<GROCER ASSSIST>

A REPORT

submitted by

<J. SANJANA REDDY>(<17MIS1012>)

in partial fulfilment for the award

of

M. Tech. Software Engineering (Integrated)

School of Computer Science and Engineering

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

21AUGUST 2020

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering

DECLARATION

I hereby declare that the project entitled **"GROCER ASSIST"** submitted by me to the School of Computer Science and Engineering, Vellore Institute of Technology, Chennai Campus, Chennai 600127 in partial fulfilment of the requirements for the award of the degree of **Master of Technology - Software Engineering (Integrated)** is a record of bonafide work carried out by me**.** I further declare that the work reported in this report has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma of this institute or of any other institute or university.

J.Sanjana Reddy
Signature

<J.Sanjana Reddy> (<17mis1012>)

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

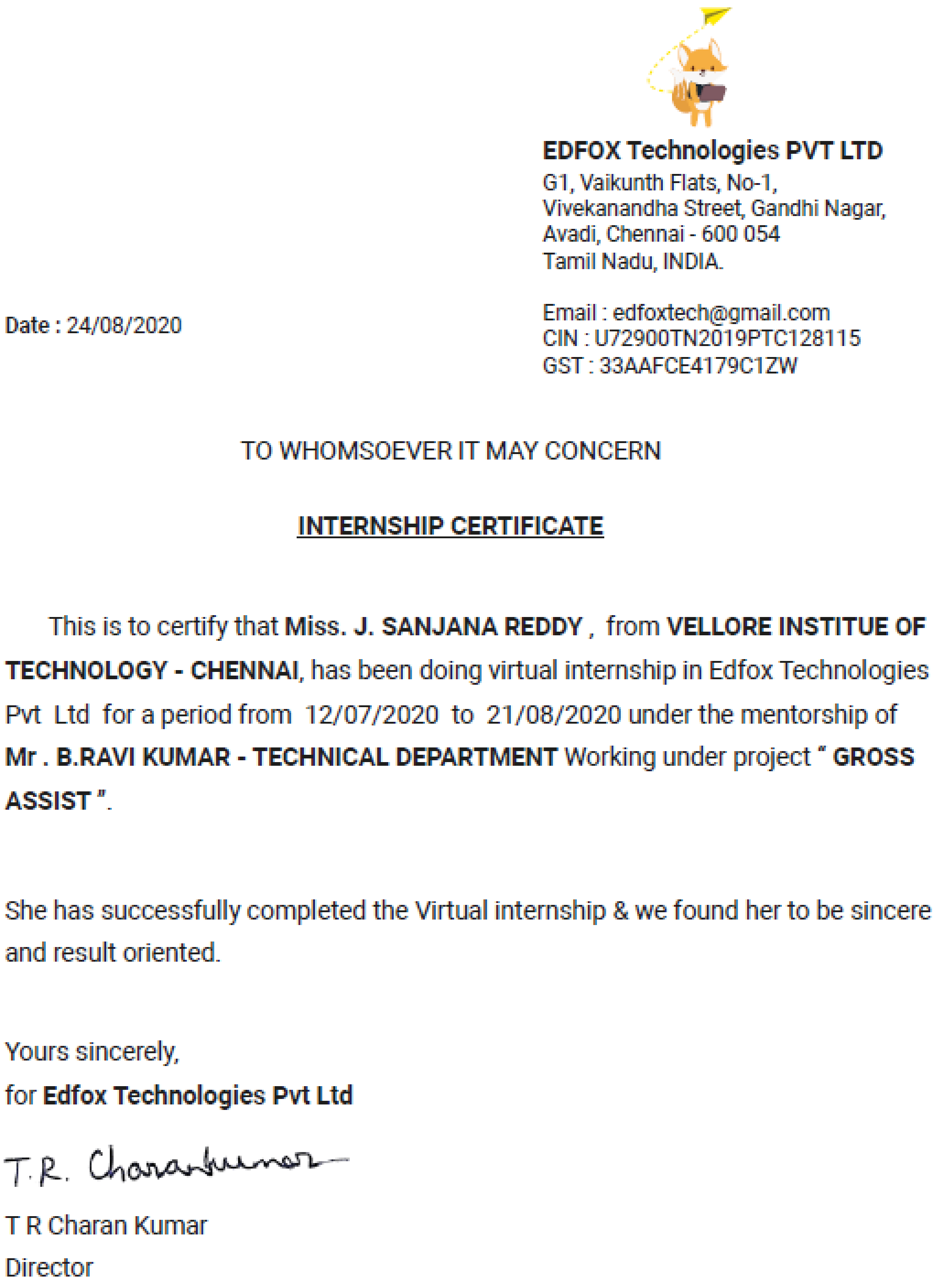
School of Computer Science and Engineering

CERTIFICATE

The project report entitled "**Grocer Assist**" is prepared and submitted by **J. Sanjana Reddy (17mis1012)**.Ithas been found satisfactory in terms of scope, quality and presentation as partial fulfilment of the requirements for the award of the degree of **Master of Technology – Software Engineering (Integrated)** in Vellore Institute of Technology, Chennai, India.

