

GST RATE FINDER

REPORT OF MAJOR PROJECT SUBMITTED FOR PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF

MASTER IN COMPUTER APPLICATION

BY

MAHUA BOSE(11701015024)

MADHURIMA SARKAR(11701015023)

SREETAMA ROY(11701015045)

UNDER THE SUPERVISION OF
MR.JAYANTA DATTA
Asst.Prof.RCCIIT



At

RCC INSTITUTE OF INFORMATION TECHNOLOGY

Affiliated to Maulana Abul Kalam Azad University of Technology

CANAL SOUTH ROAD,BELIAGHATA,KOLKATA-700015

RCC INSTITUTE OF INFORMATION TECHNOLOGY

KOLKATA-700015,INDIA



CERTIFICATE

We hereby certify that the work which is being presented in the project report entitled **“GST RATE FINDER”** in partial fulfillment of the requirements for the degree of **MASTER OF COMPUTER APPLICATION** and submitted to the Department of **MCA** of **RCC INSTITUTE OF INFORMATION TECHNOLOGY,KOLKATA,WB** is an authentic record of our own work carried out during the period from January 2018 to May 2018 under the supervision of **Prof. JAYANTA DATTA,Asst. Prof. RCCIIT.**

Name of the candidates

1.MAHUA BOSE

2.MADHURIMA SARKAR

3.SREETAMA ROY

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

JAYANTA DATTA

Dept of MCA,RCCIIT KOLKATA

Head,Dept of MCA,RCCIIT,KOLKATA

RCC INSTITUTE OF INFORMATION TECHNOLOGY

KOLKATA-700015,INDIA



CERTIFICATE OF ACCEPTANCE

The report of the project title GST RATE FINDER submitted by MAHUA BOSE(ROLLNO:MCA2015/023),MADHURIMA SARKAR(MCA2015/029),SREETAMA ROY(MCA2015/032)of MCA 6th semester of 2015 is here by recommended to be accepted for the partial fulfillment of the requirements for MCA degree in Maulana Abul Kalam Azad University Of Technology.

Signature of the External Examiner(s)

ACKNOWLEDGEMENT

I express my sincere gratitude to Mr.JAYANTA DATTA of Department of COMPUTER APPLICATION,RCCIIT and for giving most valuable suggestion ,helpful guidance and encouragement in the execution of this project work.

I will like to give a special mention to my friends.Last but not the least I am grateful to all the faculty members of the institution for their support.

DATE _____

(signature of students)

CONTENTS

1. Introduction
2. Project overview
3. Purpose of the system
4. How does the GST RATE FINDER app work?
5. Problems in existing system
6. Features of proposed system
7. GST rate finder project modules
8. Software requirement specifications
9. Minimum Hardware and Software Requirements
10. User characteristics
11. Constraints of use
12. Procedure Discussion
 - 13.1 Entity Relationship Diagram
 - 13.2 Use Case Diagram(UML)
14. Project Source Code
15. Project Snapshot
16. Future Scope
17. Conclusion
18. Bibliography

1. ABSTRACT OF THE PROJECT

Goods and Service Tax(GST) is an indirect tax in India which replaced various taxes levied by the central and state governments. There are mainly two types of GST i.e for sale within the state revenue will be shared equally between the centre and the state i.e CGST & SGST.

1.1 Introduction to the system

People by now has become well versed with GST and it is known factor that based upon the new tax system which is GST, the rate of the goods are varied but yet the public doesn't have a clear idea on which goods and services actually the GST is showing its influence.

2.PROJECT OVERVIEW :

This documents the process of designing, building and testing this android based application to be used for marketing. The "GST RATE FINDER" is a mobile based application. This application provides facility for the Admin to add products, view all products and update the database. Also the STOREMANAGER can check the GST(CGST +SGST) rate of all products and whenever there is a sale a mail is sent to Government. The application works by using the internet connection. It helps the businessman to find accurate rates of various goods and services. The application has been made available for Android platform.

3. PURPOSE OF THE SYSTEM

GST rate finder app will give you the best results in case you are struggling to find the applicable tax rates for the various goods and services. This android project provides the customer product list with GST rates entered by the android phone user. This information helps the user to find the desired products to buy. Well it consists of entire details of GST rates with CGST+SGST and it provides the basic information about the product. This project is beneficial for the shopkeepers as well as customers.

This proposed application requires internet access and thus eliminates the disadvantage of single point failure. The system gives the basic details that will be required such as an image of that product with the price and current GST rate.

4. HOW DOES THE GST RATE FINDER APP WORK?

To find out the required tax rate:

1. Enter the name or chapter heading of the commodity or service in the search box provided
2. Search result will list all the goods and services containing the name which was typed in the search box
3. Scroll down the list of description
4. Click on any specific item on the list. A display window will pop-up, containing details such as the GST rate, description of goods or services.

