

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

As the name itself suggests, this project is an attempt to manage our daily expenses in a more efficient and manageable way. Sometime we cannot remember where our money goes. And we cannot handle our cash flow. For this problem, we need a solution that everyone can manage their expenses. So, we decided to find an easier way to get rid of this problem. So, our application attempts to free the user with as much as possible the burden of manual calculation and to keep the track of the expenditure.

OBJECTIVE

Instead of keeping a diary or a log of the expenses, this application enables the user to not just keep the control on the expenses but also to generate and save reports. With the help of this application, the user can manage their expenses on a daily, weekly, and monthly basis. Users can insert and delete transactions as well as can generate and save their reports. The graphical representation of the application is the main part of the system as it appeals to the user more and is easy to understand.

1.2 PURPOSE

Using this application, users can manage all financial data and track all expense and income category wise.

- Creating a category and recording all expenses and income under the category.

- Enable the notification system user get notification daily at a specific time that

can help the user insert expense and income.

- Backup and Restore all information.

- Report are generated in PDF format in category wise or time period.

Using this application, users can manage all financial data and track all expense and income category wise.

- Creating a category and recording all expenses and income under the category.

- Enable the notification system user get notification daily at a specific time that

can help the user insert expense and income.

- Backup and Restore all information.

- Report are generated in PDF format in category wise or time period.

Using this application, users can manage all financial data and track all expense and income category wise.

An expense tracker app also helps you to pay your taxes in time and stay updated on tax deductions. All you need is to upload the taxation documents to the app. The expense tracker apps will help you stay free of worries by organizing your tax return spending into appropriate categories.

CHAPTER 2

LITERATURE SURVEY

The purpose of the Literature Survey is to give a brief overview and also to develop complete knowledge about the reference papers. The goal of the Literature Survey is to completely specify the technical details related to the main project concisely and unambiguously.

2.1 EXISTING PROBLEM

When a process is completely dependent on human effort, there are bound to be inadvertent errors. Fraudulent practices are all too common amidst chaos. Both paper-based and digital processes are often rife with approval delays. Expense management policies are hard to enforce in manual processes. There isn't a lot of transparency. With businesses being spread across geographies and employees traveling extensively for work, manual processes can cause hurdles due to limited access.

PROPOSED SYSTEM

Currently proposed system is a Cloud based expense management tools allow access on the go with mobile apps. A central database makes it possible to approve claims, submit bills, and more from any location or device.

2.2 REFERENCES

A. Intelligent Online Budget Tracker:

The development of this application has been conducted in a stepwise manner using

the well-defined methodology, RUP, customized according to the requirements of the system. Most of the goals set at the start of the development phase have been met. Security problems like web security or network security have also been treated

in the design and development of the system, thus increasing the reliability of the system. Quality management issues have also been handled satisfactorily.

B. Online Income and Expense Tracker:

This project is work more efficient than the other income and expense tracker.

The project successfully avoids the manual calculation for calculating the income and expense per month. The modules are developed efficiently and also in an attractive manner.

C. Family Expense Manager Application:

As the result, the user can make use of this application in his/her daily life. After being used it can be a part of daily life to update and view daily expenses and family expenses. This helps to keep track of expenses & manage it for the user as they are busy in their daily routine, they are not able to keep track of their incomes & expenses.

D. Personalized Expense Managing Assistant Using Android:

Some of the features are like enabling users to register to the application using an existing email or social network account, it will synchronize the user's profile information to the application. Apart from this, the application can be used to gather samples of data related to user's expenses with consents and use those sample data as parameters to assess patterns of spending. Using some data mining techniques expenses can be classified and can be used in market analysis and planning.

E. Mobikwik Expense Tracking Application:

Mobikwik came up with a new feature in their app called Expense Manager. With this feature, you can track and manage your expenditures (expenses), savings, reminders and bill payments. This is a

personal budget management app that tracks your expenditures and income and gives you recommendations to make you economically strong. The main idea of developing this feature for giving users a clear picture that how much they are spending and where they are spending and when. We remind them to pay their utilities and card bills before the due date by using the same platform in just one tap, instead of going any other way. Also serving them by giving saving tips for their good future investment.

F. Expense Tracker:

This Expense Tracker is a web application that facilitates the users to keep track and manage their personal as well as business expenses. This application helps the users to keep a digital diary. It will keep track of a user's income and expenses on a daily basis. The user will be able to add his/her expenditures instantly and can review them anywhere and anytime with the help of the internet.

G. Student Expense Tracking Application:

It is like a digital diary that keeps record of expenses done by a student.

The application keeps track of money spent and the earnings both of the student on day-to-day basis. This application helps the user to easily calculate his/her expenses on daily basis and he/she has not to write down every transaction on paper and do the hectic calculation, it simply makes the task easier.

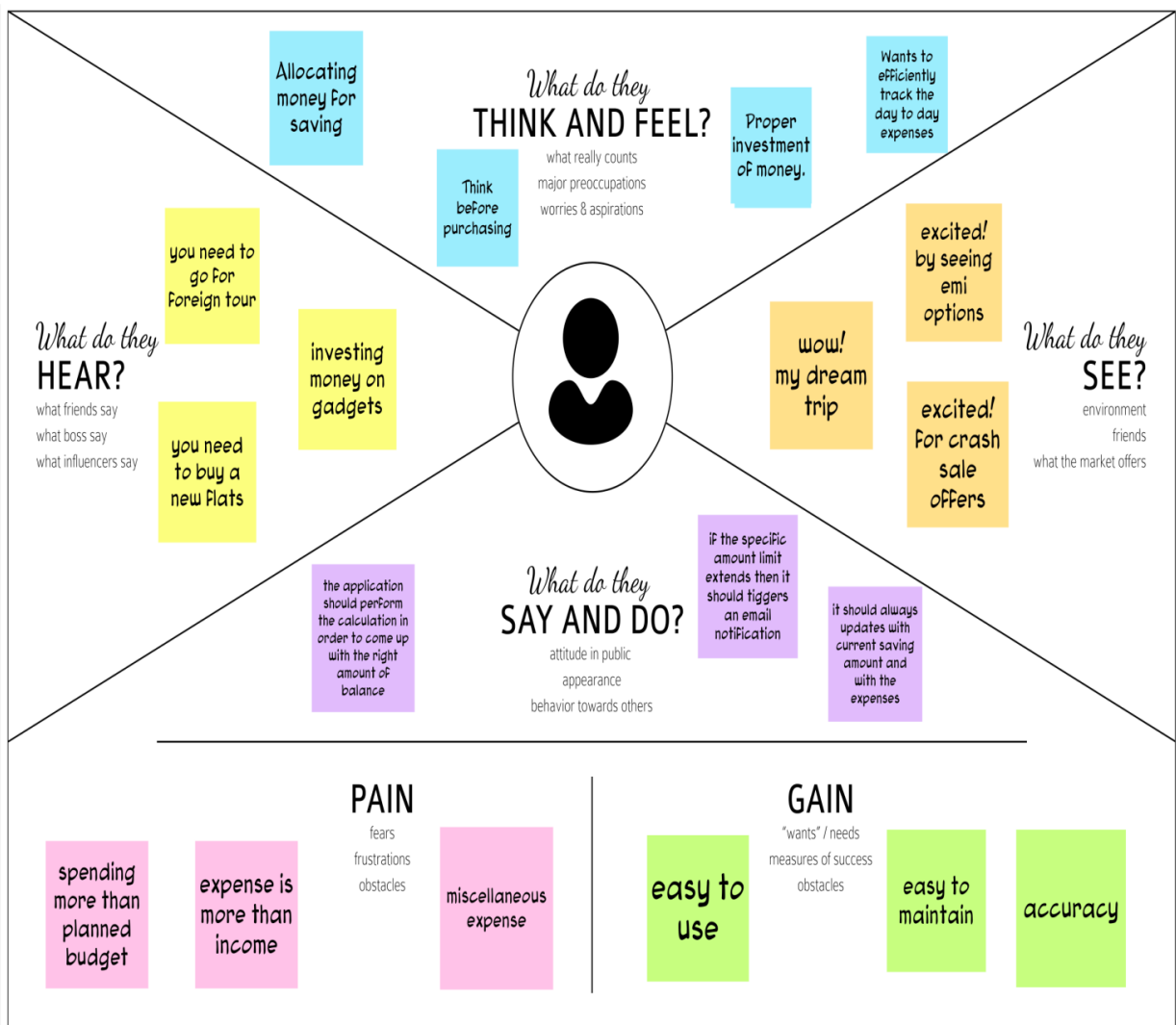
2.3 PROBLEM STATEMENT

The Expense Tracker applications is basically used in our lives on how to manage the money efficiently. By using these applications, we can easily come up with the solution in what are all the ways to optimize our money effectively. But now all the applications that are available in the market are unable to keep track of it. The applications that we will build is to manage our money and finance in a beneficial way. The main functionality is to build with this is to notify with an email alert if they exceed in their limit range of spending money

CHAPTER 3

IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas:



3.2 Ideation & Brainstorming:



Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

🕒 10 minutes to prepare
🕒 1 hour to collaborate
👤 2-8 people recommended



Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

🕒 10 minutes

A

Team gathering

Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B

Set the goal

Think about the problem you'll be focusing on solving in the brainstorming session.

