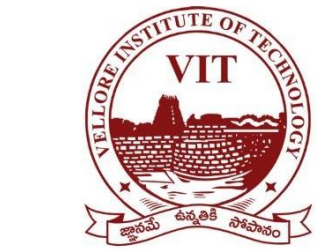


VIT-AP UNIVERSITY

Wise Rupee (Expense Tracker)

SWE4004 - Cloud Computing and its Applications

Members Involved:

APARNA SUPRIYA SURESH 20MIS7062

Under the guidance of:

DR. Sudha S V

Abstract:

This project is based on an expense and income tracking system. This project aims to create an easy, faster and smooth tracking system between the expense and the income. This project also offers some opportunities that will help the user to sustain all financial activities like digital automated diary. So, for the better expense tracking system, we developed our project that will help the users a lot. Most of the people cannot track their expenses and income one way they face a money crisis, in this case daily expense tracker can help the people to track income-expense day to day and making life tension free. Money is the most valuable portion of our daily life and without money we will not last one day on the earth. So using the daily expense tracker application is important to load a happy family. Daily expense tracker helps the user to avoid unexpected expenses and bad financial situations. This Project will save time and provide a responsible lifestyle. This system is made and supervised by the experts and satisfying by the user. HTML, CSS, Material-ui, JavaScript and react are used to develop the project

Contents

1.INTRODUCTION

Personal finance management is an important part of people's lives. However, everyone does not have the knowledge or time to manage their finances in a proper manner. And, even if a person has time and knowledge, they do not bother with tracking their expenses as they find it tedious and time-consuming. Now, you don't have to worry about managing your expenses, as you can get access to an expense tracker that will help in the active management of your finances.

People tend to overspend without realizing, and this can prove to be disastrous. Using a daily expense manager can help you keep track of how much you spend every day and on what. At the end of the month, you will have a clear picture where your money is going. This is one of the best ways to get your expenses under control and bring some semblance of order to your finances.

Today, there are several expense manager applications in the market. Some are paid managers while others are free. Even banks like ICICI offer their customers expense tracker to help them out. Before you decide to go in for a money manager, it is important to decide the type you want.

If you are looking for something simple, you need to stay away from complex applications that have a steep learning curve. You will get frustrated and not use the tracker. On the other hand, if you are looking for a feature-laden application that handles all your expenses and finances seamlessly, going in for a simple app will be useless, as it will not have the features you are looking for.

Money managers can be divided into two categories. They are:

- Simple applications that are quick and allow you to manage and track your personal expenses
- Complex applications that allow you to manage multiple user accounts and can be integrated with your credit cards, debit cards and bank accounts. These apps are for individuals who have a lot of money outflow or businesses that want to keep track of their employees' expenses.

2. OBJECTIVES

1. Track your spends automatically
2. Categorise your expenses
3. Set spending limits
4. Prioritize Your Spending
5. Identify Fraud
6. Take Control of Your Finances
7. Saving and Investment

3. CLOUD SERVICES USED IN OUR PROJECT

User's side – Software as a service (SaaS)

Developer's side – Infrastructure as a service (IaaS)

Software as a service

Software as a service (SaaS) is a software distribution model in which a cloud provider hosts applications and makes them available to end users over the internet. In this model, an independent software vendor (ISV) may contract a third-party cloud provider to host the application. Or, with larger companies, such as Microsoft, the cloud provider might also be the software vendor.

In this project we used Visual Studio code to write our code and add functionality to out project

Visual Studio Online is a Software-as-a-Service (SaaS) offering. The primary reasons it is classified as SaaS rather than PaaS are:

- You consume it in a per-user-license model.
- Your data is managed on your behalf by the Visual Studio Online ops team.
- You do not need to configure any "server side" features or setup virtual machines or patch operating systems (granted these last two are more IaaS than PaaS).

Infrastructure as a service

- Infrastructure as a service (IaaS) is a type of cloud computing service that offers essential compute, storage, and networking resources on demand, on a pay-as-you-go basis. IaaS is one of the four types of cloud services, along with software as a service (SaaS), platform as a service (PaaS), and serverless.

