Amount:

Add a new expense

Type	Place	Date	Amount	Options
------	-------	------	--------	---------

CHAPTER 9

RESULTS

9.1 Performance Metrics

An application can be a very powerful tool for businesses if once the app becomes a success. However, the success of an app is measured through numbers, metrics, and analytics. Developing an app takes quite a lot, so once you've dedicated much time, money, and effort to the process, it's mandatory to measure mobile app performance.



CHAPTER 10

ADVANTAGES & DISADVANTAGES

10.1 Advantages

- i. Improved customer service
- ii. Cloud-based solution
- iii. Order Fulfillment
- iv. Harness Customer Loyalty and Retention

10.2 Disadvantages

- v. System Clash
- vi. Reduced Physical Audits
- vii. No solution to improve or eliminate bottlenecks in the servicecycle

CHAPTER 11

CONCLUSION

Taking proper care of our record is crucial in every business, no matter how big or little, we must understand. We must educate ourselves about the idea of effective inventory management and its applications because we can see that managers do not fully grasp it. A company's inventory management system is one of the reasons for its failure. Many customs to combat failure are present, and we can start from this point. Modern technologies can support us in managing and keeping an eye on our inventory. We may learn, put new ideas into practice, and assess our company.

CHAPTER 12

FUTURE SCOPE

1. A future update shall have payment option where we can pay dues and subscriptions .
2. Linking bank accounts and also tracking shares .It can be scaled for all types of people from any type of field .
3. In further days there will be mails and payment embedded with the app , also backup details will be recorded on cloud.
4. Here user can define their own categories for expense type like food,clothing,rent etc ..

CHAPTER 13

APPENDIX

SOURCECODE:

REGISTRATION PAGE

Register.html

```
<!DOCTYPE html>
<html lang="en">
<html>
<head>
  <title>Sign Up</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <meta charset="utf-8" />
  <link rel="stylesheet" type="text/css" href="register_style.css" />
  <link
    rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css"
  />
  <link
    href="https://fonts.googleapis.com/css?family=Titillium+Web:400,300,600"
    rel="stylesheet"
    type="text/css"
  />
  <script src="https://unpkg.com/@lottiefiles/lottie-player@latest/dist/lottie-player.js"></script>
  <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.15.1/css/all.css"
integrity="sha384-vp86vTRFVJgpfF9jiIGPEEqYqlDwgyBgEF109VFjmqGmIY/Y4HV4d3Gp2irVfcrp"
crossorigin="anonymous">
</head>

<body class="body">

  <div class="login-page">
    <div class="form">
      <form>
        <lottie-player

src="https://assets4.lottiefiles.com/datafiles/XRVouUu3IX4sGWtiC3MPpFnJvZNq7lVWDCa8LSqgS/prof
ile.json"
        background="transparent"
        speed="1"
        style="justify-content: center"
        loop
```

```

        autoplay
    ></lottie-player>
    <input type="text" placeholder="full name" />
    <input type="text" placeholder="email address" />
    <input type="text" placeholder="pick a username" />
    <input type="password" id="password" placeholder="set a password" />
    <i class="fas fa-eye" onclick="show()"></i>
    <br>
    <br>
</form>
<form class="login-form">
    <button type="button" onclick="window.location.href='login.html'">
        REGISTER
    </button>
</form>
</div>
</div>
</body>
<script>
function show() {
    var password = document.getElementById("password");
    var icon = document.querySelector(".fas");

    // ===== Checking type of password =====
    if (password.type === "password") {
        password.type = "text";
    } else {
        password.type = "password";
    }
}
</script>
</html>
</html>

