**Redux:**

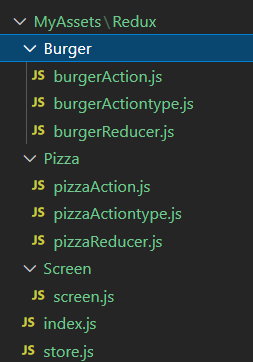
**How to use Redux with ReactNative?**

* First, we create a fresh React nativeProject.
* Go to the project folder
* **“npm install redux”**which is an official redux dependency.
* **“npm install react-redux”** which is used to connect redux to react.

# You people please refer the link

<https://www.geeksforgeeks.org/how-to-use-redux-with-reactnative/>

**My Folder Structure:**

****

**Code:**

**App.js**

**App.js:**In App.js we import **Provider**component from **‘react-redux’** and **store** from **store.js**. **Provider** is used to pass store state to all its child components.

import { StatusBar } from 'expo-status-bar';

import React from 'react';

import { StyleSheet, Text, View } from 'react-native';

import {Provider} from 'react-redux'

import Screen from './MyAssets/Redux/Screen/screen';

import store from './MyAssets/Redux/store'

export default function App() {

  return (

    <Provider store={store}>

    <Screen/>

    </Provider>

  );

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

    backgroundColor: '#fff',

    alignItems: 'center',

    justifyContent: 'center',

  },

});

**burgerActiontype.js**

**burgerActionTypes.js:**In this file, we export all the string action\_type property.

export const INCREASE\_BURGER='INCREASE\_BURGER';

export const DECREASE\_BURGER='DeCREASE\_BURGER';

**burgerAction.js**

In this, we will create our **action** function which returns action\_type and optional payload property to reducer.

import {INCREASE\_BURGER,DECREASE\_BURGER} from './burgerActiontype'

export const increaseBurgerAction=(parameter)=>{

    return{

        type:INCREASE\_BURGER,

        payload:parameter

    }

}

export const decreaseBurgerAction=()=>{

    return{

        type:DECREASE\_BURGER,

    }

}

**burgerReducer.js**

**burgerReducer.js.**In this file, we will create a **burgerReducer () function** which takes an initial state and action as a parameter and returns a new state of the store based on the **action\_type**.

import {INCREASE\_BURGER,DECREASE\_BURGER} from './burgerActiontype'

const initialState={

    numberOfBurger:0

}

const burgerReducer=(state=initialState,action)=>{

    switch (action.type) {

        case INCREASE\_BURGER:

            return{

                ...state,

                numberOfBurger:state.numberOfBurger+action.payload

            }

         case DECREASE\_BURGER:

            return{

                ...state,

                numberOfBurger:state.numberOfBurger-1

             }

        default:return state;

    }

}

export default burgerReducer;

**pizzaActiontype.js**

export const PIZZA\_INCREASE='PIAZZA\_INCREASE';

export const PIZZA\_DECREASE='PIZZA\_DECREASE';

**pizzaAction.js**

import {PIZZA\_INCREASE,PIZZA\_DECREASE} from './pizzaActiontype'

export const increasePizzaAction=(parameter)=>{

    return{

        type: PIZZA\_INCREASE,

        payload: parameter

    }

}

export const decreasePizzaAction=()=>{

    return{

        type: PIZZA\_DECREASE

    }

}

**pizzaReducer.js**

import {PIZZA\_INCREASE,PIZZA\_DECREASE} from './pizzaActiontype'

const initialState ={

    numberOfPizza:0

}

const pizzaReducer=(state=initialState,action)=>{

    switch (action.type) {

        case PIZZA\_INCREASE:

            return{

                ...state,

                numberOfPizza: state.numberOfPizza+action.payload

            }

        case PIZZA\_DECREASE:

            return{

                ...state,

                numberOfPizza:state.numberOfPizza-1

            }

        default: return state;

    }

}

export default pizzaReducer;

**index.js**

**Index.js:**This file will be used to export all **actions** from a single file. This file is completely optional to create and you can import action from their respective JavaScript files also.

// Single file for exporting all actions

export {increaseBurgerAction} from './Burger/burgerAction';

export {decreaseBurgerAction} from './Burger/burgerAction';

export {increasePizzaAction} from './Pizza/pizzaAction';

export {decreasePizzaAction} from './Pizza/pizzaAction';

**store.js**

**store.js:**In this file, we will import all the **reducers** and then create a **store** and export it to App.js.We can also create a store in App.js

import {createStore,combineReducers} from 'redux';

s

import burgerReducer from './Burger/burgerReducer';

import pizzaReducer from './Pizza/pizzaReducer'

const rootReducer=combineReducers({

    burgerReducer:burgerReducer,

    pizzaReducer:pizzaReducer

})

const store=createStore(rootReducer);

export default store;