5.PROBLEMS IN EXISTING SYSTEM:-

There are few limitations in existing android project which are following:

- In existing application only GST rates are provided. But in this application gst rates and central government GST rates are also provided.
- Existing application does not allow payment option. But in this application payment option is provided.

6.FEATURES OF THE PROPOSED SYSTEM:-

- Web based user registration
- Admin can update and insert products and GST rates.
- Store-manager can choose a product and the app will provide SGST & CGST and the total price of the product.
- Store-manager will also get a detailed information of that product.
- Faster retrieval of information
- All details will be available on click.
- This application is useful for the Government & the Store-manager.
- The application is self starter.

7.GST RATE FINDER PROJECT MODULES:-

Given are the modules for android GST rate finder project:

1. **User Module:** Via user module, user can make account into the application and can use it for future purpose.
2. **Booklet Module:** In this module, all the information is viewed via this module and user can access this module through their account.
3. **Database Module:** This module contains all the data related to the system such as users that are registered into the system and so on.
4. **Admin Module:** All the updates and addition and deletion of information related to products and other information are updated via this module. Only admin has the right to access this module.

8.SOFTWERE REQUIREMENT SPECIFICATION (SRS):

TABLE OF CONTENTS:

8.1 Introduction

8.1.1 Purpose

8.1.2 Overview

8.2 Functional Requirement

8.2.1 User signup

8.2.1 User sign in

8.3 Survey of Technology

8.4 Performance Requirement

8.5 Design Constraints

8.6 Non functional requirement

8.6.1 Security

8.6.2 Reliability

8.6.3 Availability

8.6.4 Portability

8.1 Introduction:

The project includes mainly two sections

Admin Section

Admin has all the access throughout the project.

Admin can add category, sub-category, products, image, price, GST(CGST & SGST) in the database.

Admin can update category, sub-category, products, image, price, GST(CGST & SGST) in the database.

Admin can delete category, sub-category, products, image, price, GST(CGST & SGST) from the database.

Admin can fetch information.

There is a static page available, if the static page be is changed by the central government then it will be reflected in the database.

Client Section

Client can register and sign in to the system.

If the client is sign in as a customer, then he/she can see all the details of the product(category, sub category, price, CGST, SGST).

If the client is sign in as a shopkeeper then after any payment a mail will be generated and it will be sent to central government mail id from the shopkeeper's mail id.

8.1.1 Purpose

This is a mobile based application that will allow the clients to know the information about the product details(CGST,SGST, product's category, sub category, total price). This application will reduce time. The data shall be safe, secure and easy to get. The database can be changed at any time if required.

8.1.2 Overview

This application provides an easy solution for the users who do not have the enough knowledge about all the current prices or CGST/SGST rates of the products . This system works when there is a global server storing the details of products. And it needs the usage of internet.

8.2 Functional requirement

8.2.1 User sign up

The customers/users first need to signup to use this application. After the successful registration the user can sign in to the system.

8.2.2 User sign in

User has to sign in to the system with the correct username and password.

8.3 Survey of Technology

To build this type of project we use

- **ANDROID STUDIO(Using JAVA and XML)**
- **Firebase as backend(database)**

8.4 Performance Requirement

There is no performance requirement in this system because the server request and response is dependent on the end user internet connection.

8.5 Design Constraints

The system shall be built using a standard Android building IDE like Android Studio. Along with it shall be used firebase.

8.6 Non functional requirement

8.6.1 Security

- Entry level password and username for client/customer
- Prevention of an entry of an unauthorized user.
- One user cannot view record of another user.
- Maintaining a strong record keeping.
- Assurance of keeping secrecy from external users.
- Each an every step with an assigned functional security.

8.6.2 Reliability

The system provides storage of all databases. The main pillar of the system is the backup of the database which is continuously maintained and updated to reflect the most

recent changes. Thus the overall stability of the system depends on the stability of the container and its underlying operating system.

8.6.3 Availability

The system should be available all the times, meaning the user can access it using a web server, only restricted by the down time of the server on which the system runs.

8.6.4 Portability

The application is android and scripting language based. So the end user part is fully portable. Any system with an android operating system should be able to use the features of this system. It can run on mobile phones and tablets.

9. MINIMUM HARDWARE AND SOFTWARE REQUIREMENTS:-

TECHNOLOGIES USED

Firestore

Android Studio

GPS

- **Hardware Requirements**

Hard Disk – 4 GB.

RAM – 2 GB.

Processor – i5 or higher.

Monitor

- **Software Requirements**

Front end -Android Studio

Backend- Firestore

Server- Firestore

Design Tool-Android software development Kit

Documentation Tool - Microsoft Word

Microsoft PowerPoint

10.USER CHARACTERISTICS:-

User must be comfortable in working with computer.