C

Learn how to use the facilitation tools

Use the Facilitation Superpowers to run a happy and productive session.

[Open article](#) →

1

Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

🕒 5 minutes

PROBLEM

How might we [your problem statement]?



Key rules of brainstorming

To run a smooth and productive session



Stay in topic.



Encourage wild ideas.



Defer judgment.



Listen to others.



Go for volume.



If possible, be visual.

2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

🕒 10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

KEERTHANA

Handle documents	Avoid manual errors	Report Generation
Data Analysis	Investments Planning	Budget Calculation
Processing of Data	User Friendly	Payment Remainder

HARIPRIYA

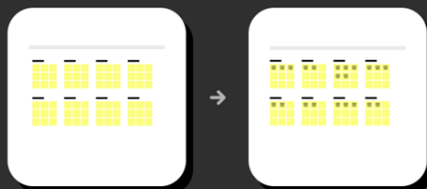
Prevent Data Losses	High Accuracy	Accurate record of money flow
Instant Solution	Efficient in Reducing Taxes	Savings plans
AI Based Bots and Advisors	Proper Guidelines	Record Expense

NANTHINEE

Track Receipts	Analyze day-to-day data	Meet the Financial Objectives
Accurate Results	Monitor Credit Score	Keep Track of Expenses
Record Every Transactions	Time Saving Method	Reveal Spending Issues

SRI ISHWARYA

More Efficient	Implementation is Easy	Budget Monitoring
Finance Management	Precised Result with All Possible Issues	Quick Solution
Alert Notification	Invoices and Bills	Easier with Software than Manual



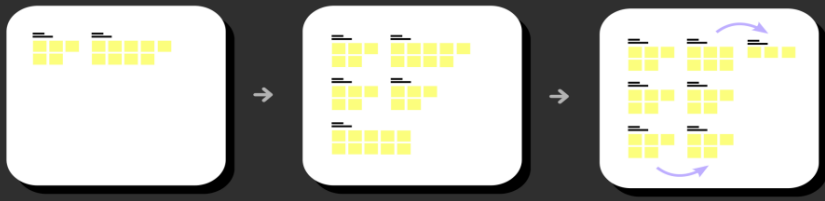
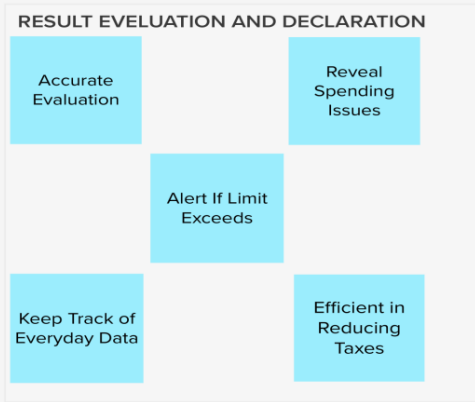
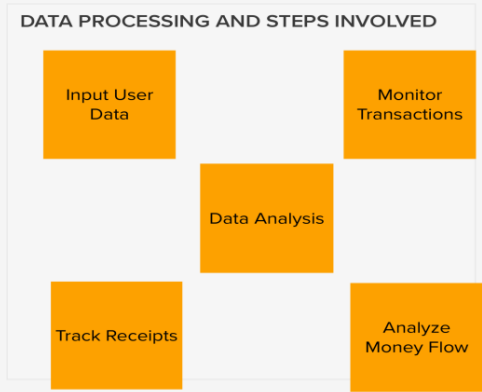
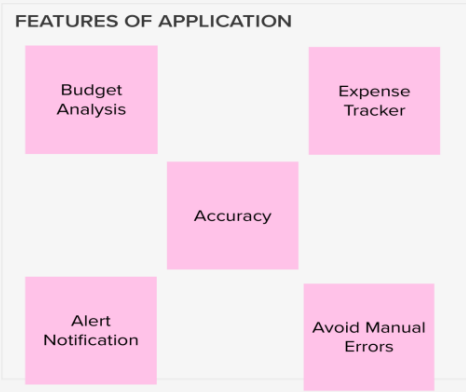
3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you can break it up into smaller sub-groups.

🕒 20 minutes

TIP
Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

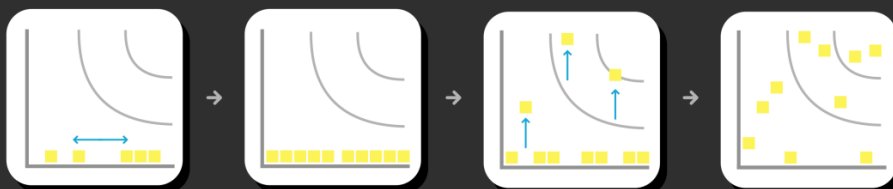
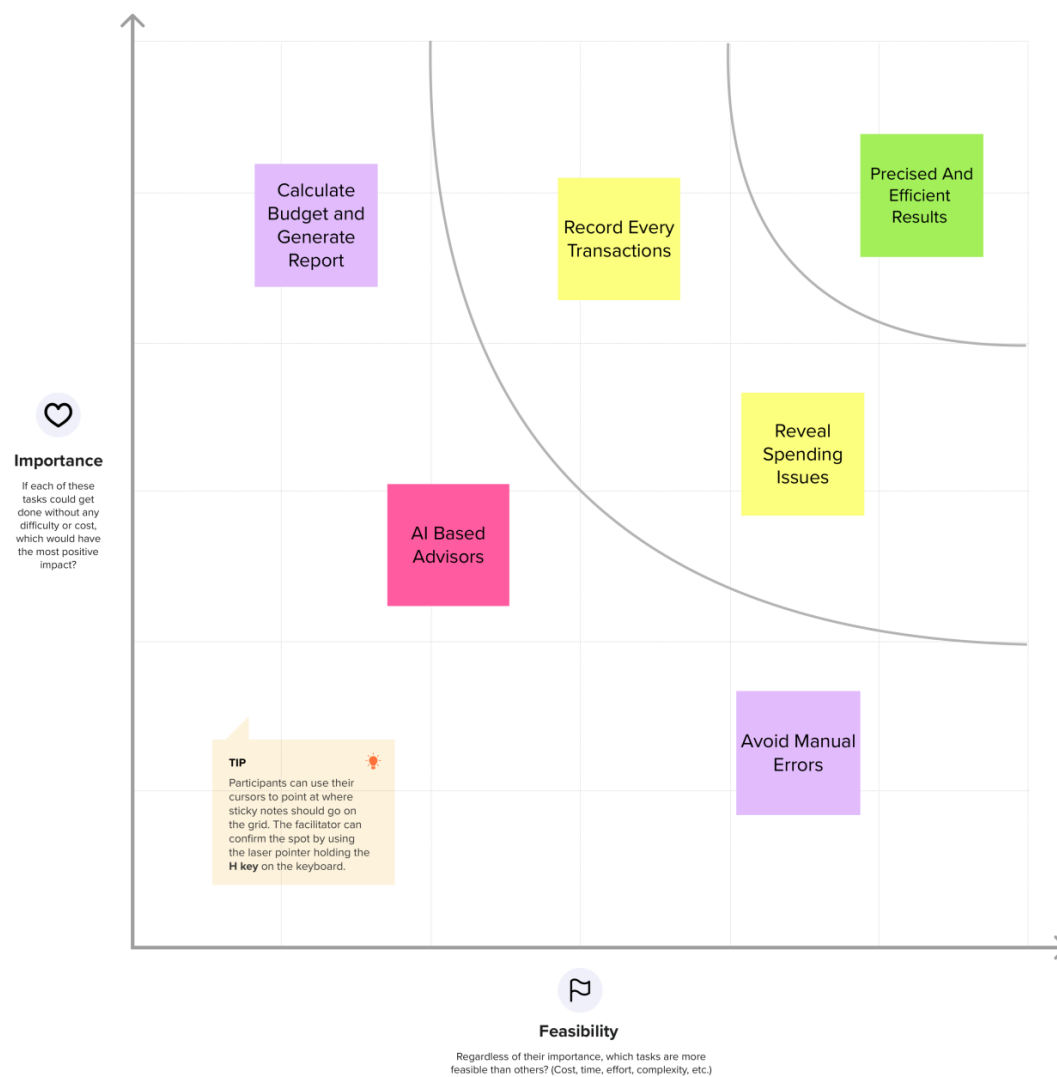


4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

🕒 20 minutes



3.3 Proposed Solution:

1.Problem Statement (Problem to be solved)

The Expense Tracker applications is basically used in our lives on how to manage the money efficiently. By using this applications, We can easily comes up with the solution in what are all the ways to optimize our money effectively. But now all the applications that are available in the market are unable to keep track of it. The applications that we will build is to manage our money and finance in a beneficial way. The main functionality is to build with this is to notify with an email alert if they exceeds in their limit range of spending money.