- Migrating your organization's infrastructure to an IaaS solution helps you reduce maintenance of on-premises data centres, save money on hardware costs, and gain real-time business insights. IaaS solutions give you the flexibility to scale your IT resources up and down with demand. They also help

  you quickly provision new applications and increase the reliability of your underlying infrastructure.

4.IMPLEMENTING THE CLOUD TECHNOLOGIES

Firebase is that it is designed to build fully functional applications without any additional components. E.g. you could serve your static web app using Firebase Hosting, manage users with Firebase Auth, use Firestore for data storage (the syncing features of Firestore are truly amazing if you're not familiar with them), and tie things together with Cloud Functions. Also, note that while Firebase was originally pretty heavily focused on mobile, especially native mobile, most of their services work great on the web regardless.

The other thing about Firebase (referencing the previous comment about it being a little weird as a "Cloud in a Cloud"), is that it essentially exists on top of GCP. For example, a Firebase Project is really a GCP project under the covers, Cloud Functions for Firebase are really some syntactic sugar on top of GCP functions, etc. While that can be a little confusing at first, it can provide some great value. We have apps where most of our backend is in GCP (e.g. running Node in App Engine Flexible, using Cloud SQL Postgres for our DB, using Cloud PubSub topics for our eventing system, etc.) but then we use Firebase Auth for our user authentication, Firebase Hosting to serve our React app, etc.

5.CLOUD DEPLOYMENT

We use Hybrid Cloud as a deployment model as it includes applications, or their components such as compute, networking, and storage, when deployed across public and private clouds. On-premises servers are also often referred to as private clouds.

In this project we will be hosting the website through firebase. Firebase Hosting provides fast and secure hosting for your web app, static and dynamic content, and microservices.

Firebase Hosting is production-grade web content hosting for developers. With a single command, you can quickly deploy web apps and serve both static and dynamic content to a global CDN (content delivery network). You can also pair Firebase Hosting with Cloud Functions or Cloud Run to build and host microservices on Firebase. Firebase is fundamentally a collection of tools developers can rely on, creating applications and expanding them based on demand.

Firebase aims to solve three main problems for developers:

Build an app, fast

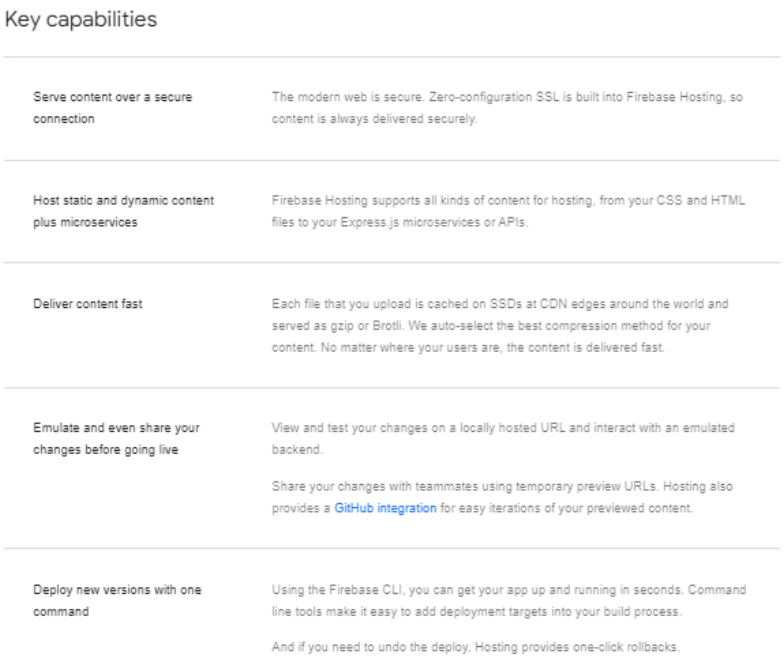
Release and monitor an app with confidence

Engage users,

Developers relying on this platform get access to services that they would have to develop themselves, and it enables them to lay focus on delivering robust application experiences.