Examined by:Mr B RAVI KUMAR - TECHNICAL DEPARTMENT

INTERNSHIP CERTIFICATE



EDFOX Technologies PVT LTD
G1, Vaikunth Flats, No-1,
Vivekanandha Street, Gandhi Nagar,
Avadi, Chennai - 600 054
Tamil Nadu, INDIA.

Email : edfoxtech@gmail.com
CIN : U72900TN2019PTC128115
GST : 33AAFCE4179C1ZW

Date : 24/08/2020

TO WHOMSOEVER IT MAY CONCERN

INTERNSHIP CERTIFICATE

This is to certify that **Miss. J. SANJANA REDDY** , from **VELLORE INSTITUE OF TECHNOLOGY - CHENNAI**, has been doing virtual internship in Edfox Technologies Pvt Ltd for a period from 12/07/2020 to 21/08/2020 under the mentorship of **Mr . B.RAVI KUMAR - TECHNICAL DEPARTMENT** Working under project " **GROSS ASSIST** ".

She has successfully completed the Virtual internship & we found her to be sincere and result oriented.

Yours sincerely,
for **Edfox Technologies Pvt Ltd**

T.R. Charankumar

T R Charan Kumar
Director

ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to my HOD (Dr.AsnathVictyPhamila Y, Head of the Department (HoD), M.Tech Software Engineering (5 year integrated), SCSE, VIT Chennai) as well as our Dean(Dr. Jagadeesh Kannan R, Dean of the School of Computer Science & Engineering, VIT Chennai) as well as (Dr. Geetha S, Associate Dean of the School of Computer Science & Engineering, VIT Chennai)who gave me the golden opportunity to do this wonderful project on the topic (Grocer Assist), which also helped me in doing a lot of Research and I came to know about so many new things I am really thankful to them.

CONTENTS

TABLE OF CONTENTS

ABSTRACT

The application “**Grocer Assist**” is mainly designed to ease the grocery shopping. It has features which help the user to make a perfect grocery shopping.

The features include the storing of, list on the items that need to be purchased which is stored in a SQlite database and operations such as delete, update and view the list have been added. **Google maps API** is used to view the grocery stores in the user required area.

The app is developed with the above applications.

Present scenario of the area while getting Groceries:

Inventory loss. ...Injury to employees. ...Injury to **customers**. ...Property damage. ...At McCue, we strive to provide solutions to these **grocery** store **problems**. ...Contact us to see how you can eliminate **problems** at your **grocery** store.

Limitations of the present scenario.

People who doesnot have knowledge about the usage of the app can face problems and there might be a chance that they can avail into mistake by ordering some other things.

Problem addressed

Due to size of the Super Market it is impossible to find the Location of the Objects and Highly Impossible to check the avilablity of the particular product in the Super market

Proposed solution

“**Grocer Assist**” is mainly designed to ease the grocery shopping. It has features which help the user to make a perfect grocery shopping.

The features include the storing of, list on the items that need to be purchased which is stored in a SQlite database and operations such as delete, update and view the list have been added. **Google maps API** is used to view the grocery stores in the user required area

INTRODUCTION

A supermarket customer service **assistant's** duties and responsibilities **can** include guiding customers **to** the items they need, replenishing the **grocery** shelves when items go out of stock, assisting with the checkout process, helping customers load their **groceries** into their vehicles, cleaning the store and gathering ...

Wheat, rice, pulses, oil, cosmetics, etc. are the part of the life such that they cannot be seen as different from life. These products are available quickly and frequently at any store, every store has their set of glossary products which need to be managed properly in such a way that as one customer come to take the product the items can be easily removed and collected at a place, then it needs to be set and customer choose them by his input.

Supermarket customer service assistants both directly help customers inside the store and via telephone, as well as help with maintaining the store. A supermarket customer service assistant's duties and responsibilities can include guiding customers to the items they need, replenishing the grocery shelves when items go out of stock, assisting with the checkout process, helping customers load their groceries into their vehicles, cleaning the store and gathering shopping carts in the parking lot. They may also help out with customer returns and interact with management and other floor workers as necessary to handle complaints and requests.Much of the work is physical and can require bending, lifting, walking and using equipment like ladders and carts. A friendly personality and a willingness to meet customers' needs are necessary to be a successful supermarket customer service assistant

So that using system it can be located quickly and give instructions to staff to take out the products and the total of the goods are presented in the order table to total the price. The bill is made on that basis and given to the customers.

