A Thesis/Project/Dissertation

Reporton

VOICE OPERATED REAL TIME EXPENSE TRACKER USING REACT JAVASCRIPT

Submitted in partial fulfilment

requirement for the award of the degree of

Bachelor's of Technology in Computer Science Engineering

GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision of

Dr. Kuldeep Singh Kaswan

Associate Professor

Submitted By

Shobhit Sharma and Parth Verma

18SCSE1010260 & 18SCSE1010007

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING GALGOTIAS UNIVERSITY,GREATER NOIDA UP INDIA
DECEMBER,2021

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the thesis/project/dissertation, entitled **"Real Time Voice Operated Expense Tracker using React JavaScript"** in partial fulfilment of the requirements for the award of the Bachelors of Technology submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of month, Year to Month and Year, under the supervision of Name **"Mr. Kuldeep Singh Kaswan"** Designation, Department of Computer Science and Engineering/Computer
Application and Information and Science, of School of Computing Science and Engineering , Galgotias University, Greater Noida.

The matter presented in the thesis/project/dissertation has not been submitted by us for the award of any other degree of this or any other places.

Shobhit Sharma(18SCSE1010260) & Parth Verma(18SCSE1010007)

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Dr.Kuldeep Singh Kaswan

ACKNOWLEDGEMENT

Apart from the efforts of our, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We would like to show my greatest appreciation to “**Dr. Kuldeep Singh Kaswan”** and also our dean “**Dr.Munish Sabarwal”**. We can’t say thank you enough for his tremendous support and help. We feel motivated and encouraged every time, We attend his meeting. Without his encouragement and guidance this project would not have materialized.

The guidance and support received from all the members who contributed and who are contributing to this project, was vital for the success of the project. We are grateful for their constant support and help.

CERTIFICATE

The FinalThesis/Project/ Dissertation Viva-Voce examination of "PARTH VERMA(18SCSE1010007) & SHOBHIT SHARMA(18SCSE1010260) " has been held on _________________ and their work is recommended for the award of B.Tech(CSE).

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator *Signature of Dean*

Date: December, 2021

Place: Greater Noida

ABSTRACT

In today’s technological society, the major role played in people’s life is played by earning. Each and every person requires money to live and adapt the standard of living. The today’s developing generation is growing day by day and in this process of the betterment of the people’s living quality, the demands of making people's life efficient and quality of life higher money is the major demand.

Therefore, in context to that particular scenario we as a web developer invested our technical and logical skills in making a user friendly and using latest technology, we created a web application called as “The Real-time voice operated expense and transaction tracking” web application.

The Real-time voice operated expense and transaction tracking application is a web application that acts as a wallet to the people and tracks all the transaction and expenses that a person is doing in his/her daily life. This application is a voice command activated application that contributes in providing user interactive UI allowing the people to easily add and remove the money as per the required date.

This is a totally automatic website that works on voice command created using React JavaScript libraries. The major components covered in this voice based application is React JavaScript and MongoDB database used to store the user data and API’s for User authentication and for data transfer to backend database. A user using the application can see a wide range of features in respective categories. This application each user a satisfaction for the login and safety of each and every particular user also its very user friendly and user oriented application.

Since this application is a web application therefore it can run on any platform either its MacOS, Android OS, Windows etc. All the operating system supports this newly designed

web application. The reason being we are using React JavaScript is that react is latest technology and also it provides advanced web feature and also a bit faster than normal JavaScript due to the introduction of JSX. The Real-time expense tracker reduces human workload and also allows user to work smartly in this world of fast technology.

This project is based on the objective to save money and work smartly for the future requirements where the user with disability can also operate this application and save money. This innovation will surely lead to equality in life of people.

Table of Contents

INTRODUCTION

1.1 Introduction

In today's firmament of Software Development there is a lot of upgradation in Interface Designing and Application Development. The convictions and execution details are changing as the people are directing the mode of application's development.

In this world of total technology and the complex world of web development there are some amazing and awesome programs to build and patterns for most web developers. Moreover, the design must have the capability to turn the path towards remodelling of design instead of lending itself to the initial common practices. The other modes i.e., web apps are one of those places where web developers can change their creative side and hope that their solutions can be more effective and flexible. In this report we will describe the most exciting and fashionable journey down the road to the web program. The desire of this job is to set up Real time expense tracking application where one can save a sufficient amount of money and protect his or her future and also lead himself/herself to a growing path. The Real-Time expense tracking application is like indirect wallet on the Internet where consumers can update the details of his/her expense in order to maximize the profit and reduce the loss of money. Due to the introduction of voice command one who is not able to work properly or is handicap can also make the most of this particular web application. Hence this also oppose discrimination.

1.2 Objective of project

The main objective of the project is to create a "VOICE BASED USER EXPENSE TRACKER APPLICATION" that allows users with the functionality of their saving and expenses tracking throughout the month. This allow the person to do expense as per the requirement and prevent wastage of money. Also due to its voice controlling feature one can add and delete the record as per the required respective date and day of the month. The voice activation feature is a positive point for the ease of handicap people that can also operate the application and work efficiently in their lives. The use of API's and user authentication help the data of each user to stay protected from outside world. Further the use of Mongo DB as a database is also a plus point for the application. MongoDB is one of the highly secure database that protects the data in efficient manner.

In order to get the user data from backend to frontend API's will be used that makes it very easy to work on.

With the introduction to Real time voice operated expense tracker, consumers aren't required to wander and waste their precious time while crawling on the Internet in search of their desired money tracking application. Instead, they only have to have the internet on the computer and just go to our

1.3 Features

Real time expense tracker, and save their record and also update them as per the requirements, based on their personal interest. During this acquisition information, one can check whether the site has its own brain or not, if you can have the record directly online, otherwise you can switch to a local bank and save your record in the bank and check the details via passbook. This will help each customer to save more time and work.

The real time expense tracking system not only reduces human workload but also saves time, space for easy tracking of monthly/yearly expense and work accordingly. Specifically, in today's world, close international relations and globalization all play a key role in promoting the ever-changing nature of real time tracking application. In addition, it also has some practical value and the user is also provided with reasonable and good advices to help the people on how much the need to save in order to have a wealthy living in this firmament.

The Real time voice operated expense tracking uses Visual Studio Code as a developing atmosphere, with JavaScript and JSX, React JavaScript as the primary languages for backend and frontend. Moreover, the system consists of MongoDB as an active database to store the various user record details as well as including all User details, the front interface design is designed using React JavaScript.

LITERATURE SURVEY

2.1 Research and Development of Web Application

We will test the app in three environments including hardware, software and network. Test hardware environment will be MacBook air laptop; software environment is mac m1 chip. The results of testing each feature on a mobile phone and a computer website revealed that the voice operated expense tracker works smoothly and there is no advertising. This provide an advantage for us to get the application work for the outside world perfectly and as per user’s interest.

2.2 Design of Web based Expense Tracker

Many people in today’s fast life do not take care about the money they are spending out. Therefore with a rapid development of technology and cost on goods its very difficult to save money now-a-days, so therefore we invested our technical knowledge in creation of a perfectly working application for tracking the entire expenses over the month.

This is a plus point for the people those who want to save the money for future use.

2.3 The Website Application Development College Challenge

Web application development college challenge has only been held two times, but it greatly encourages and promotes the creativity of the college students. It will be more difficult to win an award as the event attracts more and more competitive teams. This challenge gives us an opportunity to learn about that a lot

of ideas we think about can be implemented on android platform.

2.4 The Website- A Widely Growing Platform With its Web based Applications

Web platform is one of the most widely used web application based platform these days and also enhancing its use for making betterment in different areas of life. Website as a platform is based on the Platform as a Service and is monitored by Google, Amazon and other technical companies and primarily designed for electronic devices and different browser like safari,chrome, MozillaFirefox etc.

The advanced Smart Web applications, real-time and wireless sensor network are widening their service areas. Web application a disruptive technology, which was introduced initially on August 6, 1991 without fanfare, British Computer Scientist Tim-Berners lee published the first web application, but has much wider potential.

In this paper we are studying, one of the smart and enhancing web application which are based on Automated and tracking from remote distance. These application helps students, teachers, parents, patients and users of home appliance as anytime and anywhere basis for their daily basis need and more over we would take this technology forward for android development using a beautiful technology called as “React Native and Flutter”. These application are very useful for mobile development is one of the most rending application now days.

PROBLEM FORMULATION

3.1 Problem Formulation

Many users in their daily life try to save money in order to save money and use that for further use. Therefore in order to simplify their work and help them to save money and track their expense this application is the best practice to work with. With the rapid development of cost and expense, we thought that we will try to savemoney so if user is not able to track the expense or find difficulty in having the record then user can use the application for their personal use and save as much they want because this will help one to have daily expense record.

Money wastage irritates every person while doing purchase. So, in our expense tracking app there will be no case of wastage of money. We will also try that our users are benefited with the application advantages.

3.1 Tools used for implementation

- Visual studio
- JavaScript(Mainly ReactJs)
- JSX

SYSTEM DESIGN

4.1 Requirements

4.1.1 Practical Requirements

The Practical Requirements is a description of services that the software must offer. It defines the software system or its component. The system must fulfil the following functional requirements.

Record Customer Details -

The system must be able to record the details and information of the new customers that are being added as a new account in order to manage the data.

Catalogue of All Record -

The system must be able to record the details i.e. Catalogue of all the products in which the user is interested in order to make it easy for the user to save the records based on the particular category they want. This feature of creating catalogues of all income provides the advantage to each user to save the record based on interest in no time based on their desirable result.

Record Customer Review-

After developing any application, it is the major requirement to have the review of the other users in order to know whether the users are comfortable with the applications or the applications needs to be upgraded in order the fulfil all the requirements of the users that are lacking in it.

Therefore, the system is granted with customer review feature.

Record Daily Expense-

The Real-time voice operated expense tracker is a web application that allows the users to store the complete details of overall expense he/she is doing on every particular day of week.

Update Daily Money Record-

In the era of web development, the system fulfills the feature of upgrading the daily record and suggesting the user with the complete expense tracking of the entire day.

Temporary Storage-

The system also provides the user with temporary storage for storing the items they require at a limited duration of time i.e. for 29 days. After the duration is completed the user would be required to renew the data.

4.1.2 Non-Practical Requirements

The Non-Practical Requirements describes system's features such as **reliability, maintenance, performance, scalability, security,** and **usability**. This serves as restrictions on the layout of the system across the different backlogs. The system must fulfil the following non-functional requirements.

Availability -

The system provides the 24*7 availability which means that users can access the application and save the record also track the record on any day and at every time.

Data Confidentiality -

Nowadays people are getting advanced in the area of website development and ethical hacking. Moreover, the hackers can easily break into account of other users and steal their data.

In contrast, The Real time voice operated expense tracker provides the data confidentiality feature that protects the data of the user from being accessed by unauthorized users. In other words, only the people who are authorized to do so can gain access to sensitive data.

Improved Design and Performance -

The Real time voice operated expense tracker provides improved designing i.e. Better Interface as compared to other applications and also facilitates with better performance and faster working of the applications.

Flexible Service Based Architecture –

The Real time voice operated expense tracker also provides a flexible service-based architecture as compared to other systems.

4.1.3 FUNCTIONAL REQUIREMENT

System must fulfil the given requirements

1. Must be able record new user
2. Must keep catalogue of all the services
3. Must record user review
4. Must record daily updates by user
5. Must provide daily track on monthly income and expense

4.1.4 NON-FUNCTIONAL REQUIREMENT

System must fulfil the following non-functional requirement on availability of internet

1. Availability 24*7
2. Customer data confidentiality
3. Better design and performance
4. Flexible service-based architecture

4.1.5 SYSTEM REQUIREMENTS

4.1.5.1 SOFTWARE REQUIREMENT

Software required to make the following products were

1. Visual Studio Code
2. MacOS
3. Language: React JavaScript
4. JSX
5. Database used MongoDB database.

4.1.5.2HARDWARE REQUIREMENT

Hardware required to make the following products were

1. Laptop with specifications similar or above: MacBook air(M1,2020), 8gb RAM, Apple M1 chip
2. Hardware required to the product

4.2 FUNCTIONALITY

4.2.1 SYSTEM FUNCTIONALITY

The system consists of three section on a single page application. Firstly, the main section is the middle section that is the main working section where user is allowed to create an entry in the database based on the income they are having. When the user wants to enter a data as a money into the desired category for that desired type on a desired date then he can put that by doing the instructions mentioned on the interface or can manually create the record. This information may include the amount to be submitted as a record, category of the record i.e. in which type of income/expense that user collected or spent, date i.e. on which date user did the transaction and other details. Moreover, the details of user record can be modified and also deleted whenever the user want to delete. The left section consists of income tracking graph that's represented as a pie chart and above that total amount in the income is also mentioned. Once the user selects any of those category whether its "Income" or "Expense" then user will get the option to proceed as per the value entered.