Some of the Google Firebase platform's standout features include databases, authentication, push messages, analytics, file storage, and much more.

Since the services are cloud-hosted, developers can smoothly perform on-demand scaling without any hassle. Firebase is currently among the top app development platforms relied upon by developers across the globe.



6.Firebase Hosting

Firebase Hosting is built for the modern web developer. Websites and apps are more powerful than ever with the rise of front-end JavaScript frameworks like Angular and static generator tools like Jekyll. Whether you are deploying a simple app landing page or a complex Progressive Web App (PWA), Hosting gives you the infrastructure, features, and tooling tailored to deploying and managing websites and apps.

Using the Firebase CLI, you deploy files from local directories on your computer to our Hosting servers. Beyond serving static content, you can use Cloud Functions for Firebase or Cloud Run to serve dynamic content and host microservices on your sites. All content is served over an SSL connection from the closest edge server on our global CDN.

You can also view and test your changes before going live. Using the Firebase Local Emulator Suite, you can emulate your app and backend resources at a locally hosted URL. You can also share your changes at a temporary preview URL and set up a GitHub integration for easy iterations during development.

Firebase Hosting has lightweight hosting configuration options for you to build sophisticated PWAs. You can easily rewrite URLs for client-side routing, set up custom headers, and even serve localized content.

For serving your content, Firebase offers several domain and subdomain options:

By default, every Firebase project has subdomains at no cost on the web.app and firebaseapp.com domains. These two sites serve the same deployed content and configuration.

You can create multiple sites if you have related sites and apps that serve different content but still share the same Firebase project resources (for example if you have a blog, admin panel, and public app).

You can connect your own domain name to a Firebase-hosted site.

Firebase automatically provisions SSL certificates for all your domains so that all your content is served securely.

Implementation path

Install the Firebase CLI The Firebase CLI makes it easy to set up a new Hosting project, run a local development server, and deploy content.

Set up a project directory

Add your static assets to a local project directory, then run firebase init to connect the directory to a Firebase project.

In your local project directory, you can also set up Cloud Functions or Cloud Run for your dynamic content and microservices.

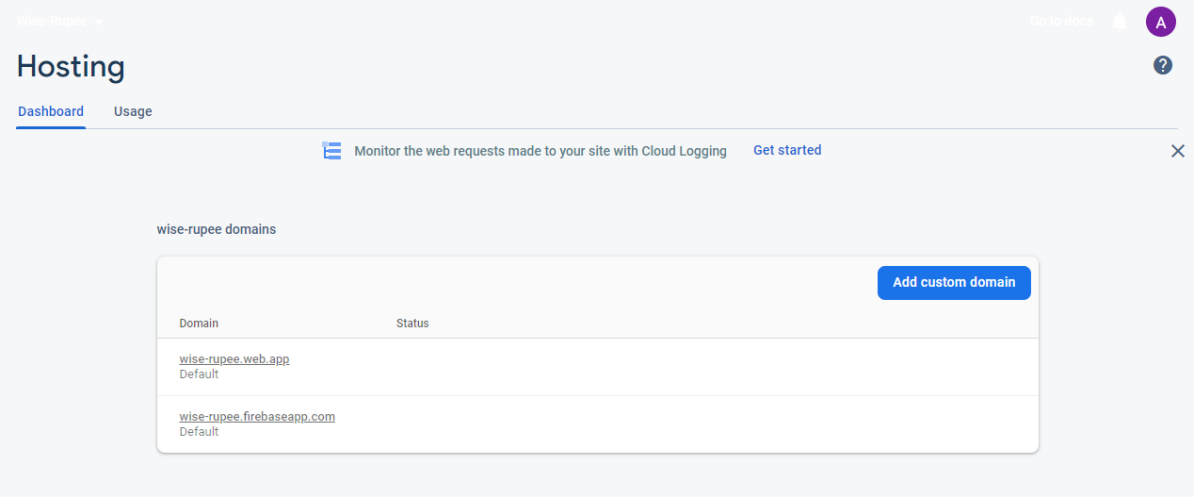
View, test, and share your changes before going live (optional)