```

register_style.css

```
@import url(https://fonts.googleapis.com/css?family=Roboto:300);
```

```
.body{
```

```
background-image: url('https://ordinaryfaith.net/wp-content/uploads/2016/03/Gray-plain-website-background.jpg');
background-repeat: no-repeat;
background-size: cover;
}
```

```
.login-page {
width: 360px;
padding: 8% 0 0;
margin: auto;
```

```
}
```

```
.form {
position: relative;
z-index: 1;
background: #48c9b0;
max-width: 360px;
margin: 0 auto 100px;
padding: 45px;
text-align: center;
box-shadow: 0 0 20px 0 rgba(0, 0, 0, 0.2), 0 5px 5px 0 rgba(0, 0, 0, 0.24);
```

```
}
```

```
.form input {
font-family: FontAwesome, "Roboto", sans-serif;
outline: 0;
background: #f2f2f2;
width: 100%;
border: 0;
margin: 0 0 15px;
```

```
padding: 15px;
box-sizing: border-box;
font-size: 14px;
border-radius: 10px;

}

.form button {
font-family: "Titillium Web", sans-serif;
font-size: 14px;
font-weight: bold;
letter-spacing: .1em;
outline: 0;
background: #17a589;
width: 100%;
border: 0;
border-radius: 30px;
margin: 0px 0px 8px;
padding: 15px;
color: #FFFFFF;
-webkit-transition: all 0.3 ease;
transition: all 0.3 ease;
cursor: pointer;
transition: all 0.2s;

}
```

```
.form button:hover, .form button:focus {
background: #148f77;
box-shadow: 0 5px 10px rgba(0, 0, 0, 0.2);
transform: translateY(-4px);
}

.form button:active {
transform: translateY(2px);
```

```
    box-shadow: 0 2.5px 5px rgba(0, 0, 0, 0.2);  
}
```

```
.form .message {
```

```
    margin: 6px 6px;  
    color: #808080;  
    font-size: 11px;  
    text-align: center;  
    font-weight: bold;  
    font-style: normal;
```

```
}
```

```
.form .message a {  
    color: #FFFFFF;  
    text-decoration: none;  
    font-size: 13px;  
}
```

```
.form .register-form {  
    display: none;  
}
```

```
.container {  
    position: relative;  
    z-index: 1;  
    max-width: 300px;  
    margin: 0 auto;  
}
```