- Customer must know basic English.

11.CONSTRAINTS OF USE:-

- GUI is only in English.
- Admin must remember login ID and password.
- Internet Connection is needed.

12 PROCEDURE DISCUSSION:-

The application has been developed using certain easy steps that are to be followed properly and mandatorily. they are....

STEP 1

Anyone who wish to use the app must sign in first. That is only the registered users are allowed to access the application. Anyone , be it the user or the admin , must sign in first. The user is supposed to select the category from a drop down box and then by clicking the corresponding button moves to the corresponding sign in form.

STEP 2

In the sign in form the user is provided with two options. The “sign up “ and the “sign in” features. The first one is focused on new

users who are not registered but want to continue with the application and the second one is for the already registered users.

STEP 3

In the sign up feature the user is supposed to provide some basic information about him or her and the by doing so they become registered and allowed to enter into the application through proper sign in process.

STEP 4

The sign in feature allows its already registered users to enter into the activity form where they are about to be enriched with the information they search for.

STEP 5

After proper sign in the user can choose any item/product either by entering product name in search box or by searching through the categories/product list.

STEP 6

The user can then select its desired product and can opt for payment option.

STEP 7

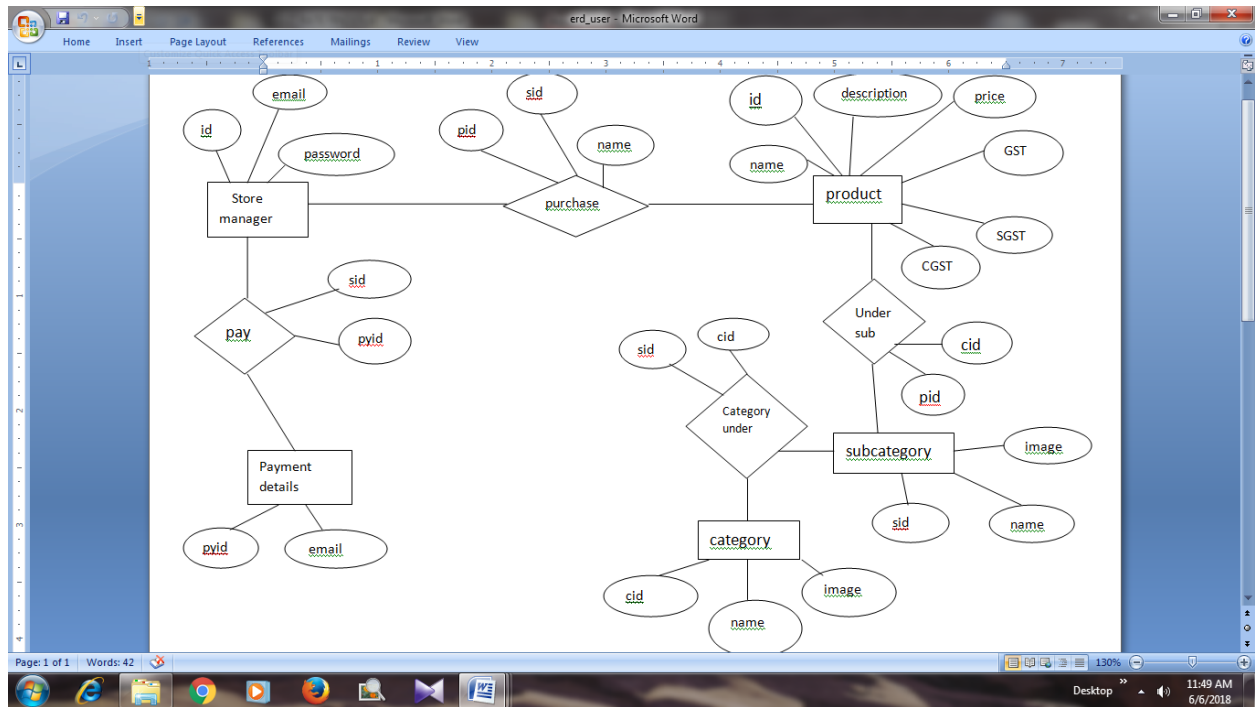
On clicking pay button email is sent to the Government mail id giving all the details of the product like product code,price etc