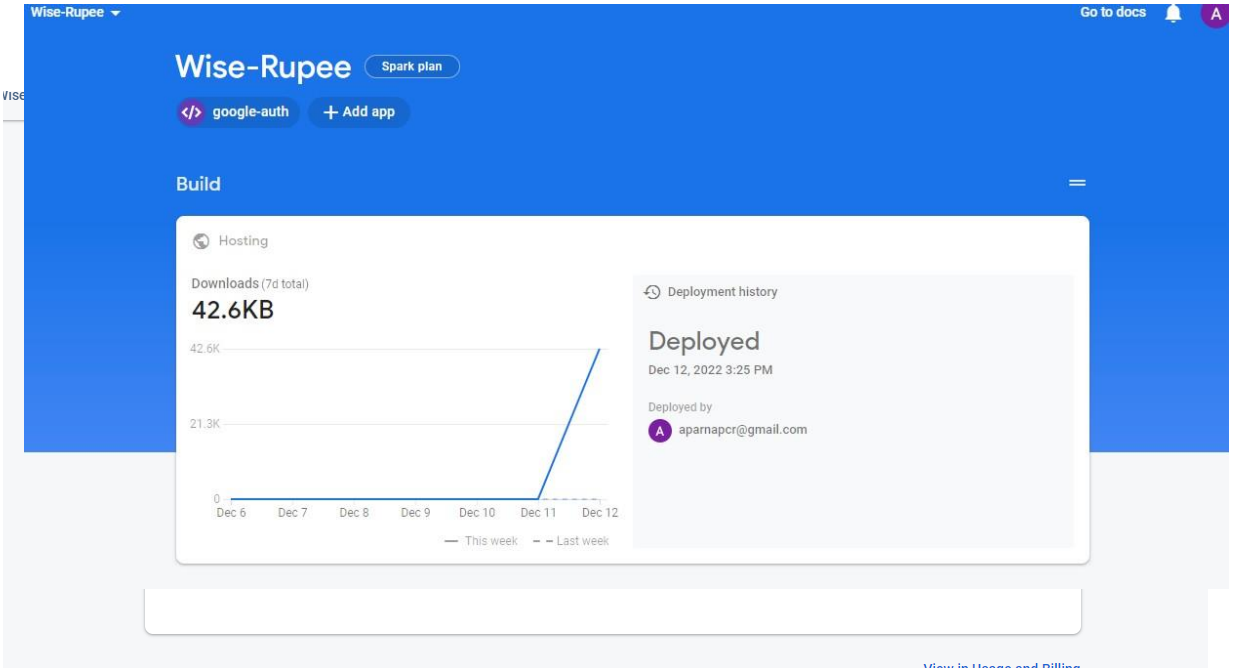
Run firebase emulators:start to emulate Hosting and your backend project resources at a locally hosted URL.