Once the user proceeds with the storing the record, all those record are displayed in the middle section as a list of complete transaction. Then user will choose based on the choice and select to reduce the amount in the form of expense or increase the INR based on income choice option.

Finally, the pie chart representation of the data is showing on left and right section.

4.3 BACKGROUND MODULE FUNCTION

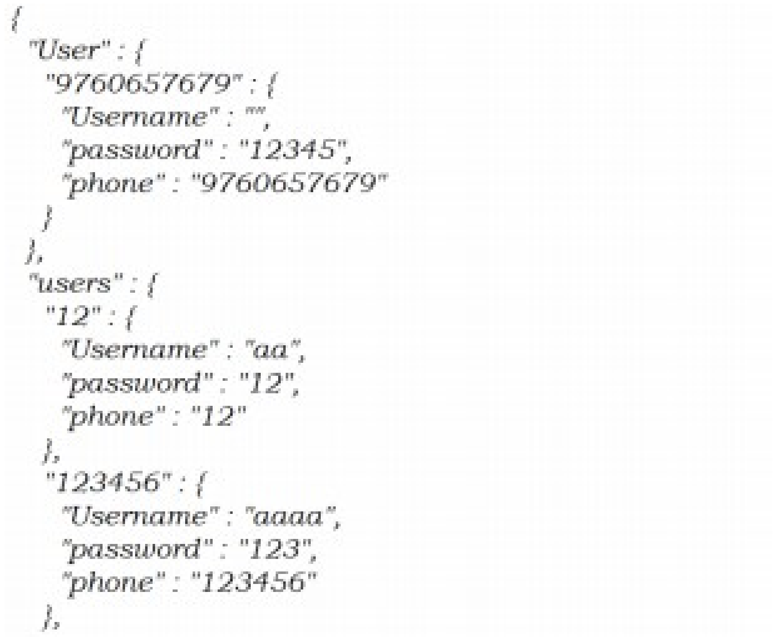
(1) Classification management: read the division, add the division of the second division, change the first division of the second division, remove the first division of the second division.

(2) Record Management-Background directors have identical practicality as foreground users in looking for saving records, however directors may manage record data like adding record, and deleting data.

4.4 LOGICAL STRUCTURE OF DB

In terms of the operating style of the program, by analysing the structure of the program and its requirements, we will determine whether the system should have these table users (t_user), degree management table (t_admin) and record table (t_comment).

A partial database is given as follows:



```
{
  "User" : {
    "9760657679" : {
      "Username" : "",
      "password" : "12345",
      "phone" : "9760657679"
    }
  },
  "users" : {
    "12" : {
      "Username" : "aa",
      "password" : "12",
      "phone" : "12"
    },
    "123456" : {
      "Username" : "aaaa",
      "password" : "123",
      "phone" : "123456"
    },
```

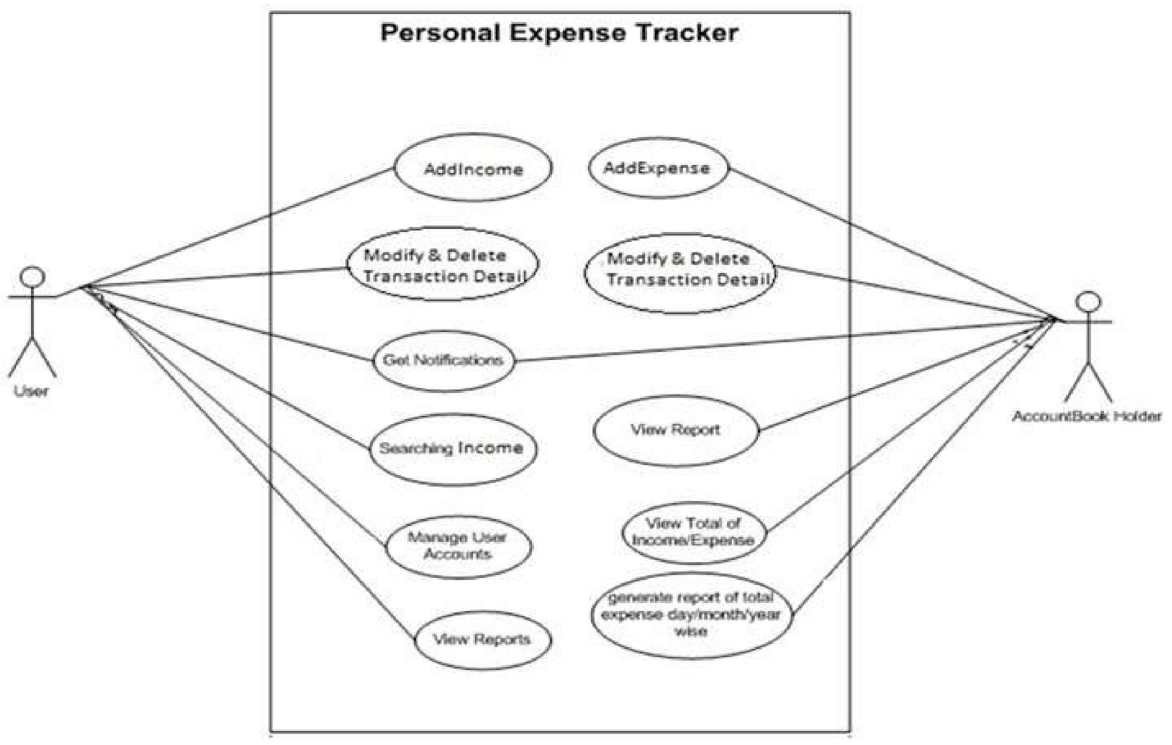
4.5 USE CASE DIAGRAM

The figure-3.4.1 is that the best explanation of Use Case Diagram for the net Expense Tracking system, it's an illustration of a user's interaction with the system that shows the link between the user and also the totally different use cases within which the user is concerned.

The figure-3.4.1 consists of actors, use cases and their relationships. The figure explains however the user connects with administration. The administration provides totally different options to patrons or user as shown. The user can place orders based on their choices. They can also place new record for a new category.

The User can browse different categories of income/expense, add amount to the record and can check out with the expense and can proceed with the tracking options. Moreover, user can cancel the existing record.

After the record is successfully stored from the payment then we can check the details accordingly. The user can check the his daily income, expense and can save a lot more money.


Personal Expense Tracker
AddIncome
AddExpense
Modify & Delete Transaction Detail
Modify & Delete Transaction Detail
User
Get Notifications
AccountBook Holder
View Report
Searching Income
Manage User Accounts
View Total of Income/Expense
generate report of total expense day/month/year wise
View Reports

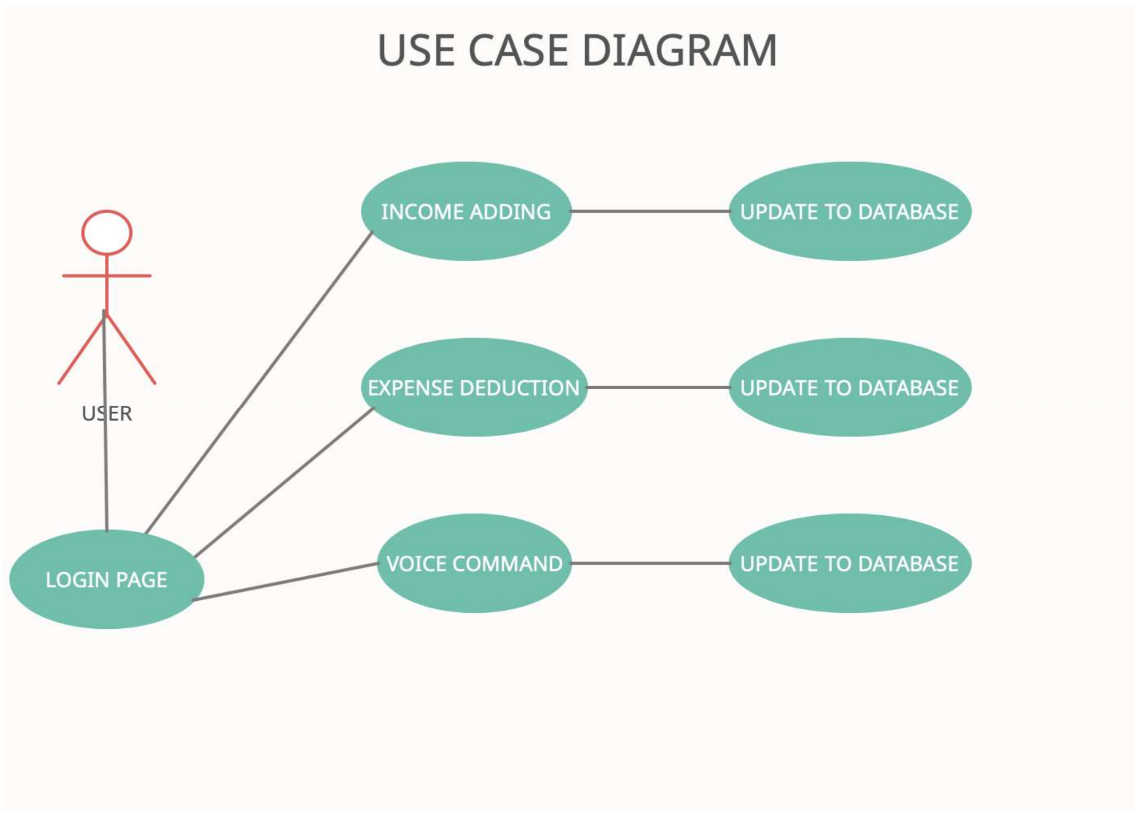


Fig-3.4.1

Use Case Diagram for Real Time Expense Tracker

4.6 FEASIBILITY ANALYSIS

4.6.1 Economic viability

Economic practicability is that the price and supplying outlook for a business project or endeavour. In different words, it's a criterion for crucial the ultimate market position of a web site. sensible economic practicability is useful for project implementation. In this system, from info to the web developer tools used area unit all free, therefore the price of the event of the system is just endowed in time and energy, therefore, the system is economical.

4.6.2 Operational practicability

Operational practicability is the parameter how well a projected system solves the issues, and takes advantage of the opportunities known throughout scope definition and the way it satisfies necessities known within the requirements analysis part of system development.

In this system, the user merely enters the book you wish to look or connected info to question, nice convenience. The user's read, reflects the human aspect. Moreover, the system is intended to realize the page layer, business logic layer and information storage layer separation, the system is additional stable and versatile. Therefore, in short, we are able to say that the system is totally possible.

MODULE DESCRIPTION

5.1 DESCRIPTION OF MODULE

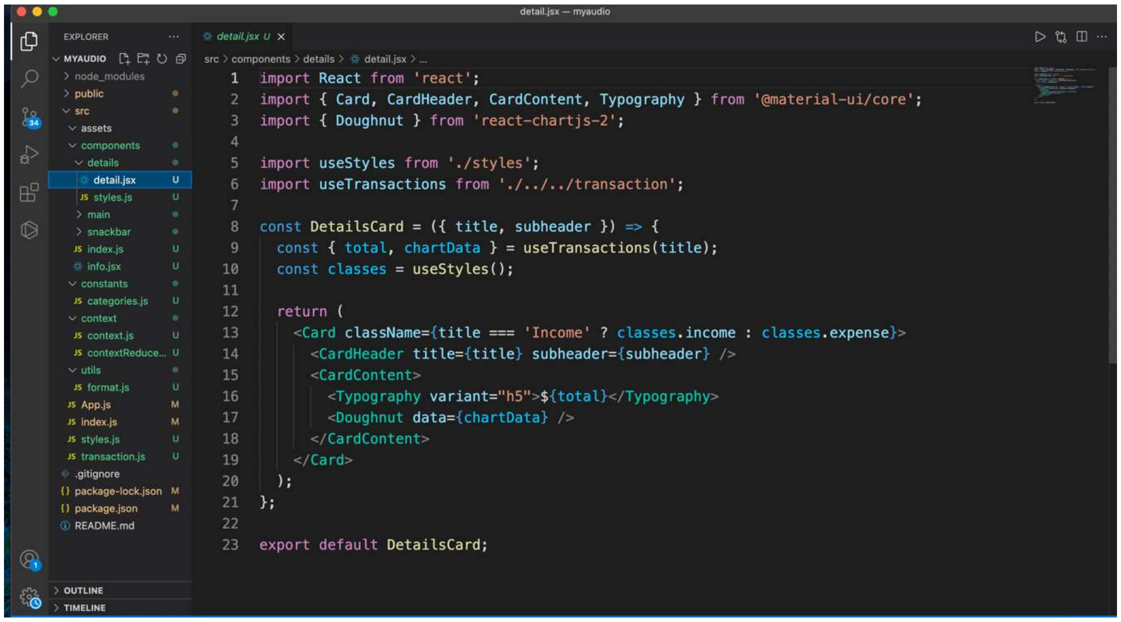
The complete audio expense module is divided into multiple components. The main reason why we call the complete application in components is that react JavaScript is a single page application in which no redirecting occur for different pages rather one single page all the components are rendered as per the routes. Moreover, the speed of rendering the components is a bit faster therefore we are using this technology as compare to other frameworks and other technologies.