**screen.js**

**Screens** which will contain all of our app screen components.  Inside **Screens** directory we will create a file named **screen.js.**

import React, { Component } from 'react';

import { Button, Text, View } from 'react-native';

import {connect} from 'react-redux';

import {increaseBurgerAction,decreaseBurgerAction,increasePizzaAction,decreasePizzaAction} from '../index'

class Screen extends Component {

    render() {

        return (

            <View>

                <View>

                <Text>Number Of Burger:{this.props.numberOfBurger}</Text>

                <Button

                title="Increase"

                onPress={()=>this.props.increaseBurger(1)}

                />

                <Button

                title="Decrease"

                onPress={()=>this.props.decreaseBurger()}

                />

            </View>

            <View>

                <Text>Number Of Pizza={this.props.numberOfPizza}</Text>

                <Button

                title="Increase"

                onPress={()=>this.props.increasePizza(1)}

                />

                <Button

                title="Decrease"

                onPress={()=>this.props.decreasePizza()}

                />

            </View>

            </View>

        );

    }

}

const mapStateToProps = (state)=>{

    return{

        // numberOfBurger:state.numberOfBurger,

        numberOfBurger:state.burgerReducer.numberOfBurger,

        numberOfPizza:state.pizzaReducer.numberOfPizza

    }

}

const mapDispatchToProps = (dispatch)=>{

    return{

        increaseBurger:(parameter)=>{dispatch(increaseBurgerAction(parameter))},

        decreaseBurger:()=>{dispatch(decreaseBurgerAction())},

        increasePizza:(parameter)=>{dispatch(increasePizzaAction(parameter))},

        decreasePizza:()=>{dispatch(decreasePizzaAction())}

    }

}

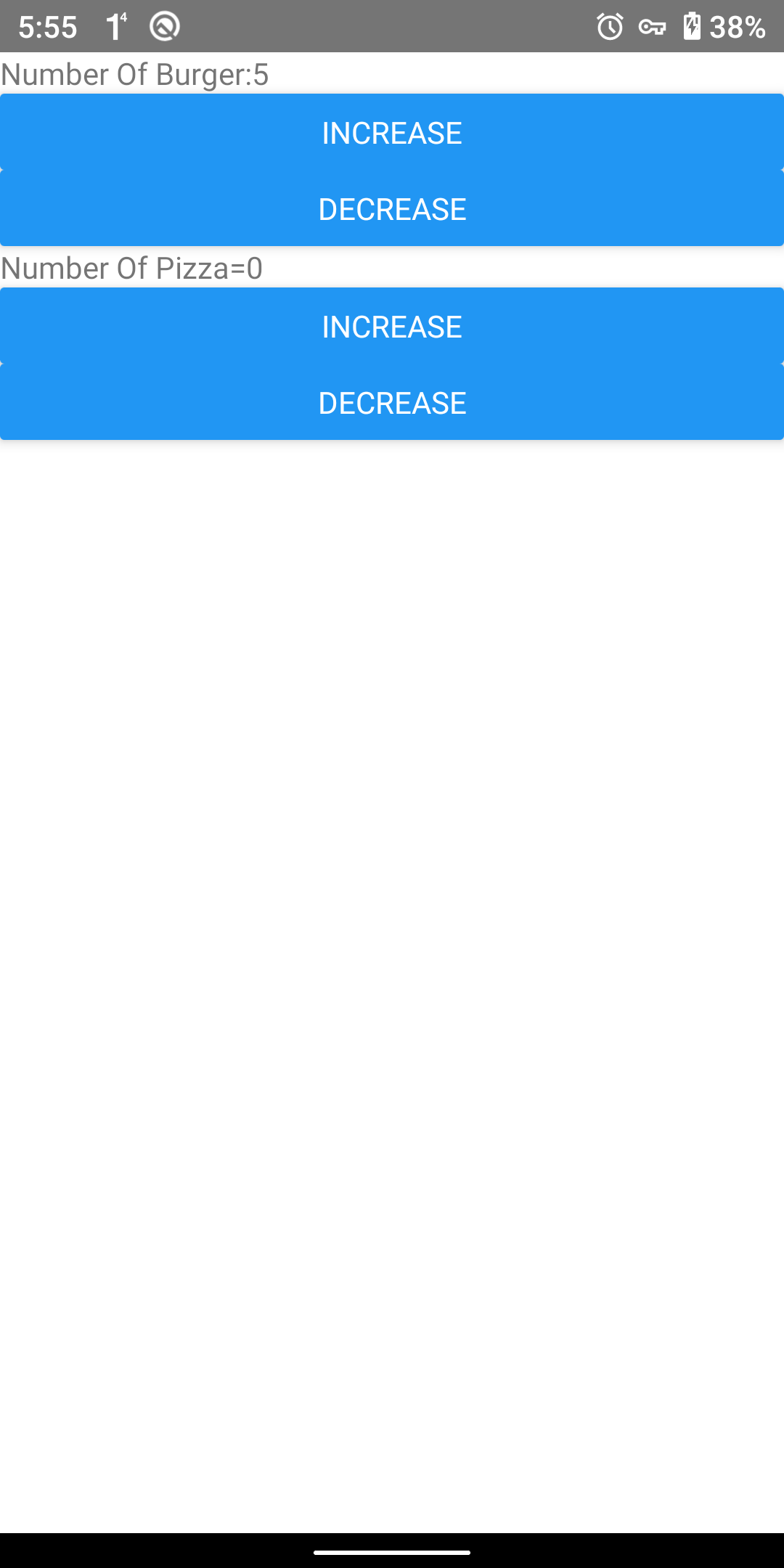
export default connect(mapStateToProps, mapDispatchToProps)(Screen) ;