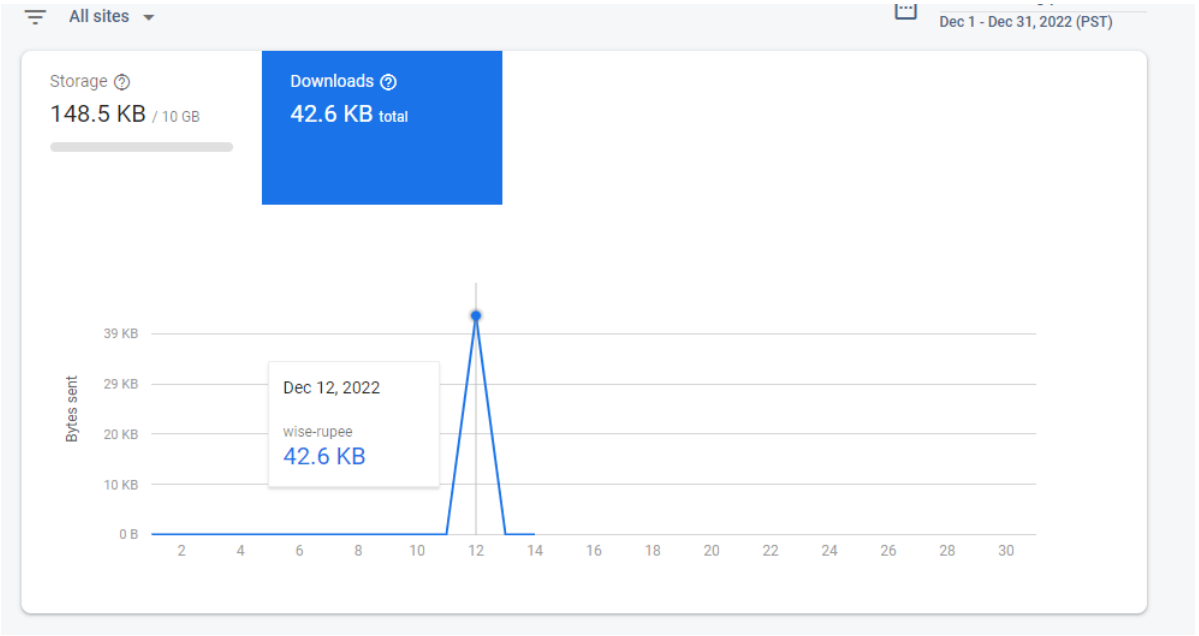
To view and share your changes at a temporary preview URL, run firebase hosting:channel:deploy to create and deploy to a preview channel. Set up the GitHub integration for easy iterations of your previewed content.

Deploy your site When things are looking good, run firebase deploy to upload the latest snapshot to our servers. If you need to undo the deploy, you can roll back with just one click in the Firebase console.

Link to a Firebase Web App (optional) By linking your site to a Firebase Web App, you can use Google Analytics to collect usage and behaviour data for your app and use Firebase Performance Monitoring to gain insight into the performance characteristics of your app.




Wise-Rupee
Go to docs
Wise-Rupee
Spark plan
google-auth
Add app
Build
Hosting
Downloads (7d total)
42.6KB
42.6K
21.3K
0
Dec 6
Dec 7
Dec 8
Dec 9
Dec 10
Dec 11
Dec 12
This week
Last week
Deployment history
Deployed
Dec 12, 2022 3:25 PM
Deployed by
aparnapcr@gmail.com


All sites
Dec 1 - Dec 31, 2022 (PST)
Storage
148.5 KB / 10 GB
Downloads
42.6 KB total
39 KB
29 KB
20 KB
10 KB
0 B
Bytes sent
Dec 12, 2022
wise-rupee
42.6 KB
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30