Coming on to the different components included in this application are as follows-

1. Details Component-

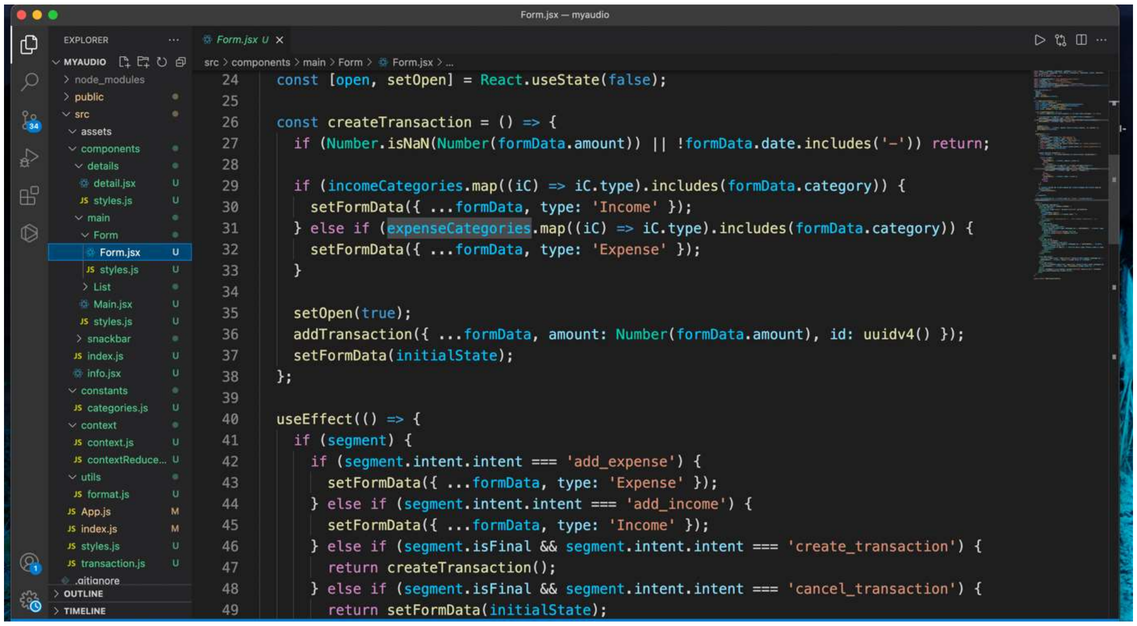
This component is the main lead component on which all the components are rendered one by one as per the user's request. Moreover, as the name suggest this components contains all the details for the entire application and all the props are transferred from this component to the entire application.

Therefore it is the main component or we can say it as the parent component of the application.

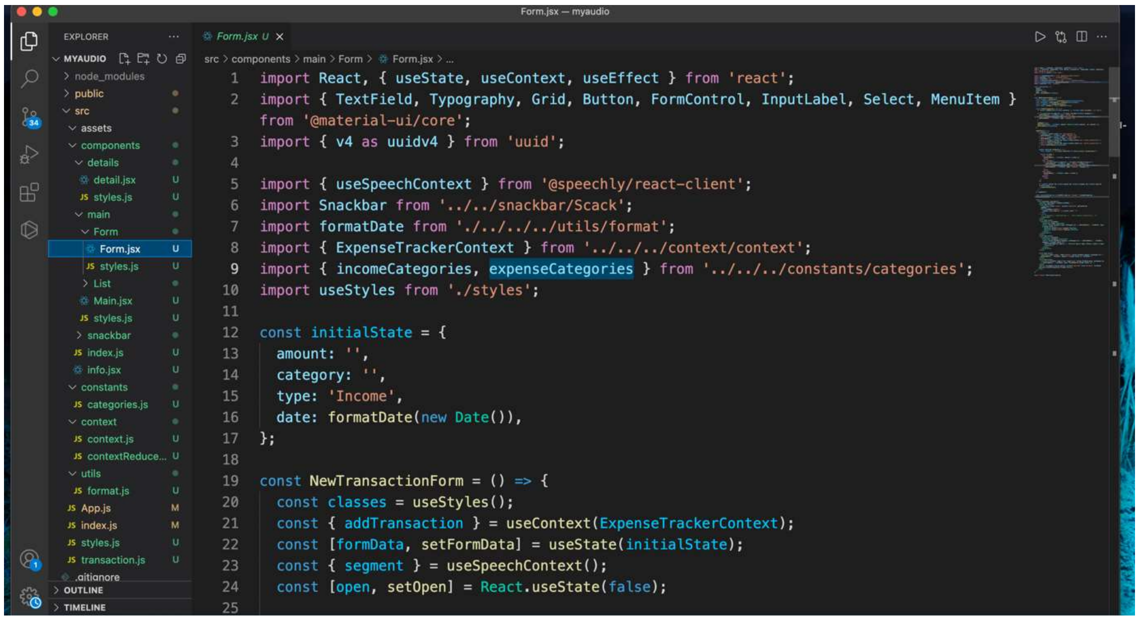


```jsx
import React from 'react';
import { Card, CardHeader, CardContent, Typography } from '@material-ui/core';
import { Doughnut } from 'react-chartjs-2';

import useStyles from './styles';
import useTransactions from './../../transaction';

const DetailsCard = ({ title, subheader }) => {
  const { total, chartData } = useTransactions(title);
  const classes = useStyles();

  return (
    <Card className={title === 'Income' ? classes.income : classes.expense}>
      <CardHeader title={title} subheader={subheader} />
      <CardContent>
        <Typography variant="h5">${total}</Typography>
        <Doughnut data={chartData} />
      </CardContent>
    </Card>
  );
};

export default DetailsCard;
```

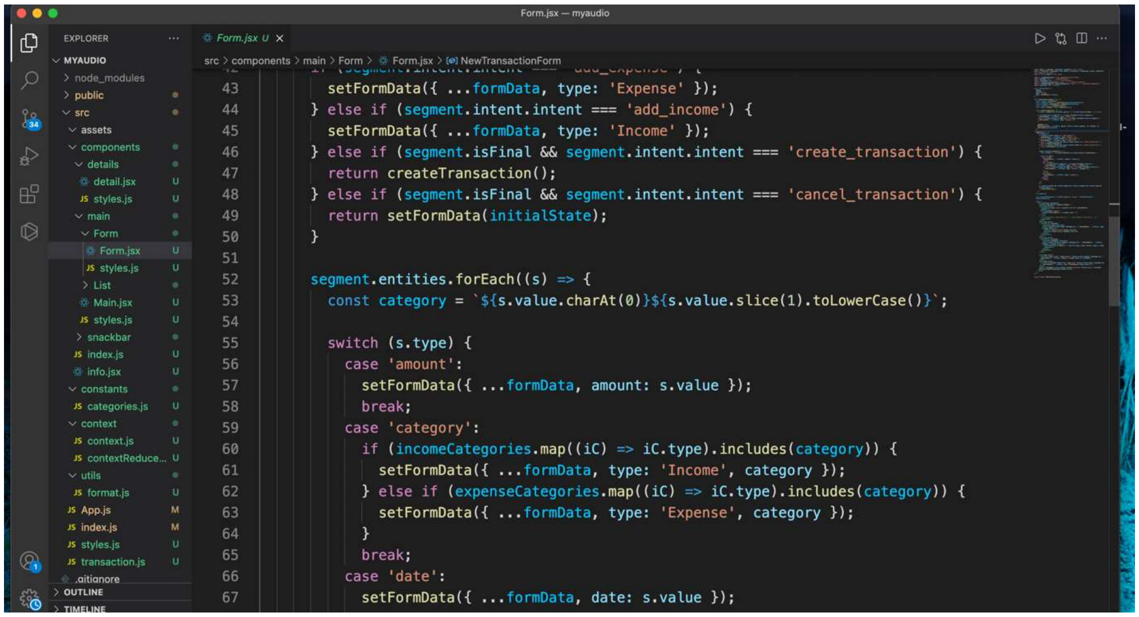
2. **Form Component-** This component is a form component that render the main input areas of the application that consist of inputs with different types like text area, button, date, number, text field etc. This component is responsible for taking the input from user and passing the input values to forward component for checking up the data and doing the entry.



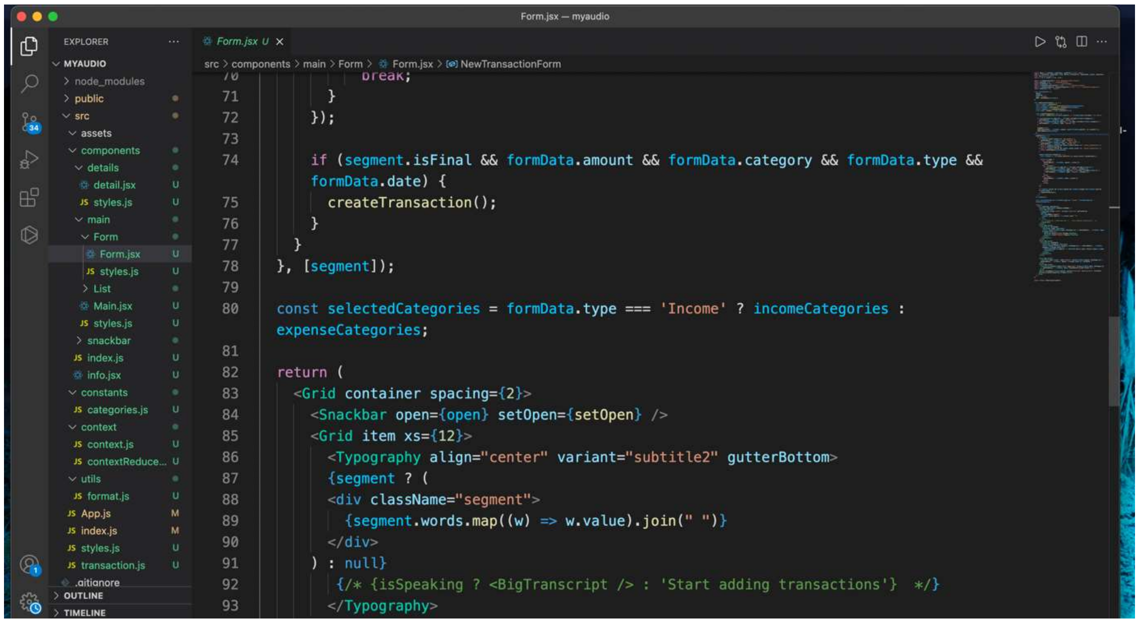
```
  const [open, setOpen] = React.useState(false);

  const createTransaction = () => {
    if (Number.isNaN(Number(formData.amount)) || !formData.date.includes('-')) return;

    if (incomeCategories.map((iC) => iC.type).includes(formData.category)) {
      setFormData({ ...formData, type: 'Income' });
    } else if (expenseCategories.map((iC) => iC.type).includes(formData.category)) {
      setFormData({ ...formData, type: 'Expense' });
    }

    setOpen(true);
    addTransaction({ ...formData, amount: Number(formData.amount), id: uuidv4() });
    setFormData(initialState);
  };

  useEffect(() => {
    if (segment) {
      if (segment.intent.intent === 'add_expense') {
        setFormData({ ...formData, type: 'Expense' });
      } else if (segment.intent.intent === 'add_income') {
        setFormData({ ...formData, type: 'Income' });
      } else if (segment.isFinal && segment.intent.intent === 'create_transaction') {
        return createTransaction();
      } else if (segment.isFinal && segment.intent.intent === 'cancel_transaction') {
        return setFormData(initialState);
```



```
import React, { useState, useContext, useEffect } from 'react';
import { TextField, Typography, Grid, Button, FormControl, InputLabel, Select, MenuItem }
from '@material-ui/core';
import { v4 as uuidv4 } from 'uuid';

import { useSpeechContext } from '@speechly/react-client';
import Snackbar from '../../snackbar/Scack';
import formatDate from './../../../utils/format';
import { ExpenseTrackerContext } from '../../../context/context';
import { incomeCategories, expenseCategories } from '../../../constants/categories';
import useStyles from './styles';

const initialState = {
  amount: '',
  category: '',
  type: 'Income',
  date: formatDate(new Date()),
};

const NewTransactionForm = () => {
  const classes = useStyles();
  const { addTransaction } = useContext(ExpenseTrackerContext);
  const [formData, setFormData] = useState(initialState);
  const { segment } = useSpeechContext();
  const [open, setOpen] = React.useState(false);
```