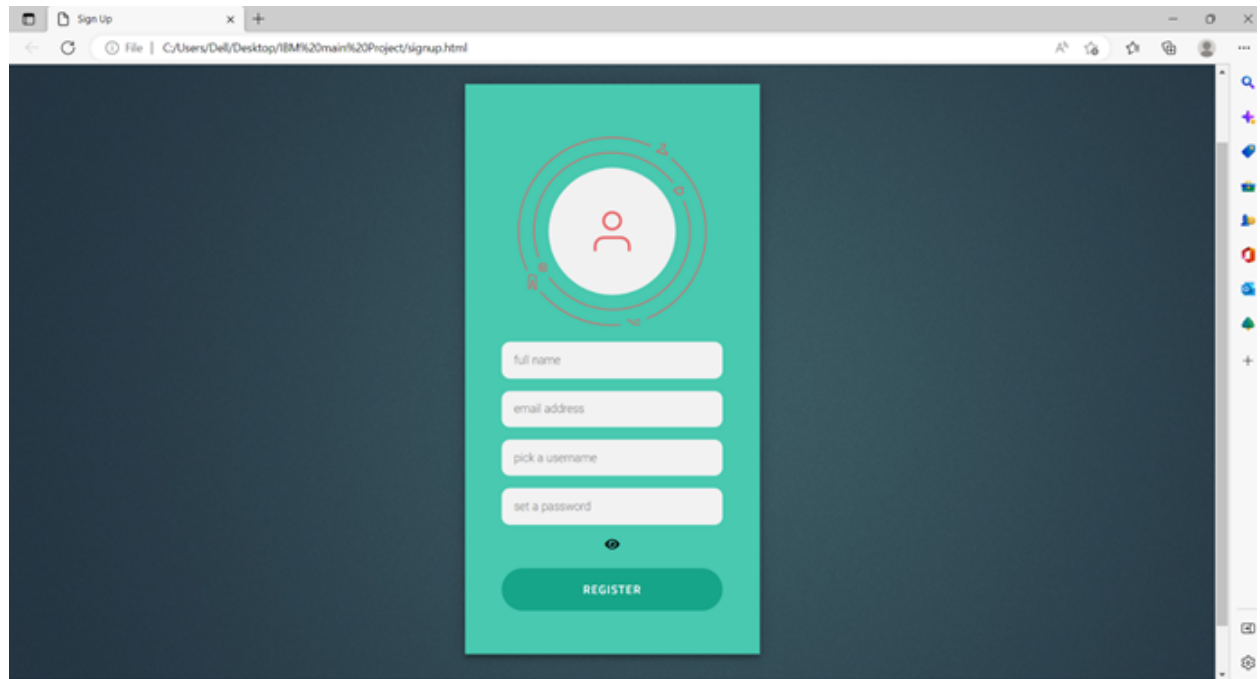
```
.container:before, .container:after {
```

```
content: "";
display: block;
clear: both;
}
.container .info {
margin: 50px auto;
text-align: center;
}
.container .info h1 {
margin: 0 0 15px;
padding: 0;
font-size: 36px;
font-weight: 300;
color: #1a1a1a;
}
.container .info span {
color: #4d4d4d;
font-size: 12px;
}
.container .info span a {
color: #000000;
text-decoration: none;
}
.container .info span .fa {
color: #EF3B3A;
}

body {
background: #76b852; /* fallback for old browsers */
background: -webkit-linear-gradient(right, #76b852, #8DC26F);
background: -moz-linear-gradient(right, #76b852, #8DC26F);
background: -o-linear-gradient(right, #76b852, #8DC26F);
background: linear-gradient(to left, #76b852, #8DC26F);
font-family: "Roboto", sans-serif;
```

```
-webkit-font-smoothing: antialiased;  
-moz-osx-font-smoothing: grayscale;  
}
```

OUTPUT:



LOGIN PAGE

Login.html


```
<!DOCTYPE HTML>
<html lang="en" >
<html>
<head>
  <title>Login</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta charset="utf-8">
  <link rel="stylesheet" type="text/css" href="login_style.css">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
  <link href='https://fonts.googleapis.com/css?family=Titillium+Web:400,300,600'
rel='stylesheet' type='text/css'>
  <link href='https://fonts.googleapis.com/css?family=Titillium+Web:400,300,600'
rel='stylesheet' type='text/css'>
  <script src="https://unpkg.com/@lottiefiles/lottie-player@latest/dist/lottie-
player.js"></script>
  <link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.15.1/css/all.css" integrity="sha384-
vp86vTRFVJgpjF9jiIGPEEqYqlDwgyBgEF109VFjmqGmIY/Y4HV4d3Gp2irVfcr
p" crossorigin="anonymous">
</head>

<body class="body">

  <div class="login-page">
    <div class="form">
      <p class="para"><b>PERSONAL EXPENSE TRACKER</b></p>
      <form method="POST">
        <lottie-player
src="https://assets4.lottiefiles.com/datafiles/XRVouU3IX4sGWtiC3MPpFnJvZNq7
IVWDCa8LSqgS/profile.json" background="transparent" speed="1"
style="justify-content: center;" loop autoplay></lottie-player>
```

```
<input type="text" placeholder="&#xf007; username"/>
<input type="password" id="password" placeholder="&#xf023; password"/>
<i class="fas fa-eye" onclick="show()"></i>
<br>
<br>
<button type="button"
onclick="window.location.href='dashboard.html'">LOGIN</button>
<p class="message"></p>
```

```
</form>
```

```
<form class="login-form">
  <button type="button"
onclick="window.location.href='register.html'">REGISTER</button>
</form>
</div>
</div>
```

```
<script>
function show(){
  var password = document.getElementById("password");
  var icon = document.querySelector(".fas")

  // ===== Checking type of password =====
  if(password.type === "password"){
    password.type = "text";
  }
  else {
    password.type = "password";
  }
};
</script>
```

```
</body>
</html>
```