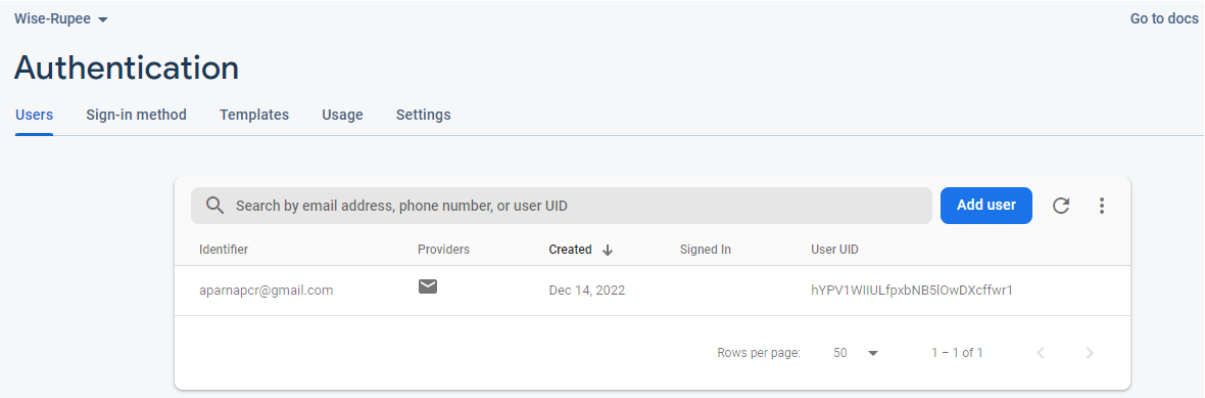
7.Services used

1. Firebase Authentication

Firebase Authentication provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app. It supports authentication using passwords, phone numbers, popular federated identity providers like Google, Facebook and Twitter, and more.

Firebase Authentication integrates tightly with other Firebase services, and it leverages industry standards like OAuth 2.0 and OpenID Connect, so it can be easily integrated with your custom backend.

When you upgrade to Firebase Authentication with Identity Platform, you unlock additional features, such as multi-factor authentication, blocking functions, user activity and audit logging, SAML and generic OpenID Connect support, multi-tenancy, and enterprise-level support.



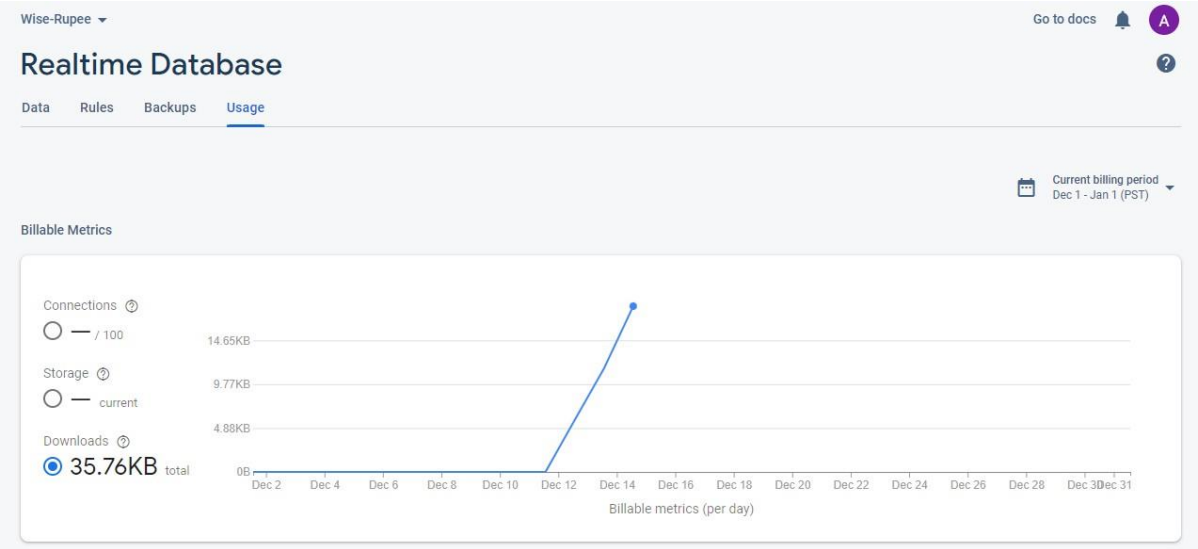
2. Firebase Realtime Database

The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in real time to every connected client. When you build cross-platform apps with our Apple platforms, Android, and JavaScript SDKs, all of your clients share one Realtime Database instance and automatically receive updates with the newest data.

Key capabilities

Realtime	Instead of typical HTTP requests, the Firebase Realtime Database uses data synchronization—every time data changes, any connected device receives that update within milliseconds. Provide collaborative and immersive experiences without thinking about networking code.
Offline	Firebase apps remain responsive even when offline because the Firebase Realtime Database SDK persists your data to disk. Once connectivity is reestablished, the client device receives any changes it missed, synchronizing it with the current server state.
Accessible from Client Devices	The Firebase Realtime Database can be accessed directly from a mobile device or web browser; there's no need for an application server. Security and data validation are available through the Firebase Realtime Database Security Rules, expression-based rules that are executed when data is read or written.
Scale across multiple databases	With Firebase Realtime Database on the Blaze pricing plan, you can support your app's data needs at scale by splitting your data across multiple database instances in the same Firebase project. Streamline authentication with Firebase Authentication on your project and authenticate users across your database instances. Control access to the data in each database with custom Firebase Realtime Database Rules for each database instance.