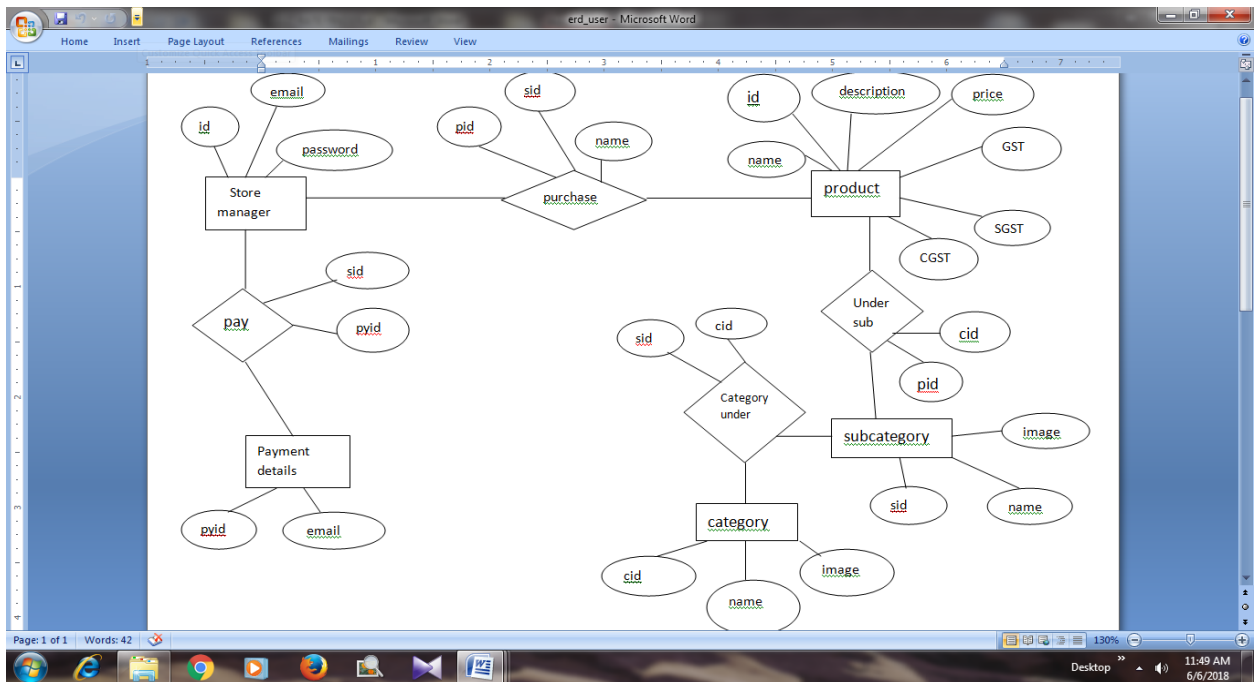
STEP 8

If signed as the admin, the admin can have two options. the “add category” and the “view all category/product”. By clicking the first option, admin will be forwarded to a form where by providing

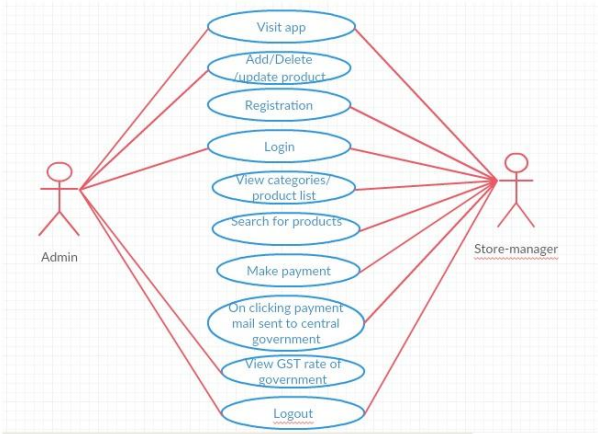
some informative data new product can be registered which will be registered into the firebase database. If clicked the view all product option, the admin can view all the registered product that have already been registered.