2.Idea / Solution description

Managing money more efficiently by using this application and it should be more user friendly. Utilize this application in a proper way which wants to make me comfortable and always keeps to track on the money in every expenditure.

3.Novelty / Uniqueness

By using this application it will alert you with a mail or alarm which will be useful for the customer to know whether the limit has been crossed or not.

4.Social Impact / Customer Satisfaction

Customer was able to find his expenses and spends. To Shows income, expenses, savings goals, credit score, investments, net worth etc. Ease of use of app and smooth ui. App provides real-time updates of how your transactions impact your budget and personalized reports are provided according to it. Warnings are given when monthly or weekly or daily expenses are crossed beyond the spending limit.

5.Business Model (Revenue Model)

Advertise about the app and the cool features available. Pro versions have extra benefits such as mention some unique things Ads are provided during the exit of app for income. Paid versions are also available to access other features such as linking bank accounts and direct access of credits. Free access is given for report generation of monthly spends. Multiple bank reports will be available only in paid versions.

6.Scalability of the Solution

To increase customers by advertising and marketing. Provide more access to the app and simple app configuration to easy usage.

3.4 Problem Solution fit:

Problem-Solution fit canvas 2.0		Purpose / Vision		
Define CS, fit into CC	1. CUSTOMER SEGMENT(S) CS Who is your customer? i.e. working parents of 0-5 y.o. kids The person who wants to track their daily expenses and to maintain their income within their allocated budget. To more concern, who wants to manage their financial status in a better way and in an efficient manner. Majorly, it appeals with the people who needs to save their money without any unwanted expenses.	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. The customer constraint is to find a suitable application, they need to analyze their expense which they had spent. This application will definitely helps to the user to track their progress and also if the user's budget exceeds then it will notifies with an message. Hence, this will makes their money management in an efficient way.	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 The application possess currently with platform oriented, and costly in their subscription amount and also most of the applications have not user friendly with an illiterate person.	Explore AS, differentiate
	2. JOBS-TO-BE-DONE / PROBLEMS J&P Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides. The main problem is to find the best application to track the customer's expense which are all they spend in their day-to-day life. The solution should be easily visualize to the persons in a statistical, pictorial, graphical representations.	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. The person who wants to save their money and the important point is more than that, their expenses it should not exceeds the limit of their income. By viewing the periodical balance in our total amount, we can make our future savings and the amount could be easily manage with the remaining expenses. And that is useful for the people to manage it in a efficient way and that helps the customer what are the essential needs and how to utilize the remaining balance amount efficiently.	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) The preliminary work of the customer on order to get the job done, he should noted the expenses for what they all use their money. They should maintain the analysis periodically. By taking the list, they can easily came to the point where they are investing their money, if the particular area exceeds means they can easily reduce their expenditure amount and they can save their money for further use.	
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. The benefits which are availed by an individual user, triggers the other individual to make use of an expense tracker to track their expenses and makes them to manage their finance and savings in an efficient way, which helps them to lead their life in a better way.	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, solve a problem and matches customer behaviour. All the financial decisions and activities that you make are unable to keep a track of it. This application makes your life easier by helping you to manage your finances efficiently. A personal finance application will not only help you with budgeting and accounting but also give you helpful insights about financial management. This also helps the customer if the amount exceeds then it will notifies the user.	8. CHANNELS of BEHAVIOUR CH 8.1 ONLINE What kind of actions do customers take online? Extract online channels from #7 By the involvement of everything at online makes the customer work easy and they can learn and also avail their needs in a easy way, at any time and at anywhere. The online mode mainly deals with the involvement of internet, without internet cannot able to get the job done. 8.2 OFFLINE What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development. By the mode of offline, they can engaged with more interactive and they can use it more activations without the involvement of internet. In this, the person can make their calculations more easy with the need of internet.	Extra online & offline CH & BE	
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 this application, the user may not aware about their expenses and they did not know their limit of using their income. After using this application, the person can able to manage their expenses and they visualize their progress amount in user friendly manner.	Identify strong TR & EM			

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

1. User Registration through Form, Registration through Gmail, Registration through LinkedIn
2. User Confirmation via Email Confirmation via OTP
3. Pro Version Accessing unlimited features
4. Non-Pro Version Limited access to selected features
5. Direct Link to Bank Account Accessing your bank account and carrying out the financial transactions
6. Customer Support Customer can access the support via help desk.

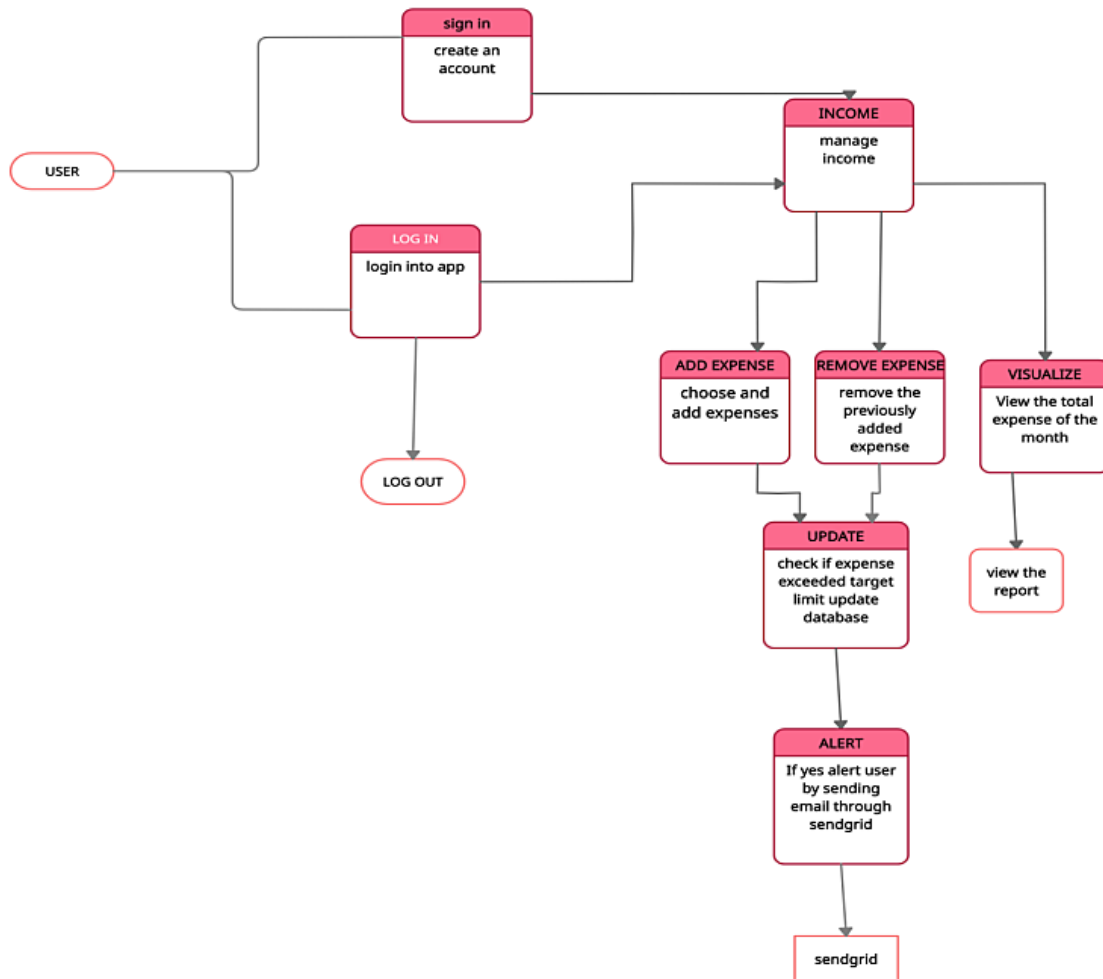
4.2 NON-FUNCTIONAL REQUIREMENTS

1. Usability By using this application, the user gets to know how much he/she is spending in his/her day-to-day life and this application will be very useful.
2. Security OTP based or finger print based login Requirement to secure his or her account.
3. Reliability Simple user interface, that means the application is very simple for people and that's the great advantage for the people for easy understanding.
4. Performance Prevent wrong expense claims thus reducing wastage.
5. Availability When it comes to availability there are two versions, they are pro version and Non pro version, where the Pro version is reimbursement whereas this uses unlimited access and Non Pro version uses the limited access.
6. Scalability Outstanding customer Experience Control the security