Realtime database:



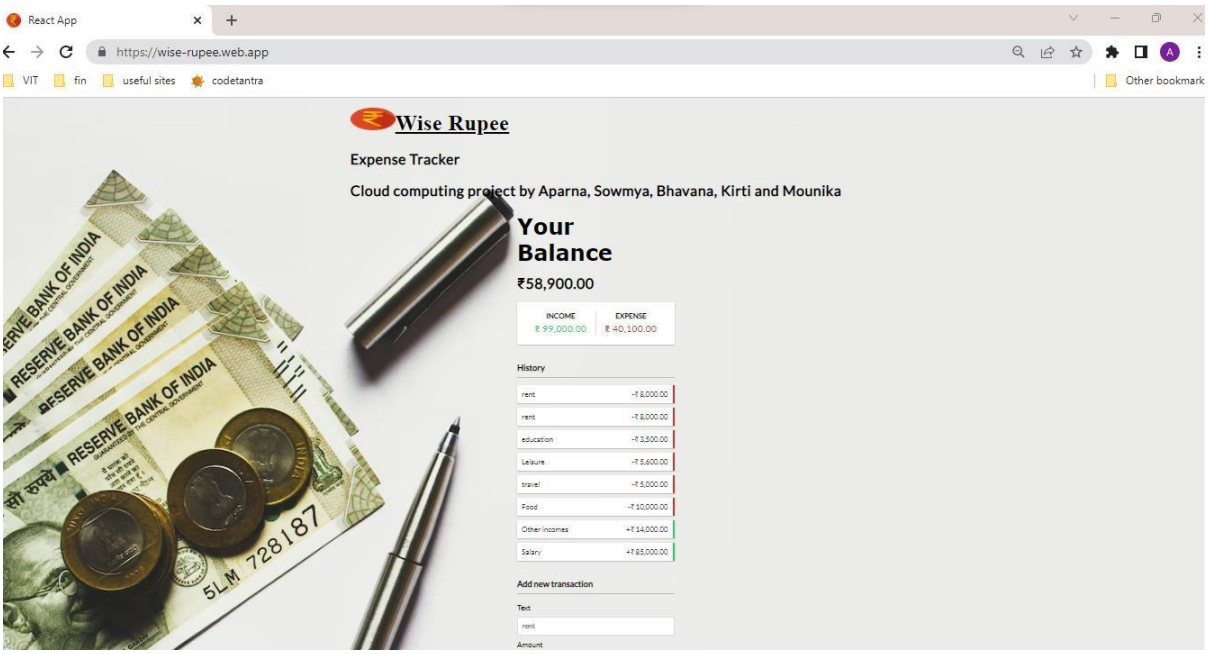
8.Project modules User Side/ Client-Side Cloud Services: SAAS For:

1. Website

Interactions

1. Add (+) main income

2. Use (+) operation to add/increase overall amount
3. Use (-) operation to subtract from the total income
4. Can add text to form categories for expenditures and gains.
5. Can edit/delete a particular attribute
6. On top balance will be highlighted after updating information.