Form.jsx — myaudio
EXPLORER
MYAUDIO
Form.jsx U
src > components > main > Form > Form.jsx > NewTransactionForm
OUTLINE
TIMELINE

```
      setFormData({ ...formData, type: 'Expense' });
    } else if (segment.intent.intent === 'add_income') {
      setFormData({ ...formData, type: 'Income' });
    } else if (segment.isFinal && segment.intent.intent === 'create_transaction') {
      return createTransaction();
    } else if (segment.isFinal && segment.intent.intent === 'cancel_transaction') {
      return setFormData(initialState);
    }

    segment.entities.forEach((s) => {
      const category = `${s.value.charAt(0)}${s.value.slice(1).toLowerCase()}`;

      switch (s.type) {
        case 'amount':
          setFormData({ ...formData, amount: s.value });
          break;
        case 'category':
          if (incomeCategories.map((iC) => iC.type).includes(category)) {
            setFormData({ ...formData, type: 'Income', category });
          } else if (expenseCategories.map((iC) => iC.type).includes(category)) {
            setFormData({ ...formData, type: 'Expense', category });
          }
          break;
        case 'date':
          setFormData({ ...formData, date: s.value });
```

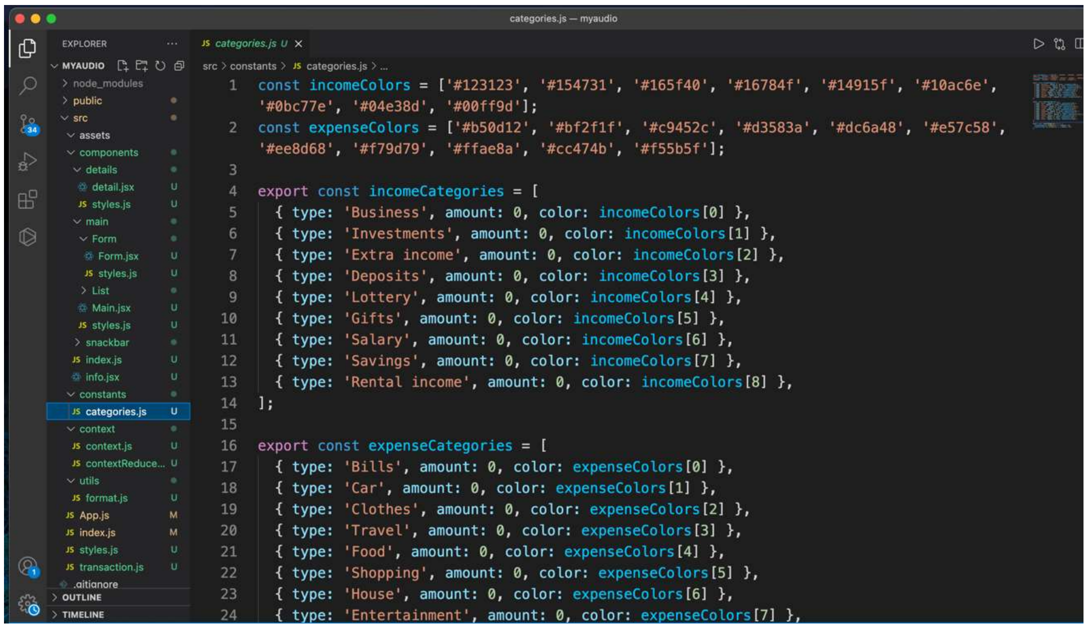

Form.jsx — myaudio
EXPLORER
MYAUDIO
Form.jsx U
src > components > main > Form > Form.jsx > NewTransactionForm
OUTLINE
TIMELINE

```
          break;
      }
    });

    if (segment.isFinal && formData.amount && formData.category && formData.type &&
    formData.date) {
      createTransaction();
    }
  }
}, [segment]);

const selectedCategories = formData.type === 'Income' ? incomeCategories :
expenseCategories;

return (
  <Grid container spacing={2}>
    <Snackbar open={open} setOpen={setOpen} />
    <Grid item xs={12}>
      <Typography align="center" variant="subtitle2" gutterBottom>
      {segment ? (
      <div className="segment">
        {segment.words.map((w) => w.value).join(" ")}
      </div>
    ) : null}
       {/* {isSpeaking ? <BigTranscript /> : 'Start adding transactions'}  */}
      </Typography>
```

3. **<u>Category Component-</u>**

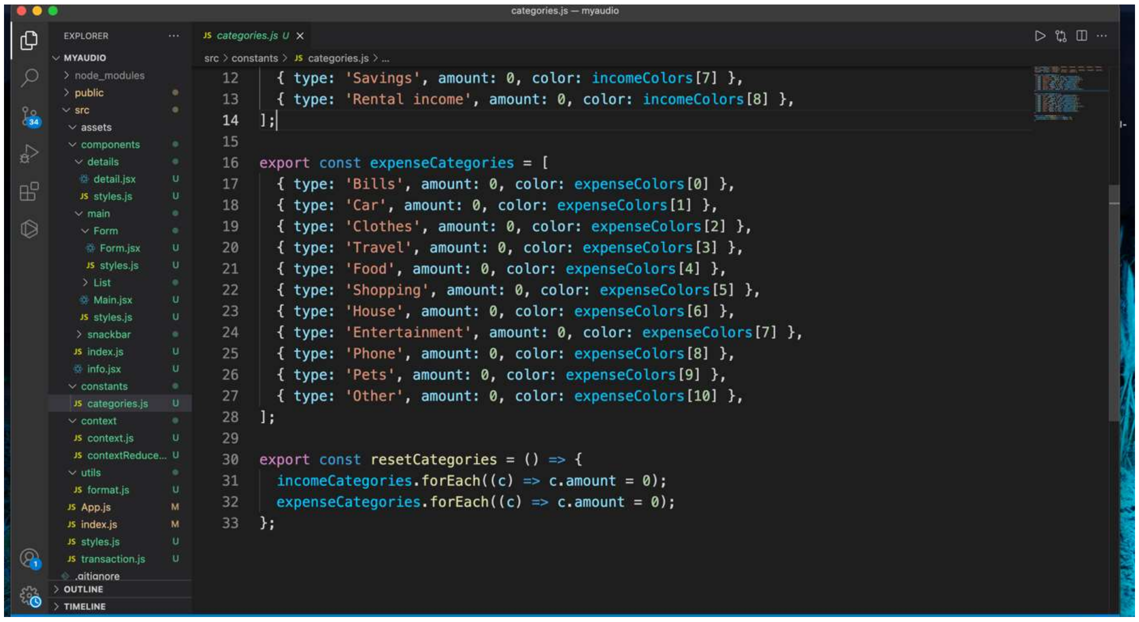
The main advantage of including this component is that it provide you features of adding your record based on the category of income you want e.g. if a user want to add income for business, investment or a user has got some lottery or gift or got bonus from the boss then it can add as per the desire what he wants.

This provide an easy for user to not get confused why he got the money or what was the main objective of adding the record in the application. Moreover based on the category he added he can tally the data.



```javascript
const incomeColors = ['#123123', '#154731', '#165f40', '#16784f', '#14915f', '#10ac6e',
'#0bc77e', '#04e38d', '#00ff9d'];
const expenseColors = ['#b50d12', '#bf2f1f', '#c9452c', '#d3583a', '#dc6a48', '#e57c58',
'#ee8d68', '#f79d79', '#ffae8a', '#cc474b', '#f55b5f'];

export const incomeCategories = [
  { type: 'Business', amount: 0, color: incomeColors[0] },
  { type: 'Investments', amount: 0, color: incomeColors[1] },
  { type: 'Extra income', amount: 0, color: incomeColors[2] },
  { type: 'Deposits', amount: 0, color: incomeColors[3] },
  { type: 'Lottery', amount: 0, color: incomeColors[4] },
  { type: 'Gifts', amount: 0, color: incomeColors[5] },
  { type: 'Salary', amount: 0, color: incomeColors[6] },
  { type: 'Savings', amount: 0, color: incomeColors[7] },
  { type: 'Rental income', amount: 0, color: incomeColors[8] },
];

export const expenseCategories = [
  { type: 'Bills', amount: 0, color: expenseColors[0] },
  { type: 'Car', amount: 0, color: expenseColors[1] },
  { type: 'Clothes', amount: 0, color: expenseColors[2] },
  { type: 'Travel', amount: 0, color: expenseColors[3] },
  { type: 'Food', amount: 0, color: expenseColors[4] },
  { type: 'Shopping', amount: 0, color: expenseColors[5] },
  { type: 'House', amount: 0, color: expenseColors[6] },
  { type: 'Entertainment', amount: 0, color: expenseColors[7] },
```

Moreover only income but also the expense can be categorized as per the desire that whether the user pays some bills, car insurance, shopping, or the user went to watch some movie or had food last night out. Hence this feature work for providing the best convenance to the user for categorize the income and expense he is spending on.

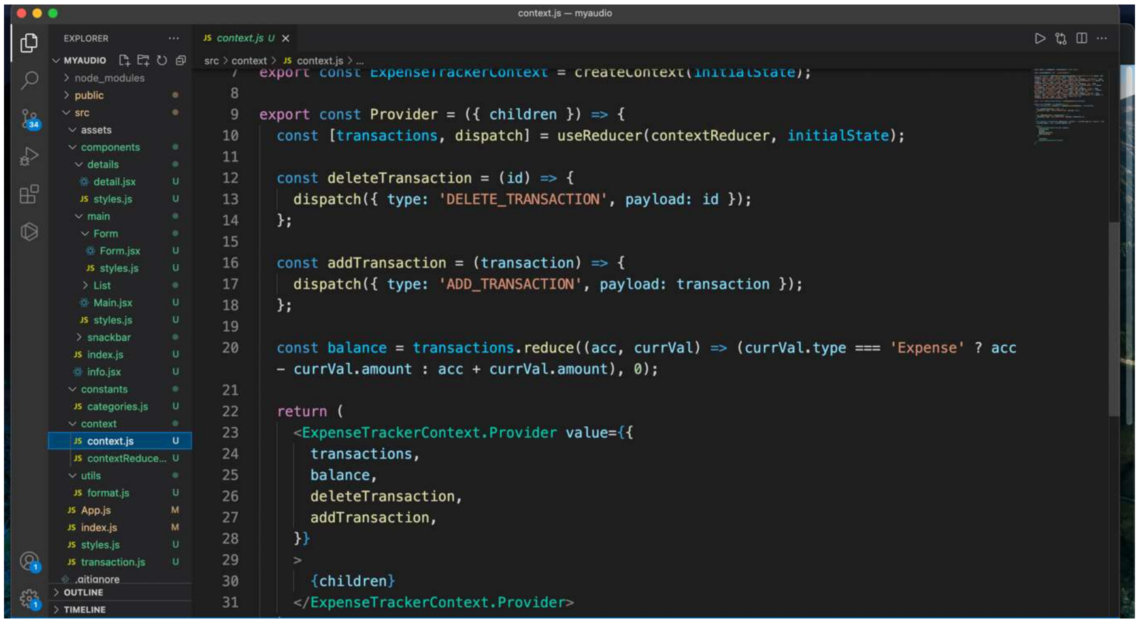


```
  { type: 'Savings', amount: 0, color: incomeColors[7] },
  { type: 'Rental income', amount: 0, color: incomeColors[8] },
];

export const expenseCategories = [
  { type: 'Bills', amount: 0, color: expenseColors[0] },
  { type: 'Car', amount: 0, color: expenseColors[1] },
  { type: 'Clothes', amount: 0, color: expenseColors[2] },
  { type: 'Travel', amount: 0, color: expenseColors[3] },
  { type: 'Food', amount: 0, color: expenseColors[4] },
  { type: 'Shopping', amount: 0, color: expenseColors[5] },
  { type: 'House', amount: 0, color: expenseColors[6] },
  { type: 'Entertainment', amount: 0, color: expenseColors[7] },
  { type: 'Phone', amount: 0, color: expenseColors[8] },
  { type: 'Pets', amount: 0, color: expenseColors[9] },
  { type: 'Other', amount: 0, color: expenseColors[10] },
];

export const resetCategories = () => {
  incomeCategories.forEach((c) => c.amount = 0);
  expenseCategories.forEach((c) => c.amount = 0);
};
```

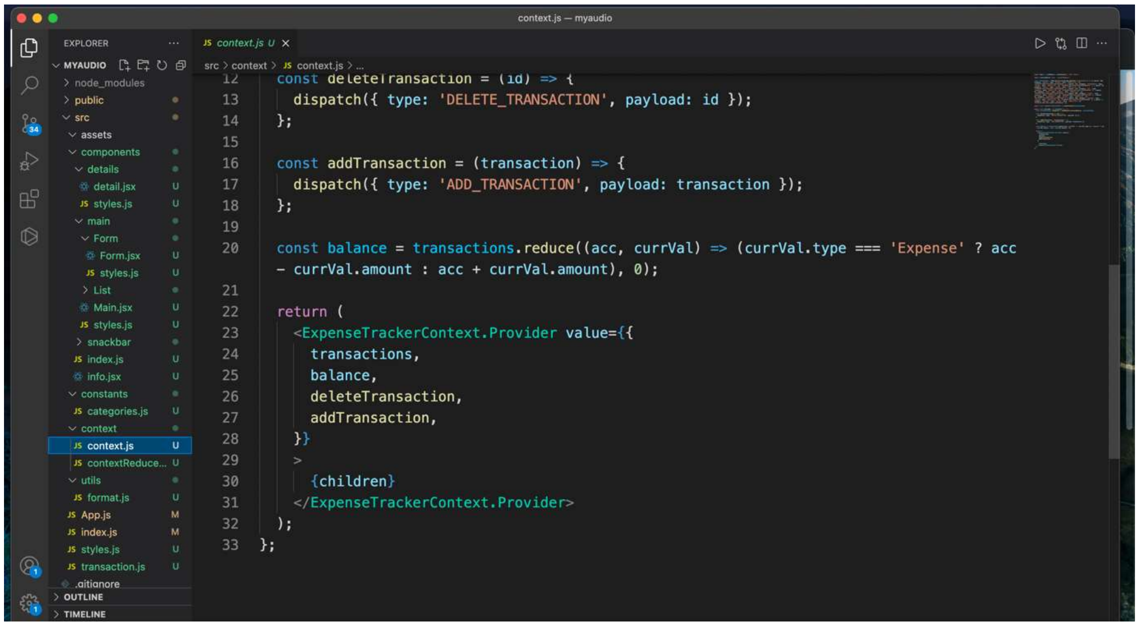
4. **<u>Context Component-</u>** This component is all about the addition and reduction of the record as per the user's choice user can add income and do expense.