13.1 ENTITY-RELATIONSHIP DIAGRAM:-





13.2 USE CASE DIAGRAM



14. PROJECT SOURCE CODES :

JAVA code: **FUNCTION PRODUCT:**

```
package com.app.utility;

import android.content.Context;
import android.content.SharedPreferences;
import android.util.Log;

import com.app.mail.GMailSender;
import com.app.model.Product;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.GenericTypeIndicator;
import com.google.firebase.database.ValueEventListener;
import com.google.gson.Gson;

import java.util.HashMap;

/**
 * Created by "Megha" on 27-03-2018.
 */

public class FunctionsProduct {

    private static String RECEIVER = "centralgovernment123@gmail.com";

    public FunctionsProduct() {

    }

    /*THIS CALL IS TO HAVE LIST OF PRODUCTS*/
    public void callListOfProduct(String id, final CallBack callBack) {

        FirebaseDatabase mDataReference = FirebaseDatabase.getInstance();
        DatabaseReference mSto = mDataReference.getReference(id);
        mSto.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                try {
                    GenericTypeIndicator<HashMap<String, Product>> t = new
GenericTypeIndicator<HashMap<String, Product>>() {
                    };
                    final HashMap<String, Product> yourStringArray =
dataSnapshot.getValue(t);
                    if (yourStringArray.size() > 0) {
                        callBack.onSuccessP(yourStringArray);
                    } else {
                        callBack.onFailure();
                    }
                } catch (Exception e) {
                    e.printStackTrace();
                    callBack.onFailure();
                }
            }
        })
    }
}
```

```

        @Override
        public void onCancelled(DatabaseError error) {
            Log.w("TAG", "Failed to read value.", error.toException());
        }
    });

}
/*THIS CALL IS TO HAVE LIST OF CATEGORIES*/

public void sendMail(final String value, final Product __productDetails,
                    final CallbackProducts callBackProducts) {

    Thread sender = new Thread(new Runnable() {
        @Override
        public void run() {
            try {

                String data = "Product Name : " +
                    __productDetails.getProduct_name() + "\n" +
                        "Category Name : " + __productDetails.getCat_name() +
                    "\n" +
                        "Sub Category Name : " +
                    __productDetails.getSub_cat_name() + "\n" +
                        "Product Base Price : Rs. " +
                    __productDetails.getProduct_price() + "\n" +
                    // "Product GST(%) : " +
                    __productDetails.getSub_cat_cgst() + "\n" +
                        "Product SGST(%) : " + __productDetails.getSub_cat_sgst()
                    + "\n" +
                        "Product CGST(%) : " + __productDetails.getSub_cat_cgst()
                    + "\n" +
                        "Product Final Price : " + value + "\n" +
                        "ProductImageURL : " +
                    __productDetails.getProduct_image_url();

                GMailSender sender = new GMailSender();
                sender.sendMail("Product Details", data, Singleton.SENDER,
                    RECEIVER, callBackProducts);

            } catch (Exception e) {
                callBackProducts.onFailure();
                Log.e("mylog", "Error: " + e.getMessage());
            } finally {
                callBackProducts.onFinal();
            }
        }
    });
    sender.start();
}

public void getProductData(Context mContext, CallbackProducts callBackProducts) {
    try {
        SharedPreferences mEditor = mContext.getSharedPreferences("shared",
            Context.MODE_PRIVATE);
        Gson gson = new Gson();
        String __productDetailsDetails = mEditor.getString("ProductDetails", "");
        Product __productDetails = gson.fromJson(__productDetailsDetails,
            Product.class);
        callBackProducts.onSuccess(__productDetails);
    } catch (Exception e) {
        callBackProducts.onFailure();
    } finally {

```



```

        callBackProducts.onFinal();
    }

}

public float getTotalPrice(Product __productDetails, CallBackSetProducts
setProducts) {
    float price = Float.parseFloat(__productDetails.getProduct_price());
    float total_gst = 1;
    if (Singleton.mProductList != null) {

//        ArrayList<Category> list = new
Utility().clearListFromDuplicateFirstName(Singleton.mainList);

//        for (Category category : list) {
String catName = __productDetails.getCat_name().toString().trim();
//        String comCatName = __productDetails.getCat_name().toString().trim();
//        if (catName.equalsIgnoreCase(comCatName)) {

String cat_gst = __productDetails.getSub_cat_gst();
String cat_sgst = __productDetails.getSub_cat_sgst();
String cat_cgst = __productDetails.getSub_cat_cgst();

if (cat_cgst != "" && cat_sgst != null) {
    float sub_cat_sgstF = Float.parseFloat(cat_sgst);
//        float sub_gstF = Float.parseFloat(cat_gst);
    float cat_cgstF;
    try {

        cat_cgstF = Float.parseFloat(cat_cgst);
    } catch (Exception e) {
        cat_cgstF = 0;
    }
    total_gst = /*sub_gstF +*/ sub_cat_sgstF + cat_cgstF;

    setProducts.onSuccess(Float.parseFloat(cat_gst), sub_cat_sgstF,
cat_cgstF);

//        textViewGSTPercent.setText(String.valueOf(sub_gstF));
//        textViewSGSTPercent.setText(String.valueOf(sub_cat_sgstF));
//        } else if (cat_gst != "" && cat_sgst != null) {
//            int cat_gstF = Integer.parseInt(cat_gst);
//            int cat_sgstF = Integer.parseInt(cat_sgst);
//            total_gst = cat_gstF + cat_sgstF;
//            setProducts.onSuccess(cat_gstF, cat_sgstF);
//            textViewGSTPercent.setText(String.valueOf(cat_gstF));
//            textViewSGSTPercent.setText(String.valueOf(cat_sgstF));
//        }
//        break;
//    }
}

float value = price * total_gst / 100;
price = price + value;
return price;
}

/*INTERFACE TO CALL BACK PRODUCTS RESULTSTS*/
public interface CallBackProducts {
    void onSuccess(Product pD);
}