The admin maintains the detail of the clients and staff can be the counter manager or any staff. In this way, this gives a sense of management to the store and brisk the pace of work.

MODULES FOR GROCER ASSIST

NO	MODULES
1.	ITEMSELECTION
2.	ADD DATA
3.	UPDATE DATA
4.	DELETEDATA
5.	MAPS

1.ITEMSELECTION:

First the android Studio Produces web page output when connected to the mobile using the USB debugger and also intial the process of the object and it produces a greater output where there may be the different other modules are involved and the projections of the object towards the customer

First the android Studio Produces web page output when connectedto the mobile using the USB debugger and also intialize the process of the object and it produces a greater output where there may be the different other modules are involved and the projections of the object towards the customer needs helping customers load their groceries into their vehicles, cleaning the store and gathering ... and also to make the site usage

- **Name of item**

The user should enter the name of item which he wantsto choose from theSupermarket. web page output when connected to the mobile using the USB debugger and also intialize the process of the object and it produces a greater output where the module name of item shows the different items that are available in the general store and the quantities shapes and size.

The main motive is to give the name of item and quantity

- **Company**

The user is requested to enter the company name ofproduct. The Product name is mentioned to avail the product of company that is brand of the particular project to be checked to avail whether the requested brand is available or not

- **Quantity**:

The quantity of user's requirement should bemention's the requirement of the amount of the product what they wantedto avail or whether the required amount is not available.

- **ID**

  The id for the product is given by the sales people of the super market to the product and the id can be noted by the customer if they require that it is very greatful that if they can remember the id instead of saying the name every time they could just tell the sales person the id and they could be able to get the product details.

2 ADD DATA:

If the customer wants to buy more than one Item .The add data can be used. The add adds all the requirements of customer and makes it as a list.The add data makes the data to get the details of the product and they are given in the form of list.

- **View all**

It displays all the items in the form oflist.Inorder to keep them in the form of the list the view all is allowed and allocated. So that they can have the clear view of the list of the groceries

3.UPDATE DATA

The customer can update the data and store it. After the customer view the data .If he want to add something to the data he can add to the data and that can be done through the button add data after using the option add data he can explicit the update data so that the data required something else to the customer is acquired and stored explicitly and in the methodology of updation of data.

The updation of data can be can done so that you can also again use the command view all and you can explicit whether the items in the account are required.

4.DELETE DATA

After the view the customer has viewed the data In case customer does not need the data .the option of deleting it is also available .He can delete it. The delete data helps the customer to mainly delete the data and in the acquired capability whether the data can be held and unnecssary objects can be deleted from the Grocer assist.

5. MAPS:

The major important one is map the customer can know the exact location of the product.So that the customer easily go at the exact place in the super market and he can get the particular package of the product. By looking into the map the time complexity of the customer dealing with the customer became so less and avail acquired to the product can be known easily.

The help of maps makes the customer acknowledgment whether the product is available in the particular super Market.

TECHNOLOGY INVOLVED

Android Studio provides a unified environment where you can build apps for **Android** phones, tablets, **Android** Wear, **Android** TV,and**Android** AutoBy using the android studio we develop an app called “grocer assist”..

WORKING OF THE ANDRIOD STUDIO WITH GROCER ASSIST

A *project* in Android Studio contains everything that defines your workspace for an app, from source code and assets, to test code and build configurations. When you start a new project, Android Studio creates the necessary structure for all your files and makes them visible in the **Project** window on the left side of the IDE (click **View > Tool Windows > Project**). This page provides an overview of the key components inside your project.