All this process comes under this component that monitor or basically work for adding and remove record. In this component there's introduction of reducer that comes in React Redux part in which a there's a container and the permission is transferred via dispatcher for addition and deletion of record.

And based on the action particular payload is rendered that give user a signal that data is added or deleted.

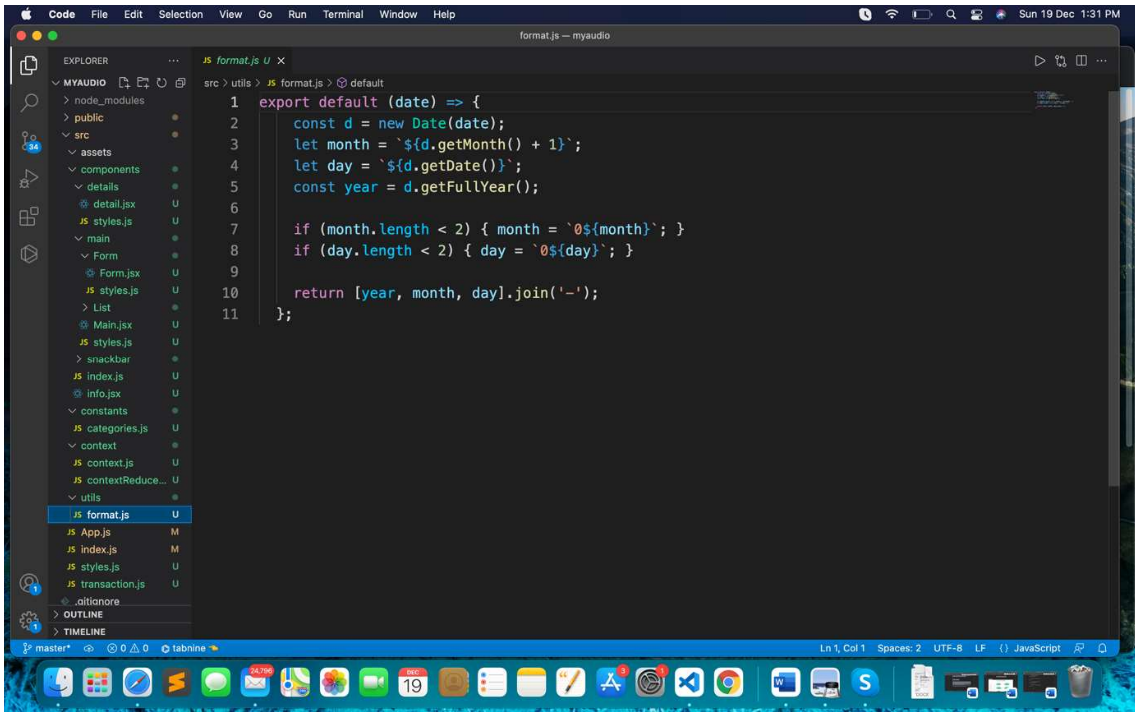


```
export const ExpenseTrackerContext = createContext(initialState);

export const Provider = ({ children }) => {
  const [transactions, dispatch] = useReducer(contextReducer, initialState);

  const deleteTransaction = (id) => {
    dispatch({ type: 'DELETE_TRANSACTION', payload: id });
  };

  const addTransaction = (transaction) => {
    dispatch({ type: 'ADD_TRANSACTION', payload: transaction });
  };

  const balance = transactions.reduce((acc, currVal) => (currVal.type === 'Expense' ? acc
- currVal.amount : acc + currVal.amount), 0);

  return (
    <ExpenseTrackerContext.Provider value={{
      transactions,
      balance,
      deleteTransaction,
      addTransaction,
    }}
    >
      {children}
    </ExpenseTrackerContext.Provider>
```



```
  const deleteTransaction = (id) => {
    dispatch({ type: 'DELETE_TRANSACTION', payload: id });
  };

  const addTransaction = (transaction) => {
    dispatch({ type: 'ADD_TRANSACTION', payload: transaction });
  };

  const balance = transactions.reduce((acc, currVal) => (currVal.type === 'Expense' ? acc
- currVal.amount : acc + currVal.amount), 0);

  return (
    <ExpenseTrackerContext.Provider value={{
      transactions,
      balance,
      deleteTransaction,
      addTransaction,
    }}
    >
      {children}
    </ExpenseTrackerContext.Provider>
  );
};
```

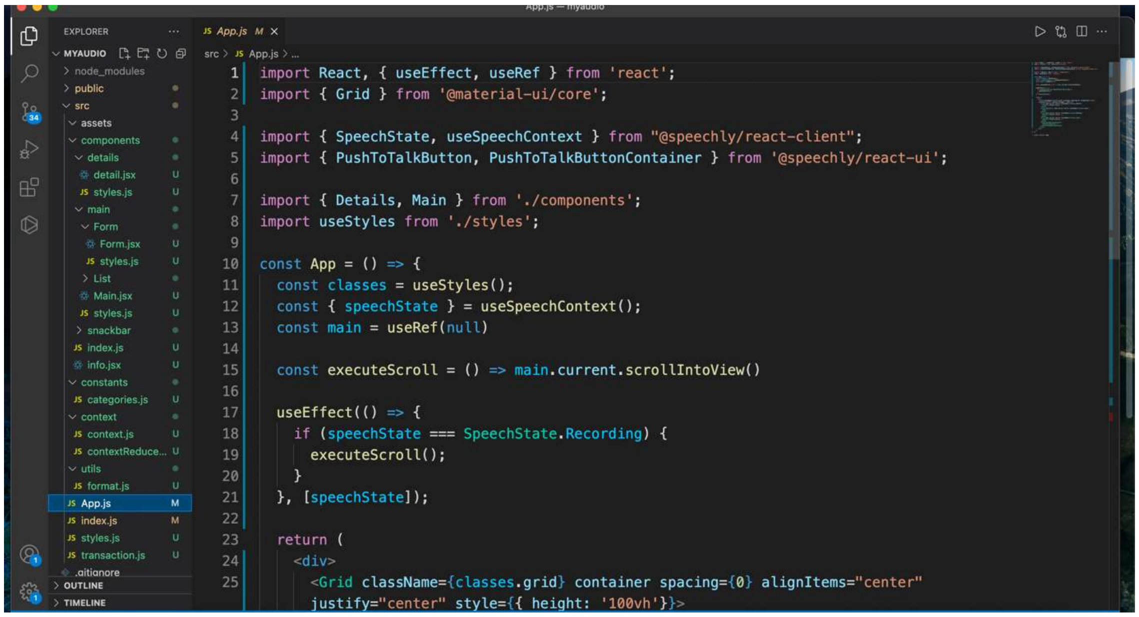
5. **<u>Format Component-</u>** This component is basically responsible for setting up the date format used in the application for storing the date of the record on which particular date the process has done in order to provide simplicity to user to track the date as well.



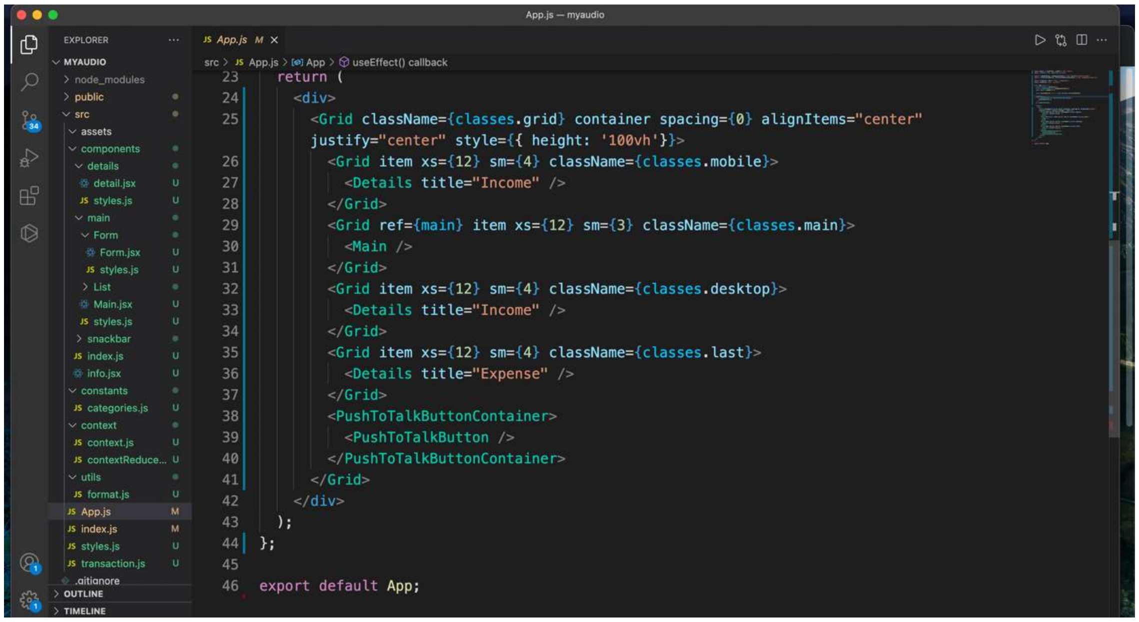
```
export default (date) => {
    const d = new Date(date);
    let month = `${d.getMonth() + 1}`;
    let day = `${d.getDate()}`;
    const year = d.getFullYear();

    if (month.length < 2) { month = `0${month}`; }
    if (day.length < 2) { day = `0${day}`; }

    return [year, month, day].join('-');
};
```

6. **<u>App Component-</u>**

This component is the component that includes different speech related java script libraries for providing the audio sensing feature that provides a final touch to the application. There are different libraries like speech recognition library, uuid library, pushtotalkbutton library etc. These libraries work collectively and provide a voice recognition feature int the application that provide an awesome feature of adding the data using human voice. The user need to say “add income to category any of amount any on date any”. Here any means the input that user wants to give.



```
import React, { useEffect, useRef } from 'react';
import { Grid } from '@material-ui/core';

import { SpeechState, useSpeechContext } from "@speechly/react-client";
import { PushToTalkButton, PushToTalkButtonContainer } from '@speechly/react-ui';

import { Details, Main } from './components';
import useStyles from './styles';

const App = () => {
  const classes = useStyles();
  const { speechState } = useSpeechContext();
  const main = useRef(null)

  const executeScroll = () => main.current.scrollIntoView()

  useEffect(() => {
    if (speechState === SpeechState.Recording) {
      executeScroll();
    }
  }, [speechState]);

  return (
    <div>
      <Grid className={classes.grid} container spacing={0} alignItems="center"
      justify="center" style={{ height: '100vh'}}>
```