login_style.css

```
@import url(https://fonts.googleapis.com/css?family=Roboto:300);

.body{
  background-image: url('https://ordinaryfaith.net/wp-
content/uploads/2016/03/Gray-plain-website-background.jpg');
  background-repeat: no-repeat;
  background-size: cover;
}

.login-page {
  width: 360px;
  padding: 8% 0 0;
  margin: auto;
}

.form {
  position: relative;
  z-index: 1;
  background: #48c9b0;
  max-width: 360px;
  margin: 0 auto 100px;
  padding: 45px;
  text-align: center;
  box-shadow: 0 0 20px 0 rgba(0, 0, 0, 0.2), 0 5px 5px 0 rgba(0, 0, 0, 0.24);
}
```

```
.form input {  
  font-family: FontAwesome, "Roboto", sans-serif;  
  outline: 0;  
  background: #f2f2f2;  
  width: 100%;  
  border: 0;  
  margin: 0 0 15px;  
  padding: 15px;  
  box-sizing: border-box;  
  font-size: 14px;  
  border-radius: 10px;  
  
}
```

```
.form button {  
  font-family: "Titillium Web", sans-serif;  
  font-size: 14px;  
  font-weight: bold;  
  letter-spacing: .1em;  
  outline: 0;  
  background: #17a589;  
  width: 100%;  
  border: 0;  
  border-radius: 30px;  
  margin: 0px 0px 8px;  
  padding: 15px;  
  color: #FFFFFF;  
  -webkit-transition: all 0.3 ease;  
  transition: all 0.3 ease;  
  cursor: pointer;  
  transition: all 0.2s;
```

```
}  
.form button:hover,.form button:focus {  
  background: #148f77;  
  box-shadow: 0 5px 10px rgba(0, 0, 0, 0.2);  
  transform: translateY(-4px);  
}  
.form button:active {  
  transform: translateY(2px);  
  box-shadow: 0 2.5px 5px rgba(0, 0, 0, 0.2);  
}
```

```
.form .message {  
  
  margin: 6px 6px;  
  color: #808080;  
  font-size: 11px;  
  text-align: center;  
  font-weight: bold;  
  font-style: normal;  
}  
.form .message a {  
  color: #FFFFFF;  
  text-decoration: none;  
  font-size: 13px;  
}
```