```

```

        void onFailure();

        void onFinal();
    }

    /*INTERFACE TO CALL BACK PRODUCTS RESULTSTS*/
    public interface CallBackSetProducts {
        void onSuccess(float gst, float sgst, float cgst);

        void onFailure();

        void onFinal();
    }
}
CHILD VIEW: package com.app.utility;

import android.view.View;
import android.widget.TextView;

import com.app.gstratefinder.R;

public class ChildViewHolder extends
com.thoughtbot.expandablerecyclerview.viewholders.ChildViewHolder {

    private TextView childTextView, child_gstTextView, child_sgstTextView;

    public ChildViewHolder(View itemView) {
        super(itemView);
        childTextView = (TextView) itemView.findViewById(R.id.list_item_artist_name);
        child_gstTextView = (TextView) itemView.findViewById(R.id.child_gst);
        child_sgstTextView = (TextView) itemView.findViewById(R.id.child_sgst);
    }

    public void setArtistName(String name) {
        childTextView.setText(name);
    }

    public void setGSTName(String name) {
        child_gstTextView.setText(name);
    }

    public void setSGSTName(String name) {
        child_sgstTextView.setText(name);
    }

}
CHILD CLASS: package com.app.utility;

import android.os.Parcel;
import android.os.Parcelable;

import com.app.model.Category;

public class ChildClass implements Parcelable {

    public static final Creator<ChildClass> CREATOR = new Creator<ChildClass>() {
        @Override
        public ChildClass createFromParcel(Parcel in) {
            return new ChildClass(in);
        }
    }
}

```

```

        @Override
        public ChildClass[] newArray(int size) {
            return new ChildClass[size];
        }
    };
    private String name, gst, sgst, cgst, csgst;
    private boolean isFavorite;

    public ChildClass(Category category, boolean isFavorite) {
//        this.name = name;
//        this.isFavorite = isFavorite;

//        if (category.getSub_cat_name() != null &&
//            !category.getSub_cat_name().equalsIgnoreCase("")) {
//            this.name = category.getSub_cat_name();
//            this.gst = category.getSub_cat_gst();
//            this.sgst = category.getSub_cat_sgst();
//            this.cgst = category.getCat_gst();
//            this.csgst = category.getCat_sgst();
//        } else {
            this.name = category.getSub_cat_name();
            this.gst = category.getCat_gst();
            this.sgst = category.getCat_sgst();
            this.cgst = category.getCat_gst();
            this.csgst = category.getCat_sgst();
//        }

    }

    protected ChildClass(Parcel in) {
        name = in.readString();
    }

    public String getName() {
        return name;
    }

    public String getGst() {
        return gst;
    }

    public String getSgst() {
        return sgst;
    }

    public boolean isFavorite() {
        return isFavorite;
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (!(o instanceof ChildClass)) return false;

        ChildClass artist = (ChildClass) o;

        if (isFavorite() != artist.isFavorite()) return false;
        return getName() != null ? getName().equals(artist.getName()) :

```

```

artist.getName() == null;

    }

    @Override
    public int hashCode() {
        int result = getName() != null ? getName().hashCode() : 0;
        result = 31 * result + (isFavorite() ? 1 : 0);
        return result;
    }

    @Override
    public void writeToParcel(Parcel dest, int flags) {
        dest.writeString(name);
    }

    @Override
    public int describeContents() {
        return 0;
    }

    public String getcGst() {
        return cgst;
    }

    public String getCSgst() {
        return csgst;
    }
}

```

XML code:

LOGIN LAYOUT XML:

```

<?xml version="1.0" encoding="utf-8"?><!--<ScrollView >-->
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_gravity="center"
    android:background="@drawable/background_base"
    android:gravity="center"
    android:orientation="vertical"
    tools:context="com.app.gstratefinder.LoginActivity">

    <TextView
        android:id="@+id/freetext"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:text="@string/app_name"
        android:textAppearance="@android:style/TextAppearance.DeviceDefault.Large"
        android:textSize="30sp"
        android:typeface="serif" />

    <android.support.design.widget.TextInputLayout
        android:id="@+id/freeinput"
        android:layout_width="match_parent"