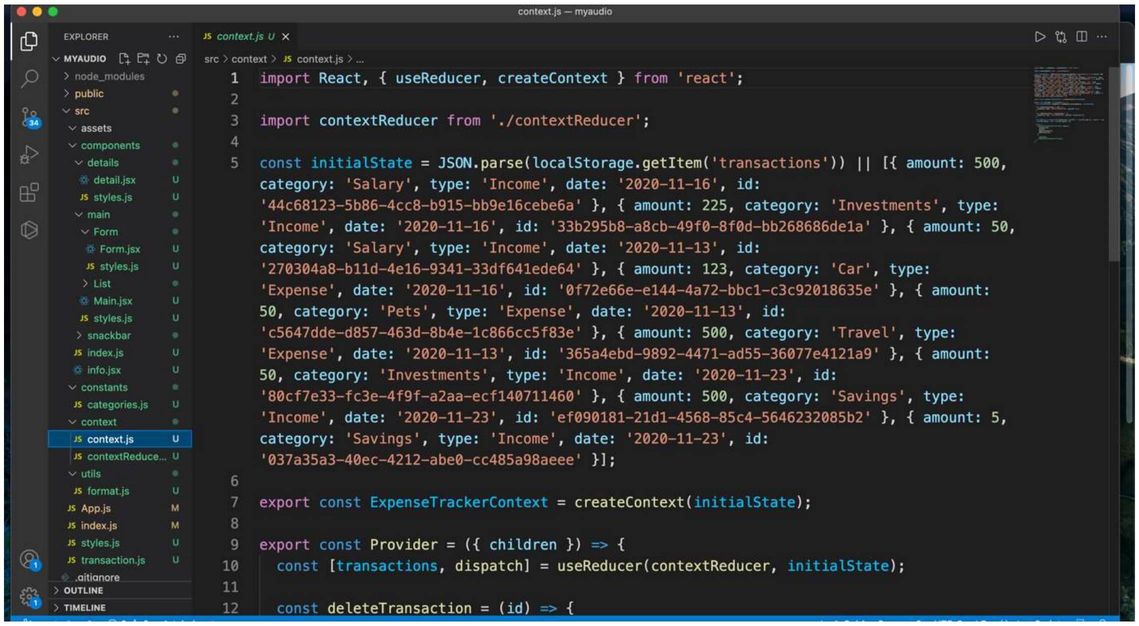


```
  return (
    <div>
      <Grid className={classes.grid} container spacing={0} alignItems="center"
      justify="center" style={{ height: '100vh'}}>
        <Grid item xs={12} sm={4} className={classes.mobile}>
          <Details title="Income" />
        </Grid>
        <Grid ref={main} item xs={12} sm={3} className={classes.main}>
          <Main />
        </Grid>
        <Grid item xs={12} sm={4} className={classes.desktop}>
          <Details title="Income" />
        </Grid>
        <Grid item xs={12} sm={4} className={classes.last}>
          <Details title="Expense" />
        </Grid>
        <PushToTalkButtonContainer>
          <PushToTalkButton />
        </PushToTalkButtonContainer>
      </Grid>
    </div>
  );
};

export default App;
```

7. Index Component-

This is also a parent component that renders the entire application and is the main component under which all other components are render and entirely complete user interface is rendered on this component as well.

Therefore the app is totally dependent on this component.

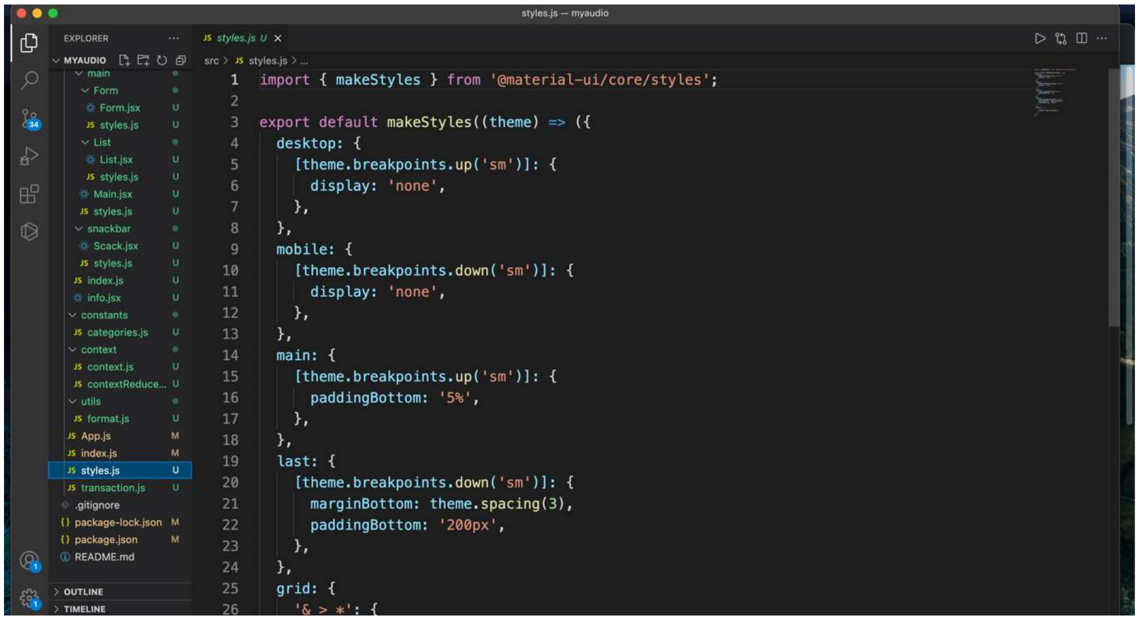


```
import React, { useReducer, createContext } from 'react';

import contextReducer from './contextReducer';

const initialState = JSON.parse(localStorage.getItem('transactions')) || [{ amount: 500,
category: 'Salary', type: 'Income', date: '2020-11-16', id:
'44c68123-5b86-4cc8-b915-bb9e16cebe6a' }, { amount: 225, category: 'Investments', type:
'Income', date: '2020-11-16', id: '33b295b8-a8cb-49f0-8f0d-bb268686de1a' }, { amount: 50,
category: 'Salary', type: 'Income', date: '2020-11-13', id:
'270304a8-b11d-4e16-9341-33df641ede64' }, { amount: 123, category: 'Car', type:
'Expense', date: '2020-11-16', id: '0f72e66e-e144-4a72-bbc1-c3c92018635e' }, { amount:
50, category: 'Pets', type: 'Expense', date: '2020-11-13', id:
'c5647dde-d857-463d-8b4e-1c866cc5f83e' }, { amount: 500, category: 'Travel', type:
'Expense', date: '2020-11-13', id: '365a4ebd-9892-4471-ad55-36077e4121a9' }, { amount:
50, category: 'Investments', type: 'Income', date: '2020-11-23', id:
'80cf7e33-fc3e-4f9f-a2aa-ecf140711460' }, { amount: 500, category: 'Savings', type:
'Income', date: '2020-11-23', id: 'ef090181-21d1-4568-85c4-5646232085b2' }, { amount: 5,
category: 'Savings', type: 'Income', date: '2020-11-23', id:
'037a35a3-40ec-4212-abe0-cc485a98aeee' }];

export const ExpenseTrackerContext = createContext(initialState);

export const Provider = ({ children }) => {
  const [transactions, dispatch] = useReducer(contextReducer, initialState);

  const deleteTransaction = (id) => {
```

8.Style Component-

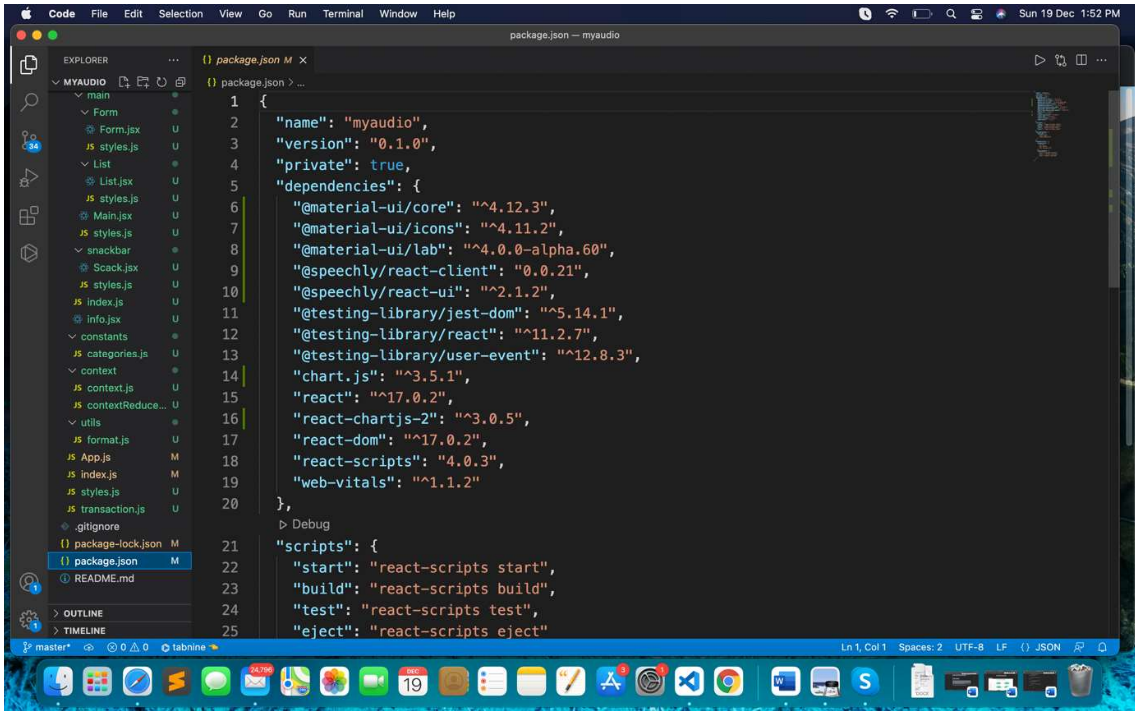
This component is the main styling component that provide a beauty to the website for better user interface and interactiv interface for the users.

It provide colors to website and make website more eye catchy and hence attracts users as well.

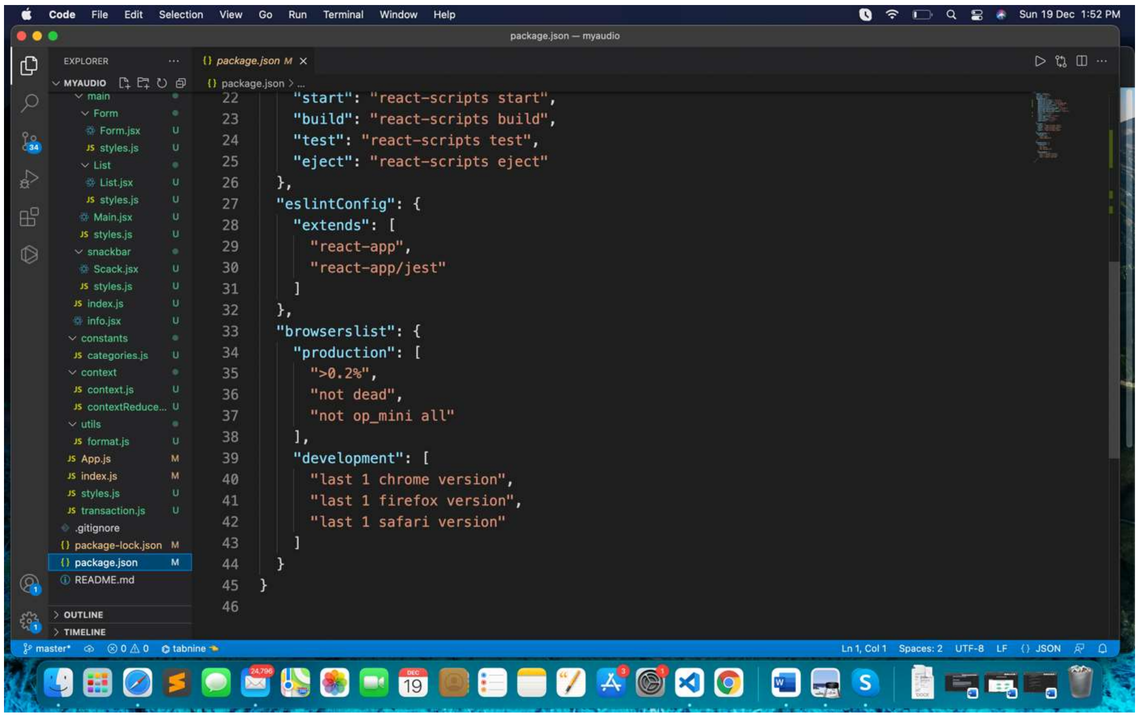


```js
import { makeStyles } from '@material-ui/core/styles';

export default makeStyles((theme) => ({
  desktop: {
    [theme.breakpoints.up('sm')]: {
      display: 'none',
    },
  },
  mobile: {
    [theme.breakpoints.down('sm')]: {
      display: 'none',
    },
  },
  main: {
    [theme.breakpoints.up('sm')]: {
      paddingBottom: '5%',
    },
  },
  last: {
    [theme.breakpoints.down('sm')]: {
      marginBottom: theme.spacing(3),
      paddingBottom: '200px',
    },
  },
  grid: {
    '& > *': {
```