CHAPTER 5

PROJECT DESIGN

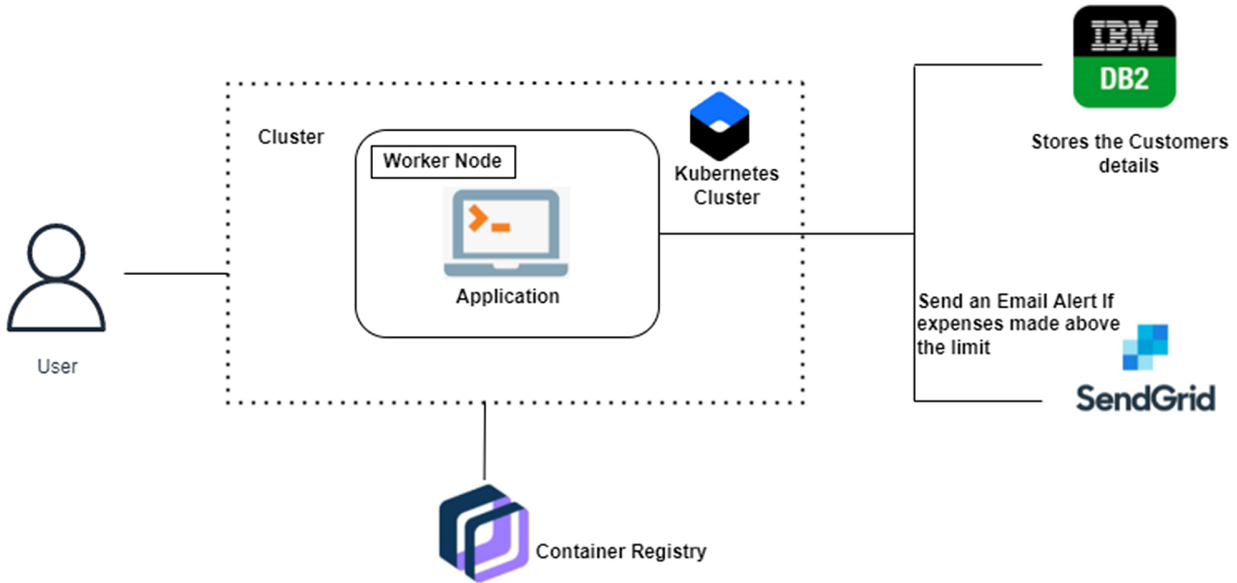
5.1 Data Flow Diagrams



USER TYPES	FUNCTIONAL REQUIREMENT	USER STORY NUMBER	USER STORY/TASK
Customer	registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.
	login	USN-2	As a user, I can log into the application by entering email & password
	add	USN-3	As a user, I can add new expenses
	remove	USN-4	As a user, I can remove previously added expenses
	view	USN-5	As a user , I can view my expenses and get insight
	Get alert message	USN-6	As the user if you exceed the budget the user should get the alert message
Administrator	Add/remove user	USN-7	As an admin , I can add or delete user details on db2 manually

5.2 Solution & Technical Architecture

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



5.2 User stories

Functional Requirement (Epic)	User Story Number	User Story / Task	Priority	Release
Registration	USN-1	As a user, I can register for the application by entering my credentials	High	Sprint 1

	USN-2	As a user, I will receive confirmation email once I have registered for the application	High	Sprint 1
Login	USN-3	As a user, I can log into the application by entering email & password	High	Sprint 1
Dashboard	USN-4	Logging in takes to the dashboard for the logged user.	High	Sprint 1

Workspace	USN-1	Workspace for personal expense tracking	High	Sprint 2
Charts	USN-2	Creating various graphs and statistics of customer's data	Medium	Sprint 2
Connecting to IBM DB2	USN-3	Linking database with dashboard	High	Sprint 2
	USN-4	Making dashboard interactive with JS	High	Sprint 2
	USN-1	Wrapping up the server side works of frontend	Medium	Sprint 3

Watson Assistant	USN-2	Creating Chatbot for expense tracking and for clarifying user's query	Medium	Sprint 3
SendGrid	USN-3	Using SendGrid to send mail to the user about their expenses	Low	Sprint 3
	USN-4	Integrating both frontend and backend	High	Sprint 3

Docker	USN-1	Creating image of website using docker/	High	Sprint 4
Cloud Registry	USN-2	Uploading docker image to IBM Cloud registry	High	Sprint 4
Kubernetes	USN-3	Create container using the docker image and hosting the site	High	Sprint 4
Exposing	USN-4	Exposing IP/Ports for the site	High	Sprint 4

CHAPTER 6

PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement(Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint 1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Keerthana P

		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Nanthinee N
	Login	USN-3	As a user, I can log into the application by entering email & password	1	High	Sri Ishwarya R
	Dashboard	USN-4	Logging in takes to the dashboard for the logged user.	2	High	Hari Priya P

Sprint 2	Workspace	USN-1	Workspace for personal expense tracking	2	High	Sri Ishwarya R
-----------------	-----------	-------	---	---	------	----------------

	Charts	USN-2	Creating various graphs and statistics of customer's data	1	Medium	Hari Priya P
	Connect to IBMDB2	USN-3	Linking database with dashboard	2	High	Nanthinee N
		USN-4	Making dashboard interactive with JS	2	High	Keerthana P

Sprint3		USN-1	Wrapping up the serverside works of frontend	1	Medium	Hari Priya P
	Watson Assistant	USN-2	Creating Chatbot forexpense tracking and for clarifying user's query	1	Medium	Keerthana P
	SendGrid	USN-3	Using SendGrid to send mail tothe user about their expens	1	Low	Nanthinee N

			es			
		USN-4	Integrating both frontend and backend	2	High	Sri Ishwarya R

Bug fixes,routine checks and improvisation by everyone in the team *Intended bugsonly						
Sprint4	Docker	USN-1	Creating image of website using docker/	2	High	Nanthinee N
	Cloud Registry	USN-2	Uploading dockerimage to IBMCloud registry	2	High	Hari Priya P

	Kubernetes	USN-3	create container using the docker image and hosting the site	2	High	Sri Ishwarya R
	Exposing	USN-4	Exposing IP/Ports for the site	2	High	Keerthana P

6.2 Sprint Delivery Schedule

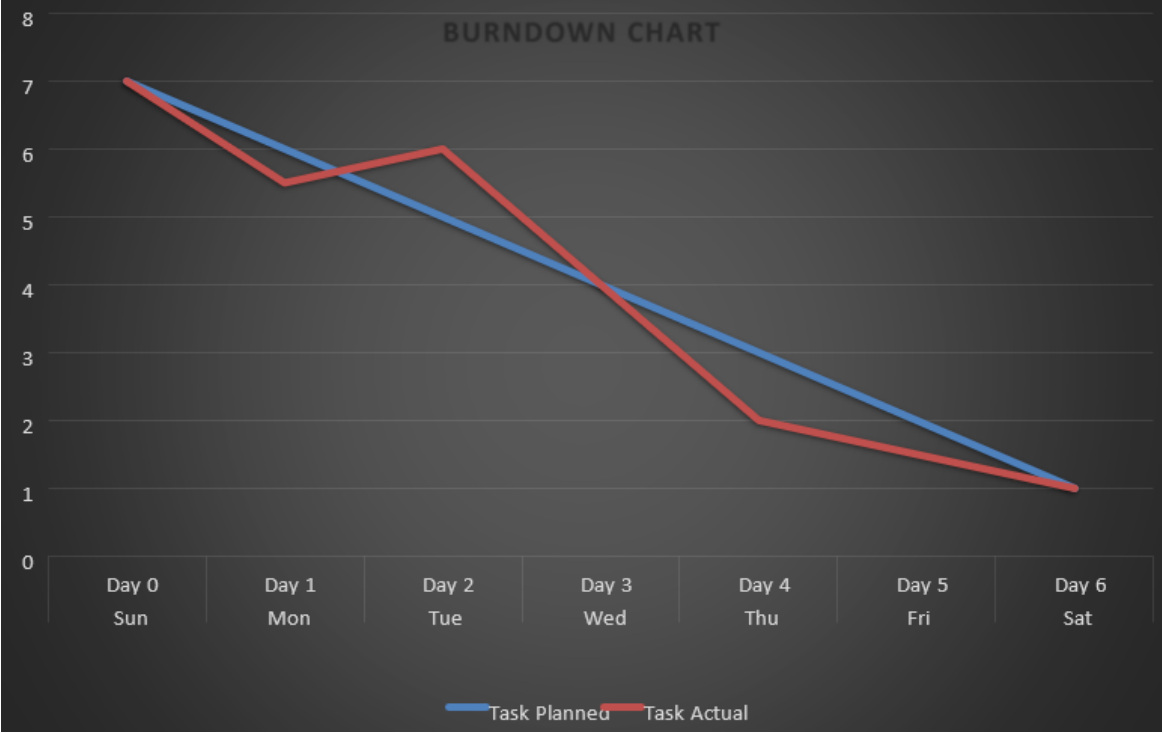
Project Tracker, Velocity & Burndown Chart:

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	24Oct 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:

Imagine we have a 10- days sprint duration and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

	Initial Estimate	24-Oct	25-Oct	26-Oct	27-Oct	28-Oct	29- Oct
Sprint number	Day 0	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6
Sprint-1	20	0	10	5	3	1	1
Sprint-2	20	2	10	4	1	1	2
Sprint-3	20	5	5	5	5	0	0
Sprint-4	20	3	3	3	3	3	5
Task planned	7	6	5	4	3	2	1
Task Actual	7	5.5	6	4	2	1.5	1



CHAPTER 7

CODING & SOLUTIONING

App.js

```
import React from 'react';
import { BrowserRouter, Routes, Route, } from "react-router-dom";import Home from
'./routes/home/Home';
import Login from './routes/forms/login/Login'; import Register from
'./routes/forms/register/Register';const App = () => {
return (
<BrowserRouter>
<Routes>
<Route path="/" element={<Home />} />
<Route exact path="/login" element={<Login />} />
<Route exact path="/register" element={<Register />} />
</Routes>
</BrowserRouter>
);
};
export default App;
```

App.css

```
.App {
text-align: center;}
```



```
App-logo {  
height: 40vmin;  
pointer-events: none;  
}  
  
@media (prefers-reduced-motion: no-preference) {  
  .App-logo {  
    animation: App-logo-spin infinite 20s linear;  
  }  
}
```

index.js

```
import React from 'react';  
import ReactDOM from 'react-dom';  
import App from './App';  
import './index.css';  
ReactDOM.render(  
  <React.StrictMode>  
    <App />  
  </React.StrictMode>,  
  document.getElementById('root')  
)
```

.routes/forms/login/login.css

```
.login {  
display: flex;  
align-items: center;
```

```
justify-content: center;
height: 100vh;}
.loginForm {
background-color: rgb(244 228 140);
padding: 0px 60px;
border-radius: 10px;}
.loginTitle {
color: rgb(139 120 2);
text-align: center;
margin-top: 30px;
}
```

.routes/forms/register/Register.js

```
import React, { useState } from 'react';
import { Link, useNavigate } from 'react-router-dom'; import FormInput from
'../formInput/FormInput'; import './register.css';
const Register = () => {
const navigate = useNavigate();
const [values, setValues] = useState({ username: "",
email: "",
password: "", confirmPassword: "",
});
const { username } = values;
const inputs = [
{
```

```
id: 1,
name: "username", type: "text",
placeholder: "Username", label: "Username",
errorMessage: "Username should be 3-16 characters and shouldn't include any special
characters",
required: true,
pattern: "^[A-Za-z0-9]{3,16}$",
},
{
id: 2,
name: "email",
type: "email", placeholder: "Email", label: "Email",
errorMessage: "It should be a valid email address", required: true,
},
{
id: 3,
name: "password", type: "password",
placeholder: "Password",
label: "Password",
errorMessage: "Password should be 8-20 characters and include at least 1 letter, 1
number and 1 special character",
required: true,
pattern: "^(?=.*[0-9])(?=.*[A-Za-z])(?=.*[!@#$%^&*])[a-zA-Z0-9!@#$%^&*]{8,20}$",
},
```

```

{
  id: 4,
  name: "confirmPassword", type: "password",
  placeholder: "Confirm Password", label: "Confirm Password", errorMessage:
  "Passwords don't match", required: true,
  pattern: values.password,
},
]

const handleSubmit = (e) => { e.preventDefault(); navigate(`/dashboard/${username}`);
};

const onChange = (e) => {
  setValues({ ...values, [e.target.name]: e.target.value });
};

return (
  <div className="register">
    <form className="registerForm" onSubmit={handleSubmit}>
      <h1 className="registerTitle">Register</h1>
      {inputs.map((input) => (
        <FormInput key={input.id}
          {...input}
          value={values[input.name]} onChange={onChange}
        />
      ))}
      <button className="registerBtn">Submit</button>

```

```
<div>

<p style={{ fontSize: "14px", marginBottom: "30px" }}>Already have an account?
<Link to="/login" className="loginLink">Click to Login</Link></p>

</div>

</form>

</div>

);

};

export default Register;

.routes/forms/register/register.css

.register { display: flex;
align-items: center; justify-content: center; height: 100vh;
}

.registerForm {
background-color: rgb(244 228 140); padding: 0px 60px;
border-radius: 10px;
}

.registerTitle {
color: rgb(139 120 2); text-align: center; margin-top: 30px;
}

.registerBtn { width: 100%; height: 50px; padding: 10px; border: none;
background-color: rgb(139 120 2); color: white;
border-radius: 10px; font-weight: bold; font-size: 18px; cursor: pointer; margin-top:
15px; margin-bottom: 15px;
```

```

}

.registerBtn:hover {
background-color: rgb(246 235 172);
color: rgb(139 120 2);
border: 2px solid rgb(139 120 2);
}

.loginLink {
text-decoration: none; margin-left: 5px; color: rgb(72 62 1);
font-weight: 500;
}

```

Connect to IBM DB2

```

conn = ibm_db.connect("DATABASE=bludb;" "HOSTNAME=6667d8e9-9d4d-4ccb-
ba32-
21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;"
"PORT=30376;" "SECURITY=SSL;"
"SSLServerCertificate=C:/Users/bhaua/OneDrive/Desktop/SEM Stuff
KCT/IBM/DigiCertGlobalRootCA.crt;"
"UID=bcv24102;"
"PWD=ilZgx2Zokf2nJOCW;" "",
"",)

```

Navbar.jsx

```

import React from 'react';
import { useNavigate } from 'react-router';
import './navbar.css';

```

```
const Navbar = () => {  
  const navigate = useNavigate();  
  const handleLogout = () => {  
    navigate('/');  
  }  
  return (  
    <div className="navbar">  
      <div className="navTitle">Personal Expense Tracker  
      <span className='navSubTitle'>speak to track</span>  
    </div>  
    <div>  
      <button className='navBtn' onClick={() => handleLogout()}>Logout</button>  
    </div>  
  </div>  
);  
}  
  
export default Navbar;
```

navbar.css

```
.navbar {  
  display: flex;  
  flex-direction: row;  
  height: 70px;  
  background-color: white;  
  border: 3px solid darkgoldenrod;
```

```
border-radius: 5px;
margin-top: 10px;
align-items: center;
justify-content: space-between;
}

.navTitle {
font-size: 30px;
margin-left: 15%;
text-align: center;
color: rgb(187, 134, 0);
font-weight: bold;
letter-spacing: 1px;
}

.navSubTitle {
letter-spacing: normal;
font-size: 20px;
color: grey;
margin-left: 75px;
}

.navBtn {
height: 70%;
margin-right: 220px;
background-color: rgb(228, 227, 194);
border-radius: 5px;
```



```
border: 2px solid rgb(111, 95, 0);
font-weight: 600;
color: rgb(194, 128, 6);
padding: 10px 20px;
cursor: pointer;
}
```

Details.jsx

```
import React from 'react';
import { Card, CardContent } from '@mui/material';
import { Doughnut, Pie, PolarArea, Radar } from 'react-chartjs-2';
import './details.css';
import useTransactions from '../useTransactions';
import 'chart.js/auto';

const Details = ({ title }) => {
  /*const [doughnatC, setDoughnatC] = useState(true);
  const [polarAreaC, setPolarAreaC] = useState(false);
  const [pieC, setPieC] = useState(false);
  const [radarC, setRadarC] = useState(false);*/
  const { chartData } = useTransactions(title);
  console.log(chartData);
  return (
    <div>
      <div style={{ display: 'flex', flexDirection: 'row', gap: '5px', marginTop: '10px' }}>
        <Card style={{}} className={title === 'Income' ? "income" : "expense"} >
```

```

{ /*<div className="chartButtonContainer" >
<button className={title === 'Income' ? (doughnatC ? "selectedIn" : "btnIn") :
(doughnatC ? "selectedEx" : "btnEx")} >
onClick={() => {
setDoughnatC(true);
setPolarAreaC(false);
setPieC(false);
setRadarC(false);
}}