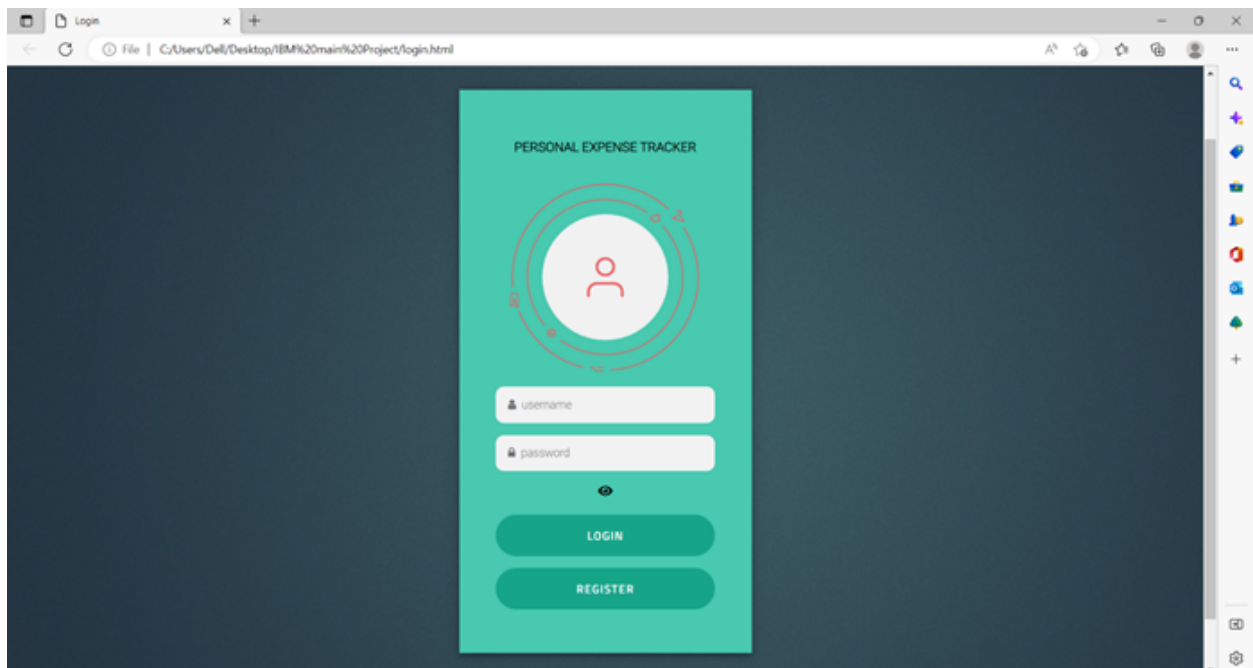
```
.form .register-form {  
  display: none;  
}
```

```
.container {  
  position: relative;  
  z-index: 1;  
  max-width: 300px;
```

```
margin: 0 auto;
}
.container:before, .container:after {
  content: "";
  display: block;
  clear: both;
}
.container .info {
  margin: 50px auto;
  text-align: center;
}
.container .info h1 {
  margin: 0 0 15px;
  padding: 0;
  font-size: 36px;
  font-weight: 300;
  color: #1a1a1a;
}
.container .info span {
  color: #4d4d4d;
  font-size: 12px;
}
.container .info span a {
  color: #000000;
  text-decoration: none;
}
.container .info span .fa {
  color: #EF3B3A;
}
body {
  background: #76b852; /* fallback for old browsers */
  background: -webkit-linear-gradient(right, #76b852, #8DC26F);
```

```
background: -moz-linear-gradient(right, #76b852, #8DC26F);
background: -o-linear-gradient(right, #76b852, #8DC26F);
background: linear-gradient(to left, #76b852, #8DC26F);
font-family: "Roboto", sans-serif;
-webkit-font-smoothing: antialiased;
-moz-osx-font-smoothing: grayscale;
}
```

OUTPUT:



EXPENSE PAGE

dashboard.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Expense Tracker App</title>
  <link
href="https://fonts.googleapis.com/css?family=Titillium+Web&display=swap"
  rel="stylesheet">
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <header>
    <h1 class="title"><b>Personal Expense Tracker </b></h1>
  <header>

  <section class="content">
    <h3 class="secondTitle">Add a new item: </h3>
    <div class="form">
      <form id="expForm">
        <div class="formLine left">
          <span for="type">Type:</span>
          <select id="type">
            <option value="chooseOne">Choose one...</option>
            <option value="Food">Food</option>
            <option value="Petrol">Petrol</option>
            <option value="Travel">Travel</option>
            <option value="Shopping">Shopping</option>
            <option value="Savings">Savings</option>
            <option value="Snaks">Snaks</option>
            <option value="Other">Other</option>
```



```

        </select>
    </div>
    <div class="formLine right">
        <span for="name">Name:</span>
        <input type="text" id="name">
    </div>