9. **<u>Package.json-</u>** This file is a primary file for containing all the information of the libraries that we have used in this application with the name and version number we are using that allows the backend developer a knowledge of what technology he used in the application. Hence developer can update to new technology when required.



```json
{
  "name": "myaudio",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@material-ui/core": "^4.12.3",
    "@material-ui/icons": "^4.11.2",
    "@material-ui/lab": "^4.0.0-alpha.60",
    "@speechly/react-client": "0.0.21",
    "@speechly/react-ui": "^2.1.2",
    "@testing-library/jest-dom": "^5.14.1",
    "@testing-library/react": "^11.2.7",
    "@testing-library/user-event": "^12.8.3",
    "chart.js": "^3.5.1",
    "react": "^17.0.2",
    "react-chartjs-2": "^3.0.5",
    "react-dom": "^17.0.2",
    "react-scripts": "4.0.3",
    "web-vitals": "^1.1.2"
  },
  ▷ Debug
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
```



```
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
  "eslintConfig": {
    "extends": [
      "react-app",
      "react-app/jest"
    ]
  },
  "browserslist": {
    "production": [
      ">0.2%",
      "not dead",
      "not op_mini all"
    ],
    "development": [
      "last 1 chrome version",
      "last 1 firefox version",
      "last 1 safari version"
    ]
  }
}
```

5.2 USER INTERFACE

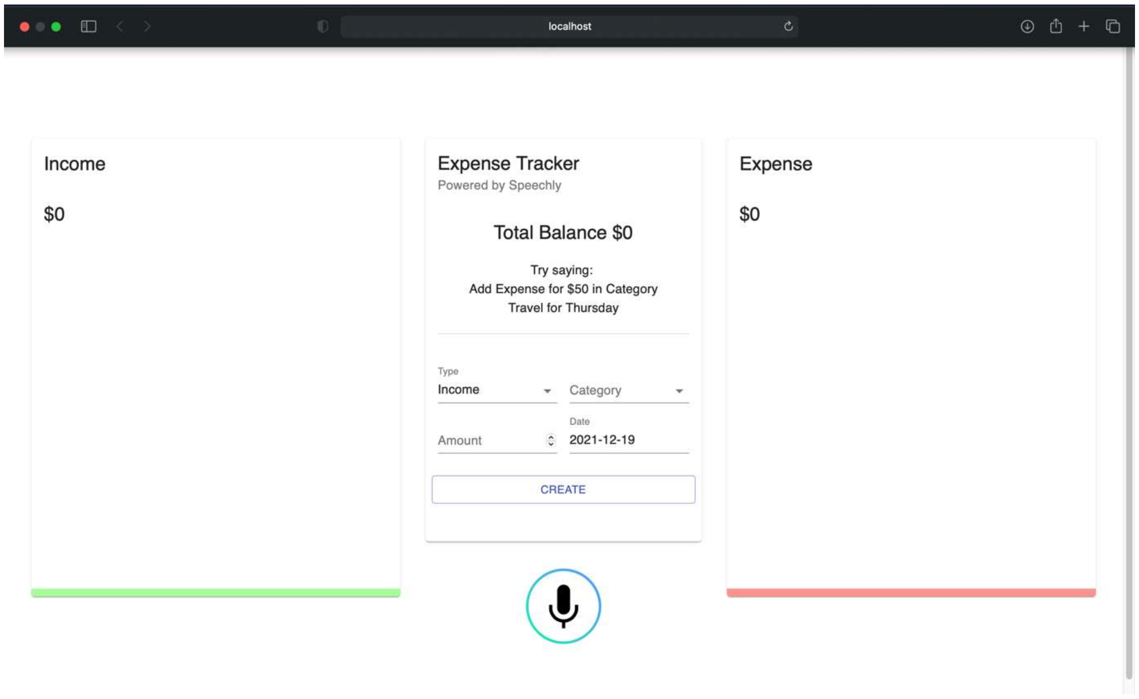
Now coming on to the user interface on how the interface look like is defined below and the instruction that how a user can add the data and select the category and date as per the choice and can interact with the application in an effective way.

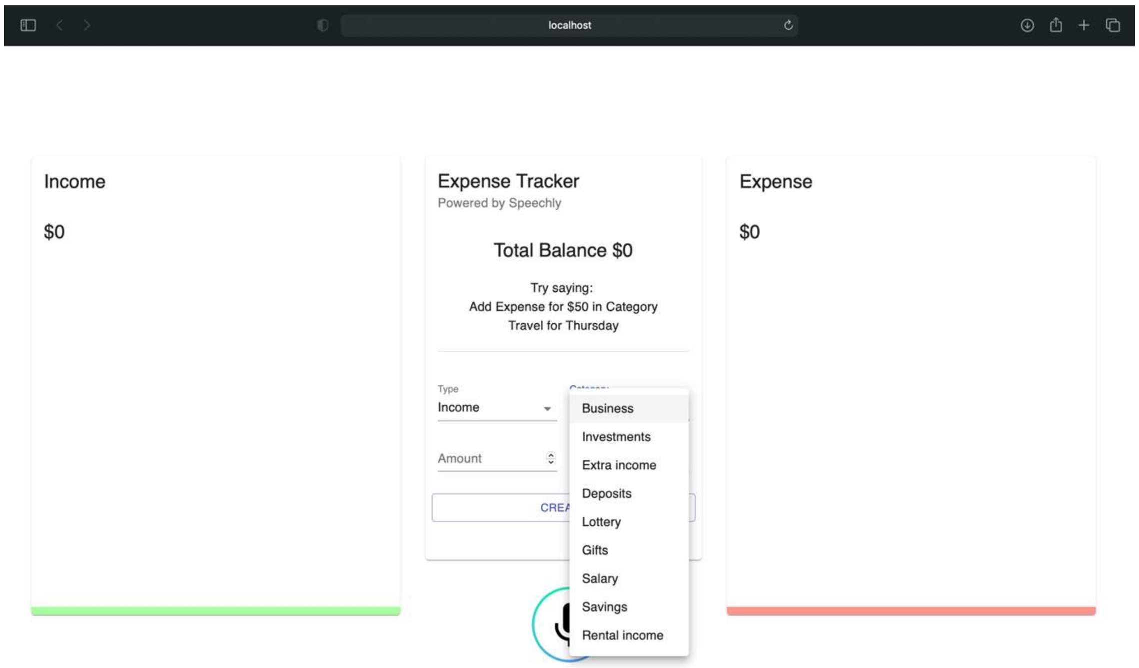
Moreover this application is a very user friendly website and is very easy for the user to work with. Also the website is self-learning website and user do not get any difficulty in working with this application.

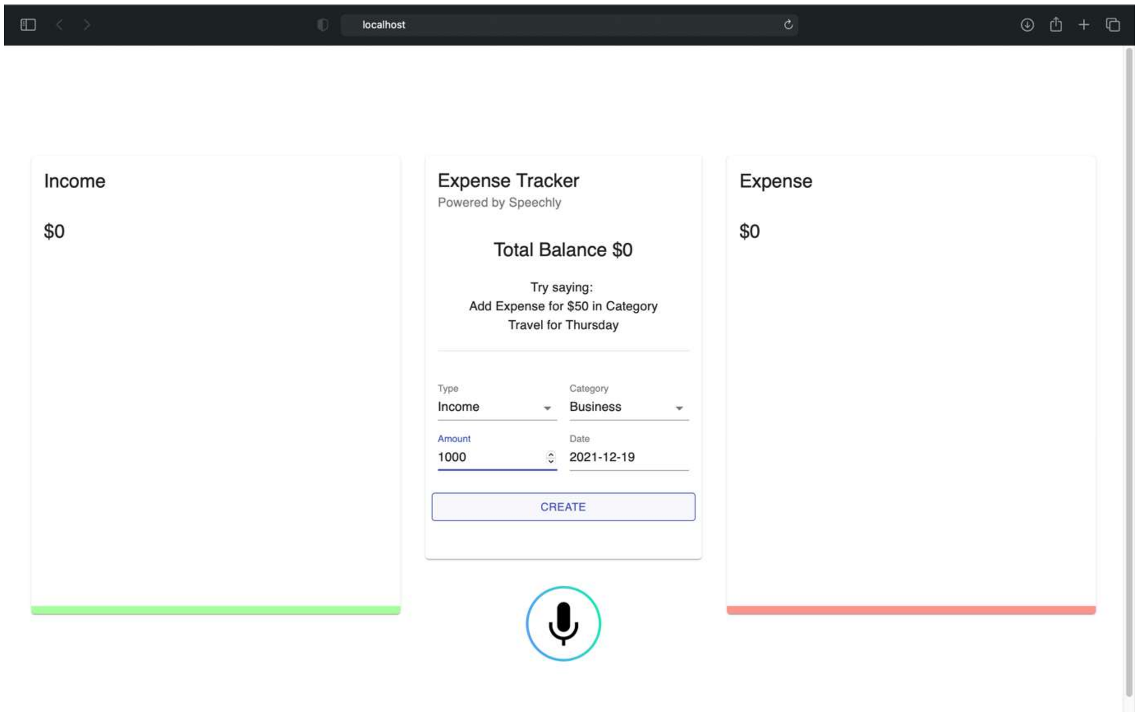
5.2.1 MANUAL ADDITION

The first method to add the record is by using the manual addition method in which user manually adds the details for the record i.e. the category of record, the sub category for the selected category, amount and the date then pressing the create button will create the record for the user.

Below you will find the photos representing the steps to add the data manually.






localhost

Income

$0

Expense Tracker
Powered by Speechly

Total Balance $0

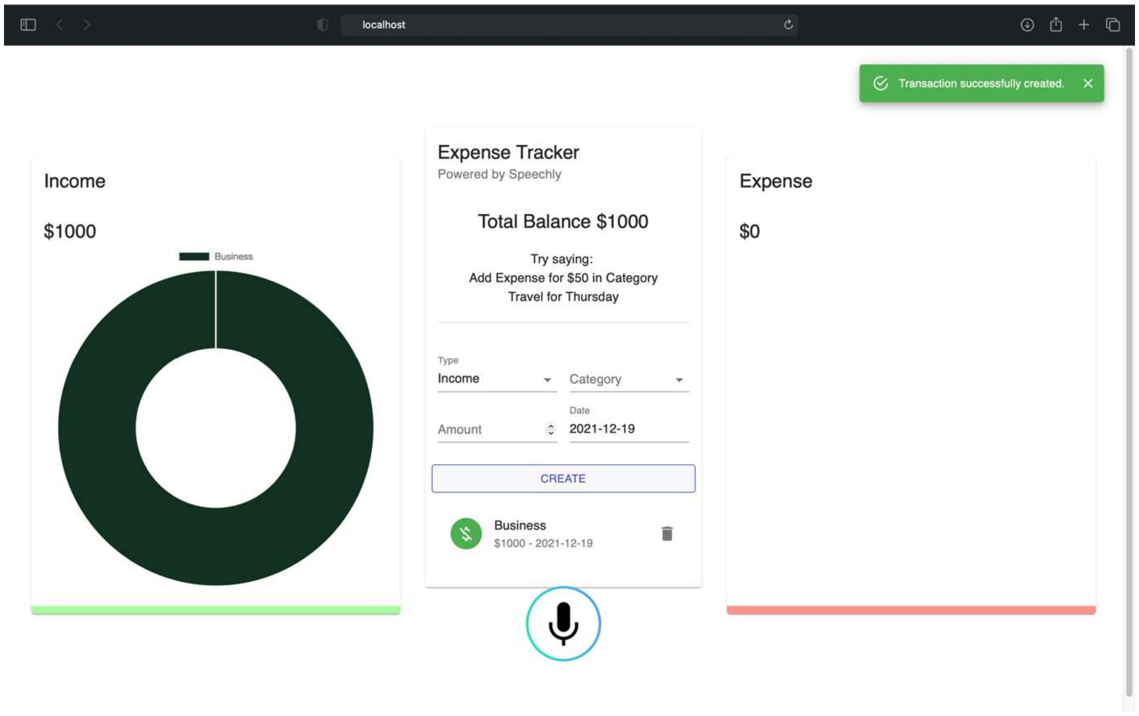
Try saying:
Add Expense for $50 in Category
Travel for Thursday

Type: Income
Category: Business
Amount: 1000
Date: 2021-12-19

CREATE

Expense

$0


localhost

Transaction successfully created.