```

details.css

```

.income {
border-top: 10px solid rgba(0, 255, 0, 0.7);
border-bottom: 10px solid rgba(0, 255, 0, 0.7);
}

.expense {
border-top: 10px solid rgba(255, 0, 0, 0.7);
border-bottom: 10px solid rgba(255, 0, 0, 0.7);
}

```

useTransactions.js

```

import { useContext } from "react"
import { expenseCategories, incomeCategories, resetCategories } from
"./constants/categories";
import { ExpenseTrackerContext } from "../context/context"
const useTransactions = (title) => {

```

```
resetCategories();

const { transactions } = useContext(ExpenseTrackerContext);

const transactionsPerType = transactions.filter((t) => t.type === title);

const total = transactionsPerType.reduce((acc, currVal) => acc += currVal.amount, 0);

const categories = title === 'Income' ? incomeCategories : expenseCategories;

var month = "";

var monthIncomeTotal = [
  {m:"January", amount:0},
  {m:"February", amount:0},
  {m:"March", amount:0},
  {m:"April", amount:0},
  {m:"May", amount:0},
  {m:"June", amount:0},
  {m:"July", amount:0},
  {m:"August", amount:0},
  {m:"September", amount:0},
  {m:"October", amount:0},
  {m:"November", amount:0},
  {m:"December", amount:0},,];

if (t.type === 'Income') {
  switch(month) {
    case '01': monthIncomeTotal[0].amount += t.amount;break;
    case '02': monthIncomeTotal[1].amount += t.amount;break;
    case '03': monthIncomeTotal[2].amount += t.amount;break;
```

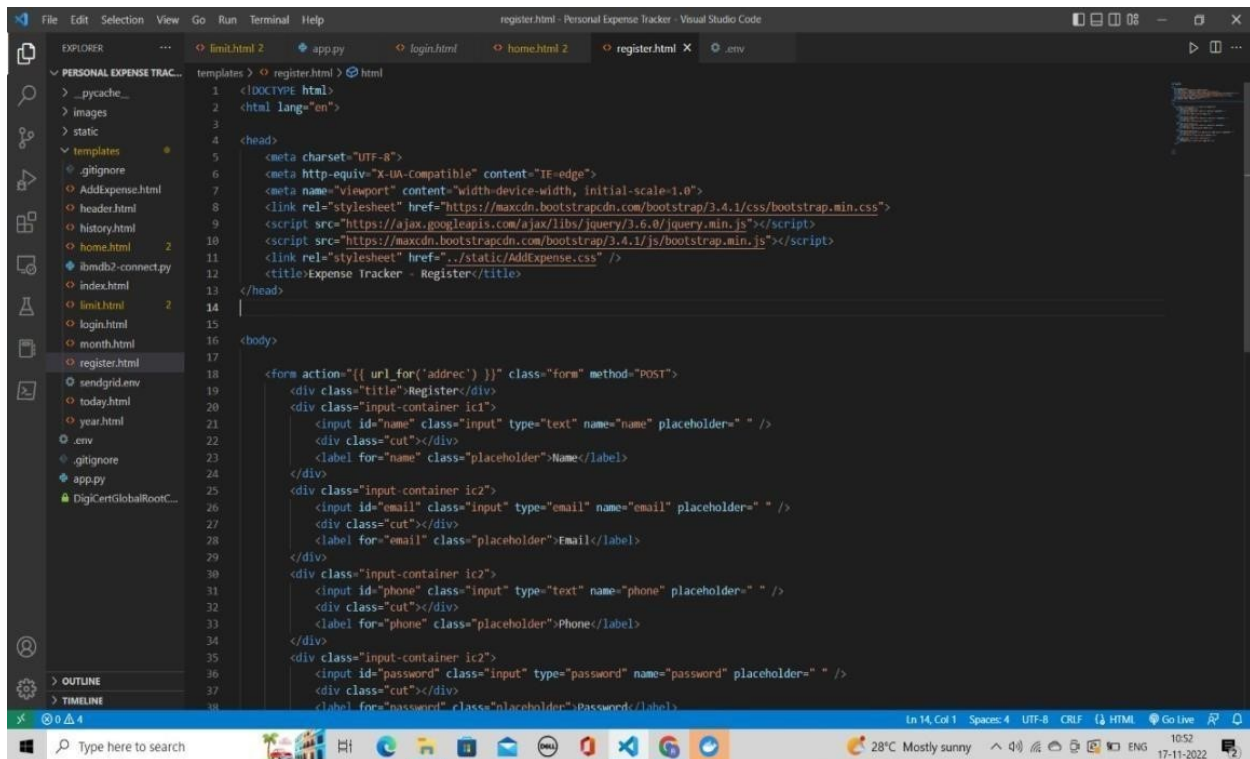
```
case '04': monthIncomeTotal[3].amount += t.amount;break;
case '05': monthIncomeTotal[4].amount += t.amount;break;
case '06': monthIncomeTotal[5].amount += t.amount;break;
case '07': monthIncomeTotal[6].amount += t.amount;break;
case '08': monthIncomeTotal[7].amount += t.amount;break;
case '09': monthIncomeTotal[8].amount += t.amount;break;
case '10': monthIncomeTotal[9].amount += t.amount;break;
case '11': monthIncomeTotal[10].amount += t.amount;break;
case '12': monthIncomeTotal[11].amount += t.amount;break;
default: break;}
}
```

detailsTrack.css

```
.incomeChart {
background-color: white;
border: 5px solid rgb(147, 147, 147);
margin: 20px 200px;
padding: 50px;
border-radius: 10px;}

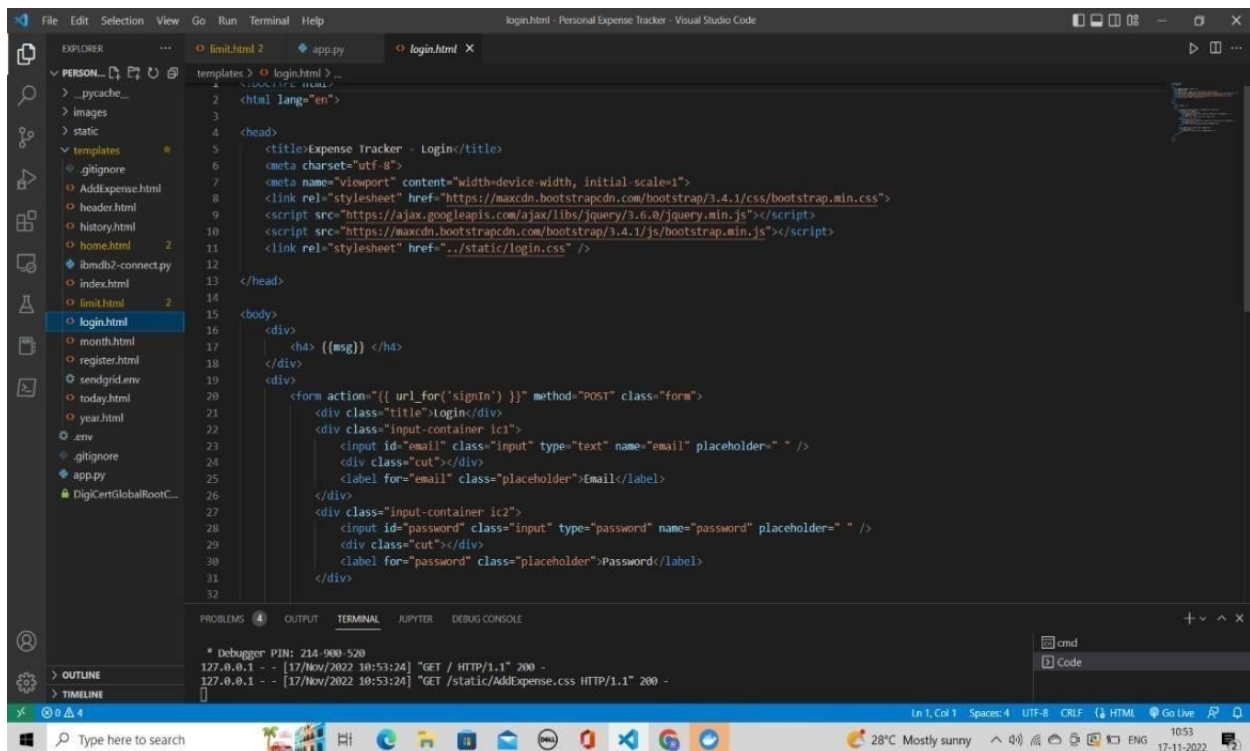
.expenseChart {
background-color: white;
border: 5px solid rgb(147, 147, 147);
margin: 20px 200px;
padding: 50px;
border-radius: 10px;}
```