    <div class="formLine left">
        <span for="date">Date:</span>
        <input type="date" id="date">
    </div>
    <div class="formLine right">
        <span for="amount">Amount:</span>
        <input type="text" id="amount">
    </div>
    <button type="submit" class="buttonSave">Add a new
expense</button>
</form>
</div>
</section>
<section class="content">
    <table class="table">
        <thead>
            <tr>
                <th>Type</th>
                <th>Name</th>
                <th>Date</th>
                <th>Amount</th>
                <th>Options</th>
            </tr>
        </thead>
        <tbody id="expenseTable">

```

```
        </tbody>
    </table>
</section>
<script src="app.js"></script>
</body>
</html>
```

style.css

```
*{
    margin:0;
    font-family: 'Titillium Web', sans-serif;
}
```

```
body{
    background:#042159;
    color:#77D9AB;
}
```

```
.title{
    width: 500px;
    color:#0869A6;
    margin:3% auto;
}
```

```
.content{
    width: 580px;
    margin: 0 auto;
```

```
background-color:#0869A6;
padding: 3%;
padding-left: 6%;
}
```

```
.secondTitle{
    text-align:left;
    margin: 2% 0;
    font-weight: 100;
}
```

```
.form{
    padding: 5px;
}
```

```
.formLine{
    display: inline-flex;
    padding: 5px 0px;
}
```

```
.left{
    float: left;
}
```

```
.right{
    float:right;
    margin-right: 100px;
}
```

```
input, select{
    width: 130px;
    margin-left: 10px;
```

```
}
```

```
/* table style */
```

```
table{
```

```
    width: 100%;
```

```
}
```

```
thead{
```

```
    background-color:#042159;
```

```
    line-height: 30px;
```

```
}
```

```
/* Button */
```

```
button{
```

```
    width: 200px;
```

```
    color:#fff;
```

```
    padding: 10px;
```

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