```

```
android:layout_height="wrap_content"
android:layout_below="@+id/freetext"
android:layout_margin="15dp">
```

```
<EditText
    android:id="@+id/editTextEmail"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:hint="@string/email_address"
    android:imeOptions="actionNext"
    android:inputType="textEmailAddress"
    android:text=""
    android:textColor="@android:color/white"
    android:textColorHint="@android:color/white"
    android:typeface="serif" />
```

```
</android.support.design.widget.TextInputLayout>
```

```
<android.support.design.widget.TextInputLayout
    android:id="@+id/freeinput1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/freeinput"
    android:layout_margin="15dp">
```

```
<EditText
    android:id="@+id/editTextPassword"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="@string/password"
    android:imeOptions="actionDone"
    android:inputType="textPassword"
    android:text=""
    android:textColor="@android:color/white"
    android:textColorHint="@android:color/white"
    android:typeface="serif" />
```

```
</android.support.design.widget.TextInputLayout>
```

```
<Button
    android:id="@+id/signinButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/freeinput1"
    android:layout_centerHorizontal="true"
    android:layout_margin="10dp"
    android:layout_marginLeft="10dp"
    android:layout_marginRight="10dp"
    android:background="@drawable/button_effect"
    android:onClick="onClickSignIn"
    android:text="@string/sign_in"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:textColor="@color/header_color"
    android:textStyle="bold"
    android:typeface="serif" />
```

```
<com.github.guilhe.circularprogressview.CircularProgressView
    android:id="@+id/freeinput3"
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:layout_alignBottom="@+id/freeinput1"
```

```

        android:layout_centerHorizontal="true"
        android:visibility="gone"
        app:backgroundAlphaEnabled="true"
        app:backgroundColor="@android:color/transparent"
        app:max="10"
        app:progress="0"
        app:progressBarColor="@android:color/white"
        app:progressBarThickness="5dp"
        app:progressThumb="true"
        app:shadow="true"
        app:startingAngle="0" />

<TextView
    android:id="@+id/register_here_id"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/signinButton"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="15dp"
    android:gravity="center_horizontal"
    android:text="@string/not_a_member"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:textColor="@android:color/white"
    android:typeface="serif" />

<TextView
    android:id="@+id/forgot_pass_id"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignEnd="@+id/freeinput1"
    android:layout_below="@+id/register_here_id"
    android:layout_marginEnd="19dp"
    android:layout_marginTop="18dp"
    android:gravity="center_horizontal|right"
    android:text="@string/change_pass"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:textColor="@android:color/white"
    android:typeface="serif" />
</RelativeLayout>

<!--
</ScrollView>-->

```

PRODUCT LAYOUT XML:

```

<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_gravity="center"
    android:background="@drawable/background_base"
    android:gravity="center">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:background="@drawable/background_base"
        android:gravity="center"
        android:orientation="vertical">

        <!-- <LinearLayout

```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:background="@drawable/border"
    android:orientation="vertical">-->
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:layout_margin="12dp"
    android:background="@color/header_color"
    android:gravity="center_horizontal">
```

```
    <TextView
```

```
        android:id="@+id/addCatProductTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="5dp"
        android:text="@string/add_product"
        android:textAllCaps="true"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:textColor="@android:color/white"
        android:textStyle="bold"
        android:typeface="serif" />
```

```
</LinearLayout>
```

```
<android.support.design.widget.TextInputLayout
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="5dp">
```

```
    <AutoCompleteTextView
```

```
        android:id="@+id/addCatProduct"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/category_name"
        android:imeOptions="actionNext"
        android:inputType="textCapSentences|textAutoComplete"
        android:padding="5dp"
        android:popupBackground="@color/header_color"
        android:singleLine="true"
        android:textAlignment="viewStart"
        android:textAllCaps="true"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:textColor="@android:color/white"
        android:textColorHint="@android:color/white"
        android:textStyle="normal"
        android:typeface="serif" />
```

```
</android.support.design.widget.TextInputLayout>
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="5dp">
```

```
    <android.support.design.widget.TextInputLayout
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
```

```

        <AutoCompleteTextView
            android:id="@+id/addSubCatProduct"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="@string/add_sub_category"
            android:inputType="textCapSentences"
            android:singleLine="true"
            android:textAppearance="?android:attr/textAppearanceMedium"
            android:textColor="@android:color/white"
            android:textColorHint="@android:color/white"
            android:textStyle="normal"
            android:typeface="serif" />

    </android.support.design.widget.TextInputLayout>

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_margin="5dp"
    android:gravity="center"
    android:orientation="horizontal"
    android:weightSum="1">

    <android.support.design.widget.TextInputLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="0.5">

        <EditText
            android:id="@+id/editTextAddProduct"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:hint="@string/product_name"
            android:inputType="text|textCapSentences"
            android:singleLine="true"
            android:text=""
            android:textColor="@android:color/white"
            android:textColorHint="@android:color/white"
            android:typeface="serif" />

    </android.support.design.widget.TextInputLayout>

</LinearLayout>

<android.support.design.widget.TextInputLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="5dp">

    <EditText
        android:id="@+id/editTextProductDes"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:hint="@string/product_description"

```