7.1 SOLUTIONING



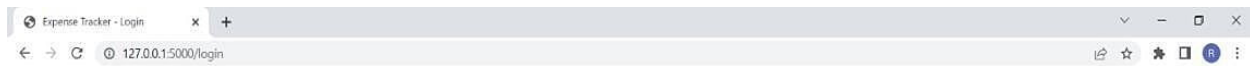
The screenshot shows the Visual Studio Code editor with the 'register.html' file open. The Explorer sidebar on the left shows the project structure, including files like 'register.html', 'login.html', 'home.html', and 'app.py'. The main editor area displays the HTML code for 'register.html', which includes a form for user registration. The code is as follows:

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
9   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
10  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
11  <link rel="stylesheet" href="static/AddExpense.css" />
12  <title>Expense Tracker - Register</title>
13 </head>
14
15 <body>
16
17   <form action="{{ url_for('addrec') }}" class="form" method="POST">
18     <div class="title">Register</div>
19     <div class="input-container ic1">
20       <input id="name" class="input" type="text" name="name" placeholder=" " />
21       <div class="cut"></div>
22       <label for="name" class="placeholder">Name</label>
23     </div>
24     <div class="input-container ic2">
25       <input id="email" class="input" type="email" name="email" placeholder=" " />
26       <div class="cut"></div>
27       <label for="email" class="placeholder">Email</label>
28     </div>
29     <div class="input-container ic2">
30       <input id="phone" class="input" type="text" name="phone" placeholder=" " />
31       <div class="cut"></div>
32       <label for="phone" class="placeholder">Phone</label>
33     </div>
34     <div class="input-container ic2">
35       <input id="password" class="input" type="password" name="password" placeholder=" " />
36       <div class="cut"></div>
37       <label for="password" class="placeholder">Password</label>
38     </div>
39   </form>
40 </body>
```



The screenshot shows the Visual Studio Code editor with the 'login.html' file open. The Explorer sidebar on the left shows the project structure, including files like 'login.html', 'register.html', 'home.html', and 'app.py'. The main editor area displays the HTML code for 'login.html', which includes a form for user login. The code is as follows:

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <title>Expense Tracker - Login</title>
6   <meta charset="utf-8">
7   <meta name="viewport" content="width=device-width, initial-scale=1">
8   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
9   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
10  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
11  <link rel="stylesheet" href="static/login.css" />
12 </head>
13
14 <body>
15
16   <div>
17     <h4>{{msg}}</h4>
18   </div>
19   <div>
20     <form action="{{ url_for('signIn') }}" method="POST" class="form">
21       <div class="title">Login</div>
22       <div class="input-container ic1">
23         <input id="email" class="input" type="text" name="email" placeholder=" " />
24         <div class="cut"></div>
25         <label for="email" class="placeholder">Email</label>
26       </div>
27       <div class="input-container ic2">
28         <input id="password" class="input" type="password" name="password" placeholder=" " />
29         <div class="cut"></div>
30         <label for="password" class="placeholder">Password</label>
31       </div>
32     </form>
33   </div>
34 </body>
```