* **Render:** This is called when the state of the store changes.
* **mapStateToProps:**This function maps the store state to the screen component to be used as props. We can also rename this other than mapStateToProps.
* **mapDispatchToProps:**This function maps the actions to the screen component to be called using props. We can also rename this other than mapDispatchToProps.
* **connect:**This is a **‘react-redux’** inbuilt function used to connect screen component to mapStateToProps and mapDispatchToProps.Always pass mapStateToProps as a first parameter and mapDispatchToProps as the second parameter to connect () function otherwise it will generate an error.

**Multiple Reducers:**In most cases, we have to use multiple reducers in order to separate states and actions. To demonstrate this, I have created another Directory named **Pizza** which contains code for **pizzaReducer.js, pizzaActionsType.js**,and**pizzaActions.js.**

**store.js:**In this, we use **combineReducers()** which is an inbuilt function of **‘redux’** to combine our reducers.

**OUTPUT:**

****

**FETCH a Data from API:**

**App.js:**

import React, { Component } from 'react';

import Fetching1 from './Components/Fetching1';

export default class App extends Component {

  render() {

    return (

    <Fetching1></Fetching1>

    );

  }

}

**Fetching1.js**

import React, { Component } from 'react';

import { FlatList, Image, ScrollView, Text, View } from 'react-native';

import { Card } from 'react-native-paper';

import Icon from 'react-native-vector-icons/AntDesign'

import Ico from 'react-native-vector-icons/Feather'

class Fetching1 extends Component {

   constructor(props) {

       super(props);

       this.state = {

           isLoading: true,

           list:[]

       }

   }

    componentDidMount() {

        fetch("https://eatoo.uberdoo.com/foodapp/public/api/guest/listRestaurant",{

            method:'POST',

            headers: {

                Accept: 'application/json',

                'Content-Type': 'application/json'

              },

                body:JSON.stringify({

                latitude:13.0358481,

                longitude:80.244455

            })

        })

    .then((res)=>res.json())

    .then((res)=>

    this.setState({

        isLoading:false,

        list: res.listRestaurants

    }))

    }

    render() {

        const {list,isLoading}=this.state

        return (

            <View >

            <View>

{/\*

                <Card style={{margin:8,height:20}}> \*/}

                    <View style={{flexDirection: 'row',margin:22}}>

                        <View >

                    <Text style={{flex:1}}>231RESTAURANT      </Text>

                    </View>

                    <View>

                    <Text style={{flex:1}}>  RELEVANCE <Icon name="down" color='orange'></Icon></Text></View>

                    <Text style={{flex:1,fontweight:'bold'}}>      FILTER  <Ico name="sliders" color='orange'></Ico></Text>

                    </View>

                {/\* </Card> \*/}

            </View>

           <FlatList

           data={list}

           renderItem={({item})=>(

               <ScrollView>

            {/\* <Card style={{flex:1,margin:10,height:100}}>

                       <Card.Title title={item.restaurantName}  >

                           <Card.Content>

                               <Text></Text>

                           </Card.Content>

                       </Card.Title>

               </Card>        \*/}

               <Card  style={{margin:10,height:150}}>

                   <View style={{flexDirection:'row'}}>

                   <Card style={{flex:1,height:150,padding:5}}>

                       <Image source={{uri:item.restaurantImage}} style={{ height:"100%",width:"100%",alignSelf:'center',borderRadius:8}} ></Image>

                   </Card>

                   <Card style={{flex:2,height:150}}>

                   <Card.Title title={item.restaurantName}  subtitle="North Indian,South Indian,Chinee..."></Card.Title>