9.IMPLEMENTATION

```
{
  "short_name": "Wiserupee",
  "name": "wiserupee expense tracker",
  "icons": [
    {
      "src": "favicon.ico",
      "sizes": "64x64 32x32 24x24 16x16",
```

```
    "type": "image/x-icon"
   },
   {
    "src": "logo192.png",
    "type": "image/png",
    "sizes": "192x192"
   },
   {
    "src": "logo512.png",
    "type": "image/png",
    "sizes": "512x512"
   }
  ],
  "start_url": ".",
  "display": "standalone",
  "theme_color": "#000000",
  "background_color": "#ffffff"
}
```

Javascript code

```
import React, { createContext, useReducer } from 'react';
import AppReducer from './AppReducer';
```

```
// Initial state
const initialState = {
  transactions: []
}

// Create context
export const GlobalContext = createContext(initialState);

// Provider component
export const GlobalProvider = ({ children }) => {
  const [state, dispatch] = useReducer(AppReducer,
initialState);

  // Actions
  function deleteTransaction(id) {
    dispatch({
      type: 'DELETE_TRANSACTION',
      payload: id
    });
  }

  function addTransaction(transaction) {
```

```
    dispatch({
      type: 'ADD_TRANSACTION',
      payload: transaction
    });
  }

  return (<GlobalContext.Provider value={{
    transactions: state.transactions,
    deleteTransaction,
    addTransaction
  }}>
    {children}
  </GlobalContext.Provider>);
}
```

ADDING TRANSCATION

```
import React, {useState, useContext} from 'react'
import { GlobalContext } from '../context/GlobalState';

export const AddTransaction = () => {
  const [text, setText] = useState('');
  const [amount, setAmount] = useState(0);
```

```
  const { addTransaction } = useContext(GlobalContext);

  const onSubmit = e => {
    e.preventDefault();

    const newTransaction = {
      id: Math.floor(Math.random() * 100000000),
      text,
      amount: +amount
    }

    addTransaction(newTransaction);
  }

  return (
    <>
      <h3>Add new transaction</h3>
      <form onSubmit={onSubmit}>
        <div className="form-control">
          <label htmlFor="text">Text</label>
          <input type="text" value={text} onChange={(e) =>
setText(e.target.value)} placeholder="Enter text..." />
        </div>
```

```
        <div className="form-control">
          <label htmlFor="amount"
            >Amount <br />
            (negative - expense, positive - income)</label
          >
          <input type="number" value={amount} onChange={(e)
=> setAmount(e.target.value)} placeholder="Enter amount..."
/>
        </div>
        <button className="btn">Add transaction</button>
      </form>
    </>
  )
}
```

BALANCE JAVASCRIPT CODE

```
import React, { useContext } from 'react';
import { GlobalContext } from '../context/GlobalState';

//Money formatter function
function moneyFormatter(num) {
let p = num.toFixed(2).split('.');
return (
```

```
    '₹' + (p[0].split('')[0]=== '-' ? '-' : '') +
    p[0]
      .split('')
      .reverse()
      .reduce(function (acc, num, i, orig) {
        return num === '-' ? acc : num + (i && !(i % 3) ? ',' : '') +
acc;
      }, '') +
    '.' +
    p[1]
  );
}

export const Balance = () => {
  const { transactions } = useContext(GlobalContext);

  const amounts = transactions.map(transaction =>
transaction.amount);

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

  return (
    <>
```

```
    <h2><font face = "Verdana " size = "10">Your
Balance</font><br /></h2>
    <h1>{moneyFormatter(total)}</h1>
    </>
  )
}
```

10. CONCLUSION

making this application we assure that this application will help its users to manage the cost of their daily expenditure. It will guide them and make them aware about their daily expenses. It will prove to be helpful for the people who are frustrated with their daily budget management, irritated because of the amount of expenses and wish to manage money and to preserve the record of their daily cost which may be useful to change their way of spending money. In short, this application will help its users to overcome the wastage of money.

11. REFERNCES

1. https://console.firebase.google.com/u/0/
2. https://firebase.google.com/docs/functions
3. https://www.srijan.net/resources/top-13-must-have-features-for-your-expense-tracking-app#:~:text=An%20expense%20tracking%20app%20is,%2C%20weekly%2C%20and%20monthly%20basis.
4. https://firebase.google.com/docs/functions/unit-testing
5. https://youtu.be/U5aeM5dvUpA?list=PLl-K7zZEsYLmOF_07IayrTntevxtbUxDL