```
    font-size: 1.1em;
```

```
    line-height: 20px;
```

```
    background-color: #04BF8A;
```

```
    border-radius: 5px;
```

```
    margin: 14px 25%;
```

```
    cursor: pointer;
```

```
}
```

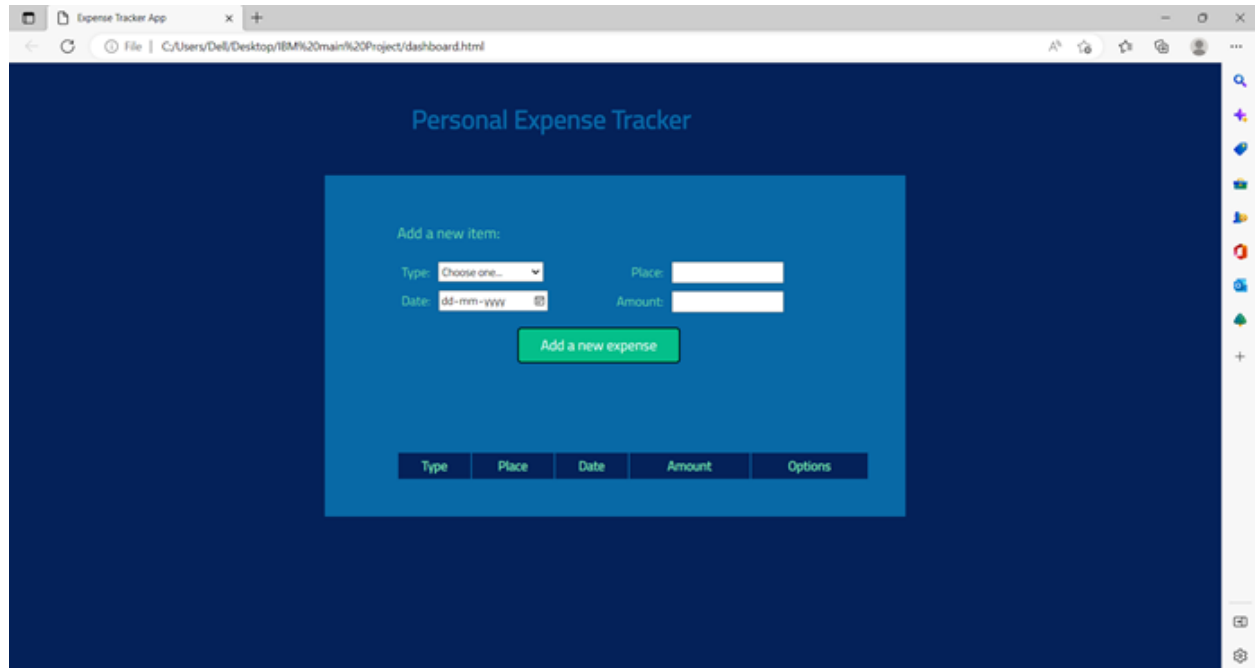
```
a{
```

```
    text-decoration: underline;
```

```
    cursor: pointer;
```

```
}
```

Output:



DB FILE :

```
from flask import session
```

```
import ibm_db
```

```
import hashlib
```

```
import datetime
```

```
import os
```

```
class Db:
```

```
    def __init__(self) -> None:
```

```
        hostname="ea286ace-86c7-4d5b-8580-  
3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud"
```

```
        uid="ljr08913"
```

```
        pwd="0wfoXAtjinaXZZxO"
```

```
        driver="{IBM DB2 ODBC DRIVER}"
```

```
        db="bludb"
```

```
        port="31505"
```

```
        cert="DigiCertGlobalRootCA.crt"
```

```
dsn=(  
  
    "DATABASE={0};"  
  
    "HOSTNAME={1};"  
  
    "PORT={2};"  
  
    "UID={3};"  
  
    "SECURITY=SSL;"  
  
    "SSLServerCertificate={4};"  
  
    "PWD={5};"  
  
).format(db,hostname,port,uid,cert,pwd)  
  
self.conn=ibm_db.connect("dsn","",")
```

```
def generateId(self) -> str:  
  
    return hashlib.md5("{}{}{}".format(  
  
        session["active"],
```

```
datetime.datetime.now().strftime('%m%d%Y%H%M%S%f')  
  
).encode()).hexdigest()
```

```
def execute(self, query: str) -> bool:
```

```
    try:
```

```
        ibm_db.exec_immediate(self.conn, query)
```

```
    return True
```

```
    except:
```

```
        print("SQLSTATE = {}".format(ibm_db.stmt_error()))
```

```
    return False
```

```
def get(self, table_name: str, condition: str, columns : str = "*") -> tuple:
```

```
    try:
```

```
        query = f"SELECT {columns} FROM {table_name} WHERE  
{condition}"
```



```
print(query)
```

```
stmt = ibm_db.exec_immediate(self.conn, query)
```

```
return ibm_db.fetch_tuple(stmt)
```

```
except:
```

```
print("SQLSTATE = {}".format(ibm_db.stmt_error()))
```

```
return ()
```

```
def getall(self, table_name: str, condition: str, columns : str = "*") -> list:
```

```
    query = f"SELECT {columns} FROM {table_name} WHERE  
{condition}"
```

```
print(query)
```

```
stmt = ibm_db.exec_immediate(self.conn, query)
```

```
data = []
```

```
while True:
```

```
    temp = ibm_db.fetch_tuple(stmt)
```

```
if temp != False:
```

```
    data.append(temp)
```

```
else:
```

```
    break
```

```
return data
```

```
def delete(self, table_name:str, condition: str) -> bool:
```

```
    query = f"DELETE FROM {table_name} WHERE {condition}"
```

```
    print(query)
```

```
    return self.execute(query)
```

```
def insert(self, table_name: str, values: list) -> bool:
```

```
    try:
```

```
        valuestup = ','.join("{}{}{}".format(x) for x in values)
```

```
        query = f"INSERT INTO {table_name} VALUES ({valuestup})"
```

```
print(query)
```

```
return self.execute(query)
```

```
except Exception as e:
```

```
    print(e)
```

```
return False
```