```
        android:inputType="textCapWords|textCapSentences"
        android:singleLine="true"
        android:text=""
        android:textColor="@android:color/white"
        android:textColorHint="@android:color/white"
        android:typeface="serif" />

</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextInputLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="5dp">

    <EditText
        android:id="@+id/editTextProductPrice"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:hint="@string/product_price"
        android:inputType="numberDecimal"
        android:singleLine="true"
        android:textColor="@android:color/white"
        android:textColorHint="@android:color/white"
        android:typeface="serif" />

</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextInputLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:visibility="visible">

    <EditText
        android:id="@+id/editTextCatGST"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="15dp"
        android:hint="@string/gst"
        android:imeOptions="actionNext"
        android:inputType="numberDecimal"
        android:singleLine="true"
        android:textAlignment="viewStart"
        android:textColor="@android:color/white"
        android:textColorHint="@android:color/white"
        android:typeface="serif" />

</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextInputLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:visibility="visible">

    <EditText
        android:id="@+id/editTextCatSGST"
        android:layout_width="match_parent"
```

```

        android:layout_height="wrap_content"
        android:layout_marginTop="15dp"
        android:hint="@string/sgst"
        android:imeOptions="actionNext"
        android:inputType="numberDecimal"
        android:singleLine="true"
        android:textAlignment="viewStart"
        android:textColor="@android:color/white"
        android:textColorHint="@android:color/white"
        android:typeface="serif" />

</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextInputLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:visibility="visible">

    <EditText
        android:id="@+id/editTextCatCGST"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="15dp"
        android:hint="@string/cgst"
        android:imeOptions="actionDone"
        android:inputType="numberDecimal"
        android:singleLine="true"
        android:textAlignment="viewStart"
        android:textColor="@android:color/white"
        android:textColorHint="@android:color/white"
        android:typeface="serif" />

</android.support.design.widget.TextInputLayout>

<!--<android.support.design.widget.TextInputLayout-->
<!--android:layout_width="match_parent"-->
<!--android:layout_height="wrap_content"-->
<!--android:layout_margin="5dp"-->

<!--<EditText-->
<!--android:id="@+id/editTextAddProductRate"-->
<!--android:layout_width="match_parent"-->
<!--android:layout_height="wrap_content"-->
<!--android:layout_marginTop="10dp"-->
<!--android:hint="product_gst"-->
<!--android:inputType="number"-->
<!--android:singleLine="true"-->
<!--android:text=""-->
<!--android:textColor="@android:color/white"-->
<!--android:textColorHint="@android:color/white"-->
<!--android:typeface="serif" />-->

<!--</android.support.design.widget.TextInputLayout-->

```

15.PROJECT SNAPSHOT:-

GST Rate Finder

Email
abc@abc.abc

Password
.....|

SIGN IN

Not a member . Register Here!
Forgot Password?

REGISTER HERE

Email
a@abc.com

Password
.....

Confirm Password
.....|

SIGN UP

GST Rate Finder

Email

admin@admin.com

Password

.....

SIGN IN

Not a member . Register Here!

[Forgot Password?](#)



Product List

hp laptop

Product Name : Hp laptop
Category : Electronics
SubCat : Laptop

Product Name : Hp laptop
Category : Electronics
SubCat : Laptop

Category Sub Category Product

Government GST Rates

Name	GST	SGST	CGST
Laptop	18%	9%	9%
Mobile	12%	6%	6%
Camera	28%	14%	14%
Furniture	18%	9%	9%
Garments	5%	2.5%	2.5%
Paper Product	5%	2.5%	2.5%
Computer	18%	9%	9%
Fridge	28%	14%	14%
Children	5%	2.5%	2.5%

Product Details

PAY

Product Name : Cotton Saree
Product Category : Garments

Base Price : Rs. 599

GST (%) : 5.0

SGST (%) : 2.5

CGST (%) : 2.5

Total Price : Rs. 628.95

Description : Designer Saree

Category Sub Category Product

Category List

Household

Extras

Test

Music

Electronics

Garments

Category Sub Category Product

Category Name Here

Furniture

Add Sub Category

Dressing table

Product Name

Dressing table 310001

Product Description

Large mirror

Product Price

36000

GST Rate

14

SGST Rate

7

CGST Rate

7

Upload



16.FUTURE SCOPE:-

- The development process of this application can be easily followed by adding new features like payment can be done via card or cash.
- We will be working on the project to make it more interactive.

17. CONCLUSION:-

We provide a GST RATE FINDER application that is designed for Government and store manager on android smart phones. The goal of this application is to provide to eliminate fraudulent from the system so that whenever a good is sold the information is not hidden from Government and there is no plagiarize.

18. BIBLIOGRAPHY :-

- www.w3schools.com
- www.youtube.com
- www.developer.android.com