Income

$1000

Expense Tracker
Powered by Speechly

Total Balance $1000

Try saying:
Add Expense for $50 in Category
Travel for Thursday

Type: Income
Category
Amount
Date: 2021-12-19

CREATE

Business
$1000 - 2021-12-19

Expense

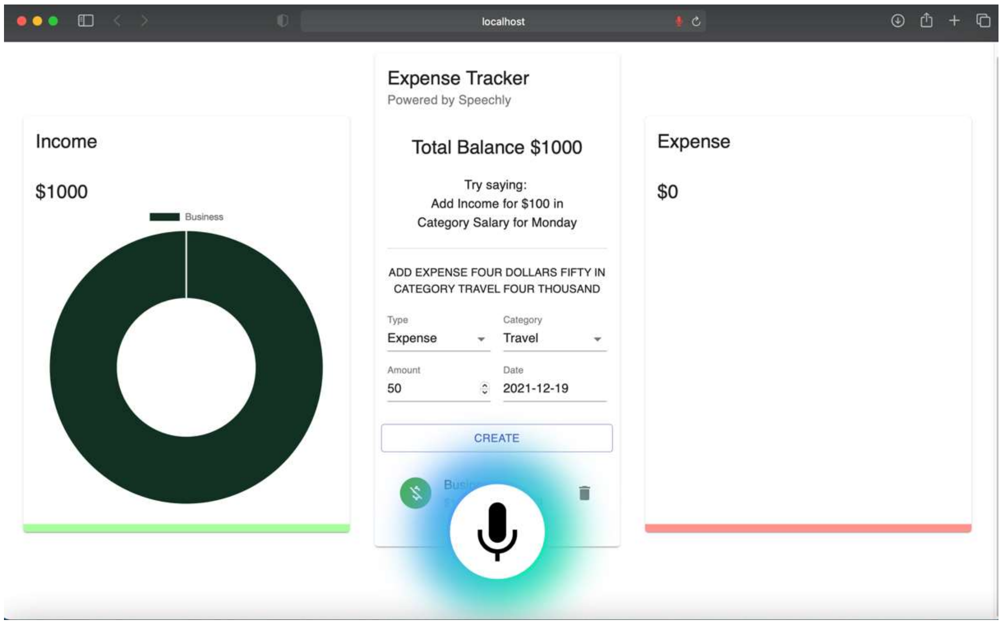
$0

5.2.2 AUDIO TRACKER

Now coming on to the advance feature of the application is adding data using audio command. The application provide a newly feature to add the data using a voice command as shown in below pictures.

User need to provide a specific command and the data will be added into the records.

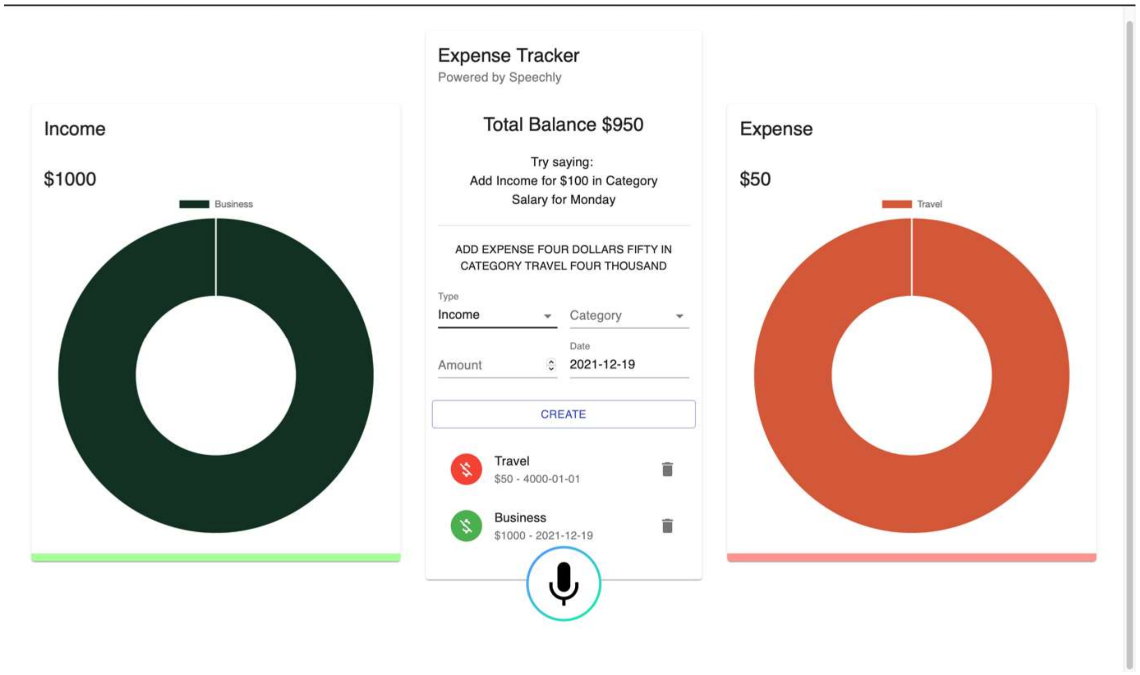
The format for the command is as “Add expense for $100 in category bills on Monday” and the resulting data will be added into the record.



Try saying:
Add Income for $100 in Category Salary for Monday

ADD EXPENSE FOUR DOLLARS FIFTY IN CATEGORY TRAVEL FOUR THOUSAND

Type: Expense
Category: Travel
Amount: 50
Date: 2021-12-19



Expense Tracker
Powered by Speechly

Total Balance $950

Try saying:
Add Income for $100 in Category Salary for Monday

ADD EXPENSE FOUR DOLLARS FIFTY IN CATEGORY TRAVEL FOUR THOUSAND

Type: Income
Category
Amount
Date: 2021-12-19

CREATE

Travel
$50 - 4000-01-01

Business
$1000 - 2021-12-19

Income
$1000

Expense
$50

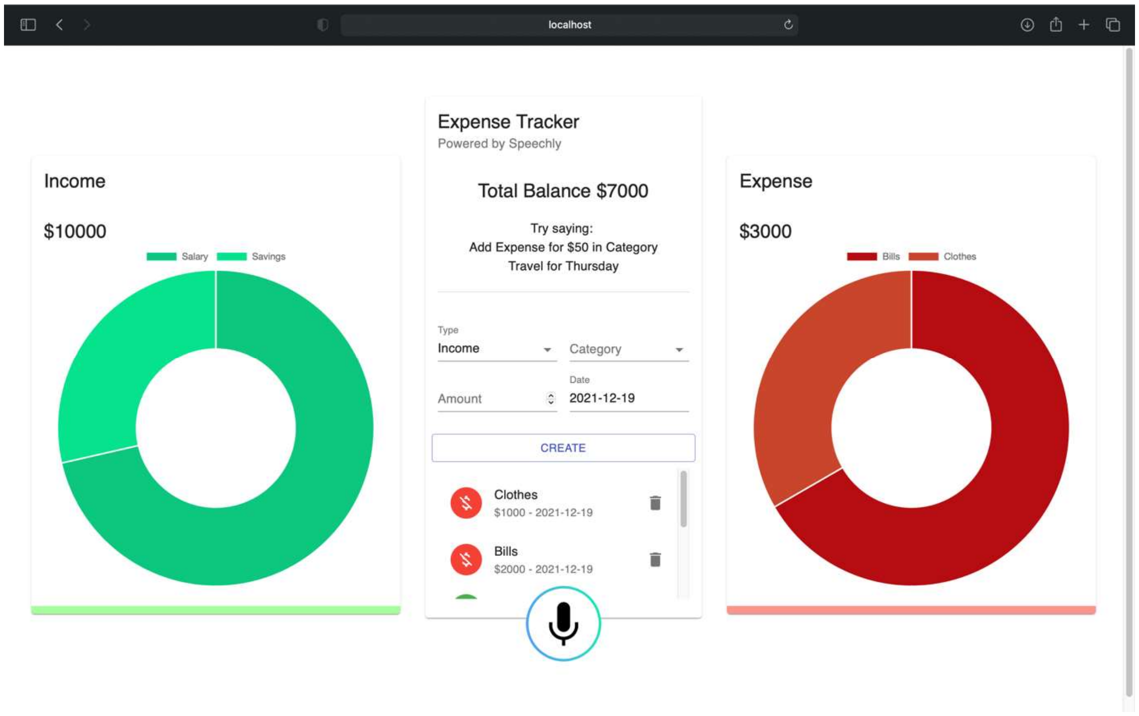
RESULTS

Based on the prior working and execution of the application on the website it was noted that it provide accurate result as per the convenance of the user. The application is perfect in working and the user interface is very favourable to the user operating the application.

User can easily add and remove the data as per the choice they want. Moreover the website is auto explanatory to the user and user do not find any difficulty to work with the application. First whenever the site loads the user is provided with the interface showing the inputs for the amount, category, date and the category of the income or expense. User then have to select the category. After that user then select to add the data by clicking on the create button then after that instance the data is added.

There's another alternative way to add the data is that using the audio command user can say the command and can add the data to the list.

Based on the addition of the data and the removal of the data a pie chart representation of that particular data is created according. There are 3 section in this application, the left and right section of the application is consisting of the pie chart representation of the data added to the category of income that is represented by the green colour chart and the rightmost section consist of the pie chart representation of the data added to the category of the expense added that is represented by the red colour chart. Moreover, coming on to the middle part of the application then it consist of the main input area where the user can add the data based on the requirement mentioned in the above examples and photos.


localhost

Income

$10000

Salary Savings

Expense Tracker

Powered by Speechly

Total Balance $7000

Try saying:
Add Expense for $50 in Category
Travel for Thursday

Type: Income

Category

Amount

Date: 2021-12-19

CREATE

Clothes
$1000 - 2021-12-19

Bills
$2000 - 2021-12-19

Expense

$3000

Bills Clothes

The above diagrams depicts the working of the application where user has added the 10000 as the income showing on the left side and the expense user has done is represented on the right side.

FUTURE SCOPE

It is important to note that there is no specific application that can help you to track the amount of expense that are done as per daily basis. Since every problem has a different solution and therefore this is a solution for that problem. Now a days it's very difficult to keep a track on the daily expenses and income so in order therefore this is perfectly designed for that scenario.

As discussed above we have briefly described how that application is working and how the user can interact with the application efficiently.

The above described model is the proposed model we have launched. Moreover, we would launch another advance version of our model that would promote the future development of Expense Tracking application. A new trend will be introduced in future that would have a great impact on users and attract many users to use the application at a larger scale. In context to new trend, we will be adding an **'Voice Generated Deletion of Record** to our system. It means after as like we are adding the record in the application, we would get another advantage of deleting the record by saying the command and interacting with the microphone.

Moreover the future scope of the application is also specified to develop the android version of the expense tracking application so that user can use the application portably.

CONCLUSION

In this project, we have proposed a react java script based Voice Operated Expense Tracker. We have discussed our proposed model using React java script. The performance evaluation of the proposed Real time expense tracker is carried out in terms of validation accuracy. We analysed our proposed model using the real users and evaluate their performance. Result of the experiment shows that the model proposed is better in terms of user interaction and expense calculation. Moreover, user find it quite easy to work and also benefited with the features. We have also analysed that the real time expense tracker due to its voice operated feature is not only favourable to the general user but also helpful for handicap people to make the most of the application and allow then to operate the application as well.

The real time expense tracking system not only reduces human workload but also saves time, space for easy tracking of monthly/yearly expense and work accordingly. Specifically, in today's world, close international relations and globalization all play a key role in promoting the ever-changing nature of real time tracking application. In addition, it also has some practical value and the user is also provided with reasonable and good advices to help the people on how much the need to save in order to have a wealthy living in this firmament.

“The Real-time voice operated expense and transaction” tracking application is a web application that acts as a wallet to the people and tracks all the transaction and expenses that a person is doing in his/her daily life. The main objective behind the introduction of this particular application is to allow user to keep an eye on the expense he/she is doing in the daily life in order to minimize the wastage of money and increase in the saving of money so that it can be used in future work. This application is a voice command activated application that contributes in providing user interactive UI allowing the people to easily add and remove the money as per the required date.

Moving forward for the future development the future model of the Real time Expense tracker is also going to be released and the future model would be android application corresponding to the current working model and also consist of different feature. The introduction to this model will surely help most of the people to calculate their expense and income and also allow the user to manage the user issues related to the savings.

REFERENCE

1. Live Code Stream, Vol. 3, No.8, https://livecodestream.dev/post/how-to-control-your-react-app /.
2. Ngai, E.W.T. & A. Gunasekaran (2007). A review for mobile commerce research and applications. Decision Support Systems, vol. 43, no. 1, pp. 3-15.
3. Cyr, D., M. Head & A. Ivanov (2006). Design aesthetics leading to m-loyalty in mobile commerce. Information & Management, vol. 43, no. 8, pp. 950-963.
4. [Sumitra, Ushio & Jun Yoshii (2010). Enhancement of ecommerce via mobile accesses to the Internet. Electronic Commerce Research and Applications, vol. 9, pp. 217-227. 5. PocketHacks.com (2011). iOS page, All about Windows Mobile, Windows Phone 7 and Android devices, latest news, freeware apps and hacks. Retrieved August 20, 2011 from http://pockethacks.com/ios/.
5. https://www.javatpoint.com/. React JavaScript, HTML,CSS
6. https://www.udemy.com/, Udemy(Platform for learning)
7. **https://stackoverflow.com/**
8. **https://www.geeksforgeeks.org/**
9. **https://www.wikipedia.org/**