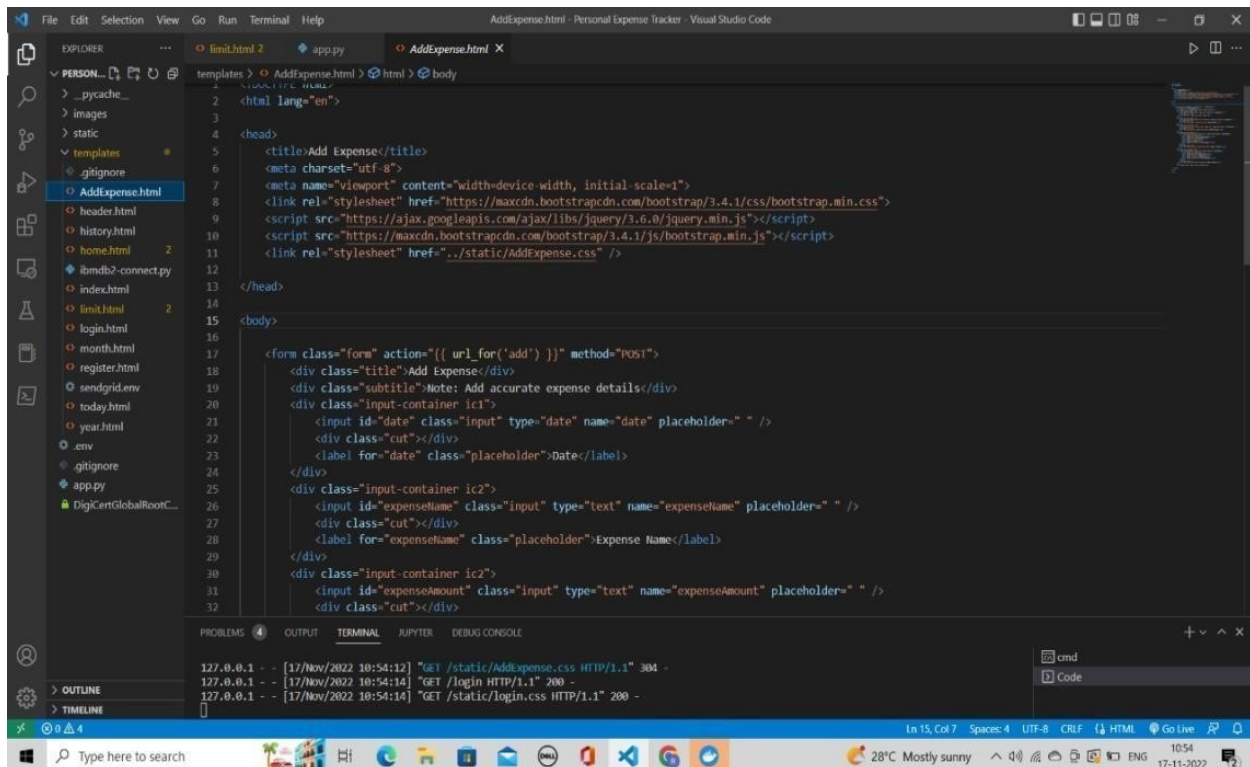
Login

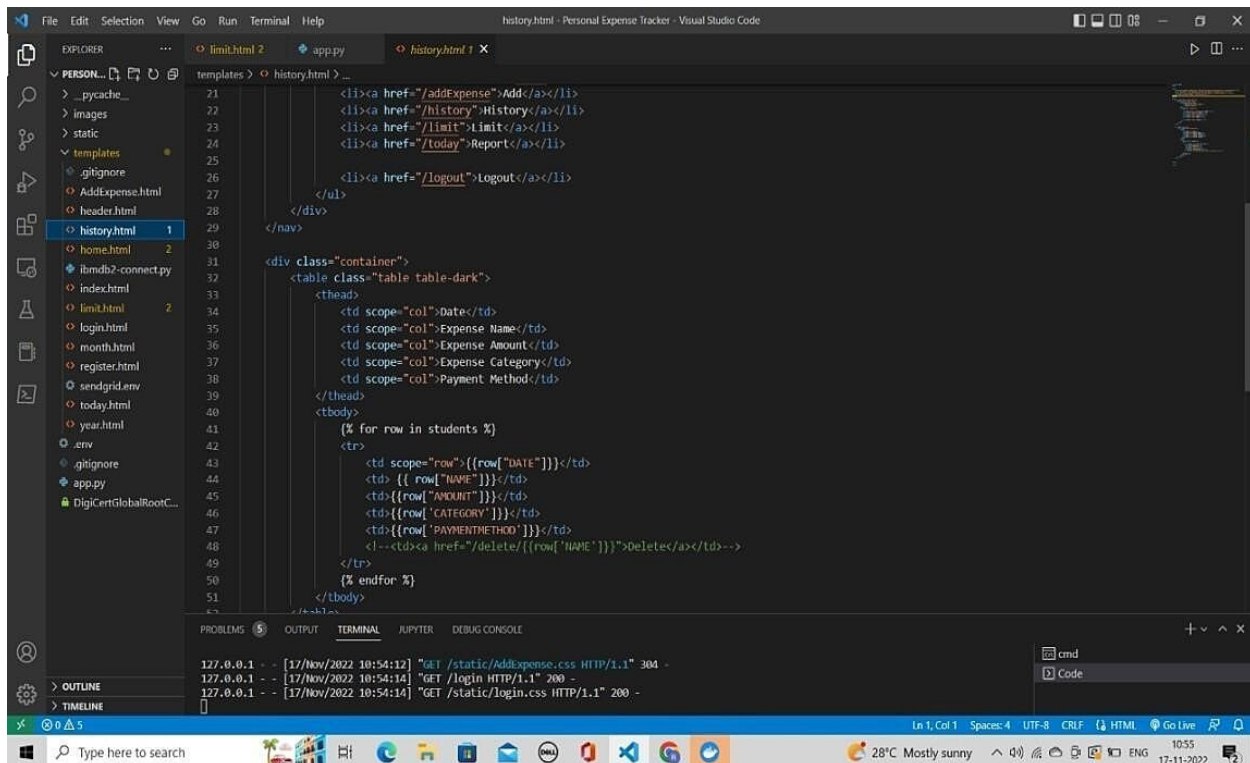
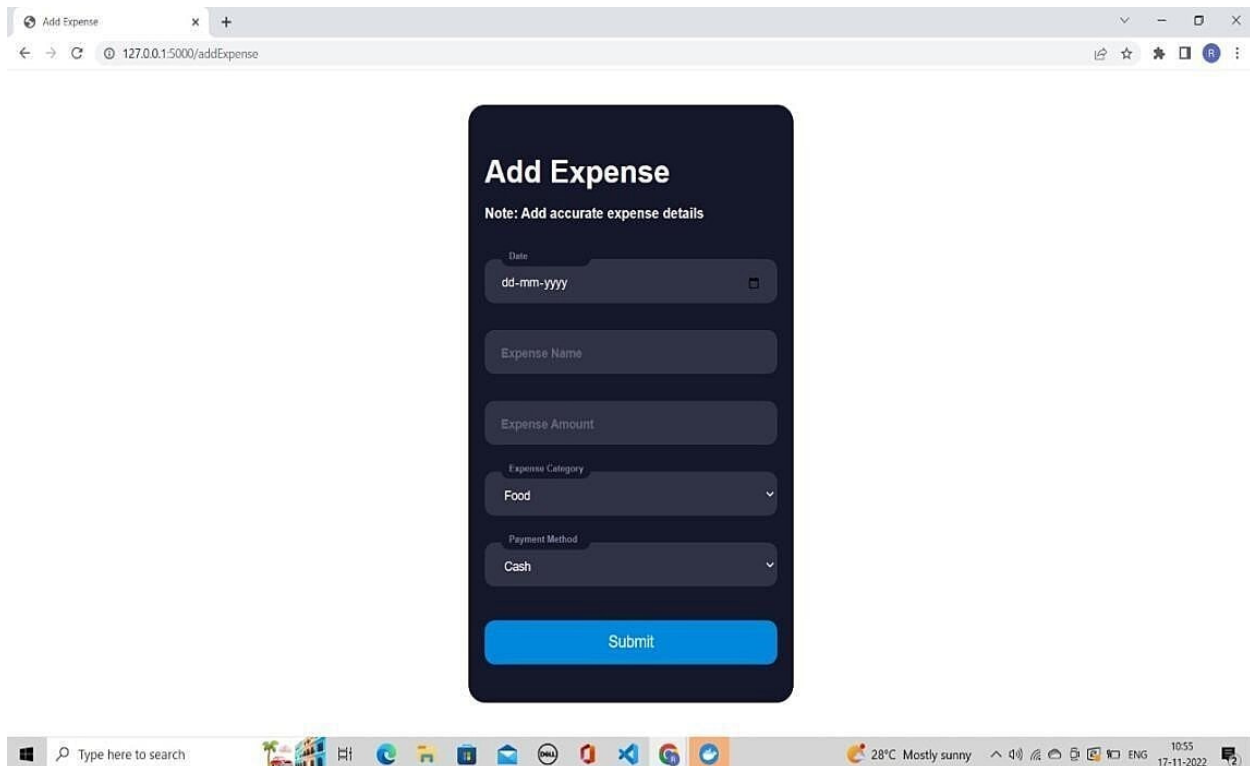
Email

Password

Login

New User? [Register Here](#)





ENTER the MONTHLY LIMIT to avoid over EXPENSES



CHAPTER 8

TESTING

8.1 Test Cases

Test cases are a set of actions performed on a system to determine if it satisfies software requirements and functions correctly as it is claimed to perform.

8.2 User Acceptance Testing

Before deploying the software application to a production environment the end user or client performs a type of testing known as user acceptance testing, or UAT to ensure whether the software functionalities serve the purpose of development.

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [PERSONAL EXPENSE TRACKER APPLICATION] project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	10	0	0	10
Client Application	50	0	0	50
Security	1	0	0	1

Outsource Shipping	3	0	0	3
--------------------	---	---	---	---

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	3	1	2	16
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	13	12	25	74

3. Test Case Analysis

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

Exception Reporting	8	0	0	8
Final Report Output	4	0	0	4
Version Control	2	0	0	2

CHAPTER 9

RESULTS

9.1 Performance Metrics

9.1.1 Hours Worked :500 hours

9.1.2 Stick to Timelines :80%

9.1.3 Consistency of the Product :95%

9.1.4 Efficiency of the Product :98%

9.1.5 Quality of the Product :99.9%

CHAPTER 10

ADVANTAGES & DISADVANTAGES

Advantages:

1. You have no control over your money. If you don't check your spending and create a budget, you will have no control whatsoever on your money. Instead, money will control you, and you will either have perpetual lack of funds or you will end up steeped in debt. A money manager app helps you decide between short-term and long-term spending .
2. You have no financial goals If you are spending money frivolously, you will not have money to set financial goals. However, when you have a daily expense manager, you will be able to work with limited resources and use your money in a wise manner so that you can create financial goals and ensure you meet them.
3. You are unaware what is happening with your money. If you are clueless about how much is your inflow and how much you are spending, you will not know at the end of the month what happened to your money. An expense tracker helps you figure out what is happening to your money, and whether you can afford something you want.
4. You spend and save in a haphazard manner If you don't have great financial management skills, you will not know how to categorize your expenses.

However, tracking your expenses and

Budgeting them will help you become aware of how much you have to allocate to each expense category, and if you are short, you will be able to make adjustments with ease.

5. You have no clue about making your money work for you. In this day and age, when expenses are going through the roof, it has become crucial that you learn to make your money work for you so that you can create a nest egg for the future.
 6. You don't have funds for emergencies Remember, emergencies come when you least expect. Hence, if you don't have money stashed away for a rainy day, you will end up borrowing from family and friends. This way you could get into debt that will be difficult to pay back due to your poor money management skills
- Disadvantages

Disadvantages:

1. Your information is less secure, and probably being used and sold. If the service is free, then the product is you. Mint.com, like other financial apps, is a free service. They have to pay their bills somehow, so regardless of what their privacy policy may or may not say, just assume that your spending history and trends are going to be recorded and analyzed, by someone, somewhere.
2. Now, you shouldn't have to worry about credit card fraud or identity theft, these companies are large enough and secure enough that you'll never have to worry about something like that. Just recognize that your information, most likely anonymous, will be used and potentially even sold.
3. Personally, I have no problem with that, but if you do, then make sure you avoid these types of services. Automating everything to do with your finances can

make you financially lazy. If your bills are paid automatically and your finances are tracked automatically, then what is there left for you to do? Not a lot, to be honest. So you might stop caring about what you're spending and where your money is going. Eventually you may look at your Mint data and realize that you've blown your budget over the last two months, but by then it is too late. So if you do choose to use this program, ensure that you are also being diligent in checking in on your finances. Set up a weekly or biweekly check for yourself to go through your finances and hit on all the important points.

CHAPTER 11

CONCLUSION

After making this application we assure that this application will help its users to manage the cost of their daily expenditure. It will guide them and 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 wishes to manage money and to preserve the record of their daily cost which may be useful to change their way of spending money. In short, this application will help its users to overcome the wastage of money.

CHAPTER 12

FUTURE SCOPE

The further enhancements that are planned for developing the application to be user friendly and further features like login with gmail or social media accounts using facebook or twitter and also using SSO login so that it is helpful for users to get into the platform easily.

CHAPTER 13

APPENDIX

GitHub Link: <https://github.com/IBM-EPBL>

Project Demo Link: <https://github.com/IBM-EPBL/IBM-Project-5332-1658757938/blob/main/Final%20Deliverables/Project%20Demo%20Video.mp4>