                     <Card.Content>

                         <View style={{flexDirection:'row'}}>

                             <Text style={{flex:1}}>Time</Text>

                             <Text style={{flex:1}}>Price</Text>

                             <Text style={{flex:1,alignSelf:"flex-end"}}>   Rating</Text>

                         </View>

                         <View style={{flexDirection:'row',padding:4}}>

                         <Text style={{flex:1,fontWeight:'bold',marginLeft:-4}}>{item.time} mms</Text>

                             <Text style={{flex:1,fontWeight:'bold'}}>{item.displayCostForTwo}</Text>

                             <View style={{backgroundColor:'#5AE595',flex:1,height:20,width:0,borderRadius:3,fontWeight:'bold',}}>

                             <Text style={{color:'white',fontWeight:'bold',alignContent:'center'}}>   <Icon size={15} name="star" color='white'></Icon>  {item.averageReview}</Text>

                             </View>

                         </View>

                         </Card.Content>

                   </Card>

                   </View>

               </Card>

               </ScrollView>

           )}

           ></FlatList>

        </View>

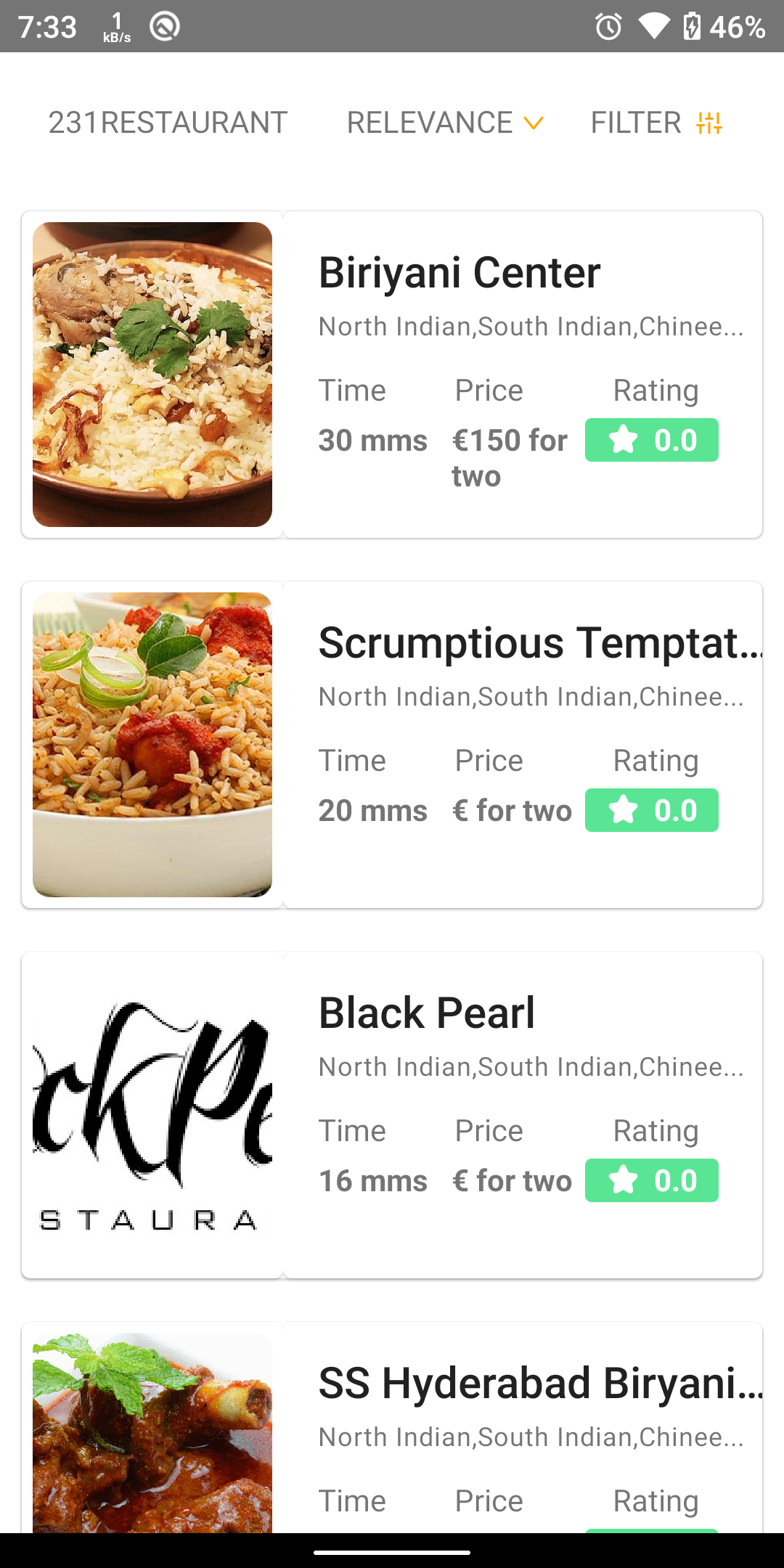
        );

    }

}

export default Fetching1;

**OUTPUT:**

****