Android app module

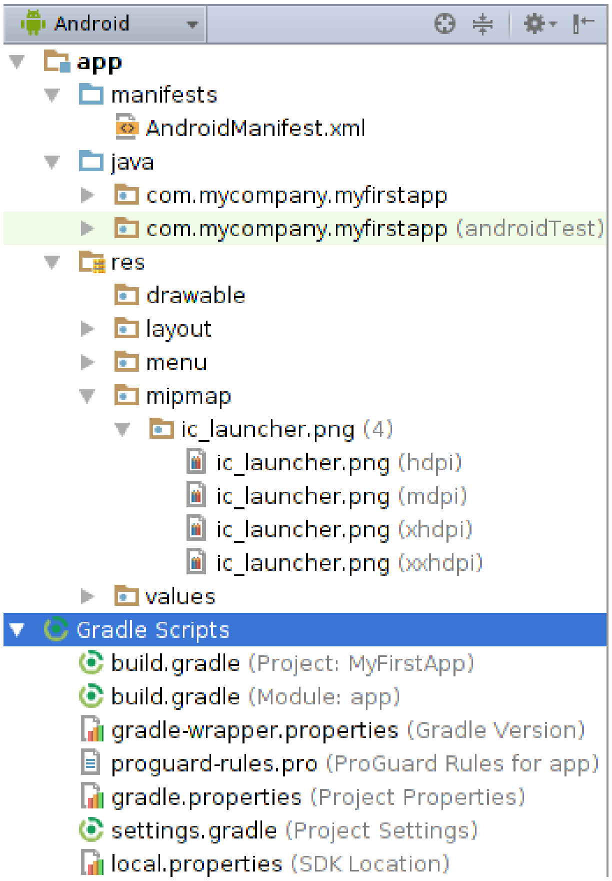
Provides a container for your app's source code, resource files, and app level settings such as the module-level build file and Android Manifest file. When you create a new project, the default module name is "app".

In the **Create New Module** window, Android Studio offers the following types of app modules:

- Phone & Tablet Module

```
dependencies {
  compile project(':my-library-module')
}
```

Project files



```
Android
▼ app
    ▼ manifests
          AndroidManifest.xml
    ▼ java
        ▶ com.mycompany.myfirstapp
        ▶ com.mycompany.myfirstapp (androidTest)
    ▼ res
          drawable
        ▶ layout
        ▶ menu
        ▼ mipmap
            ▼ ic_launcher.png (4)
                  ic_launcher.png (hdpi)
                  ic_launcher.png (mdpi)
                  ic_launcher.png (xhdpi)
                  ic_launcher.png (xxhdpi)
        ▶ values
▼ Gradle Scripts
      build.gradle (Project: MyFirstApp)
      build.gradle (Module: app)
      gradle-wrapper.properties (Gradle Version)
      proguard-rules.pro (ProGuard Rules for app)
      gradle.properties (Project Properties)
      settings.gradle (Project Settings)
      local.properties (SDK Location)
```

By default, Android Studio displays your project files in the **Android** view. This view does not reflect the actual file hierarchy on disk, but is organized by modules and file types to simplify navigation between key source files of your project, hiding certain files or directories that are not commonly used. Some of the structural changes compared to the structure on disk include the following:

- Shows all the project's build-related configuration files in a top-level **Gradle Script** group.
- Shows all manifest files for each module in a module-level group (when you have different manifest files for different product flavors and build types).
- Shows all alternative resource files in a single group, instead of in separate folders per resource qualifier. For example, all density versions of your launcher icon are visible side-by-side.Within each Android app module, files are shown in the following groups:
- **Manifests**

  Contains the AndroidManifest.xml file.

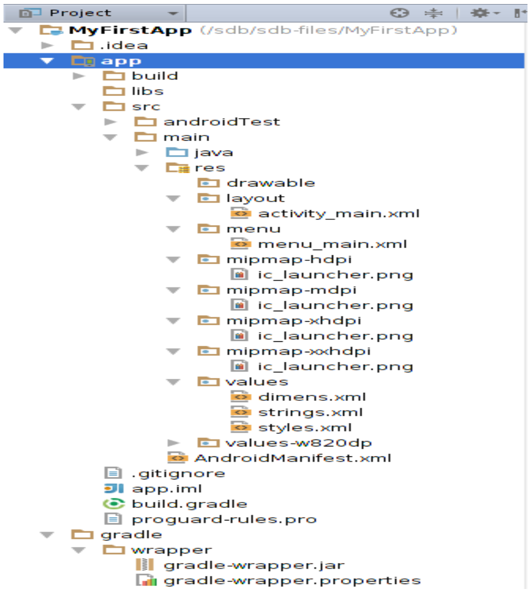
java

Contains the Java source code files, separated by package names, including JUnit test code.

res

Contains all non-code resources, such as XML layouts, UI strings, and bitmap images, divided into corresponding sub-directories. For more information about all possible resource type

The Android project view



```
Project
MyFirstApp (/sdb/sdb-files/MyFirstApp)
  .idea
  app
    build
    libs
    src
      androidTest
      main
        java
        res
          drawable
          layout
            activity_main.xml
          menu
            menu_main.xml
          mipmap-hdpi
            ic_launcher.png
          mipmap-mdpi
            ic_launcher.png
          mipmap-xhdpi
            ic_launcher.png
          mipmap-xxhdpi
            ic_launcher.png
          values
            dimens.xml
            strings.xml
            styles.xml
          values-w820dp
        AndroidManifest.xml
    .gitignore
    app.iml
    build.gradle
    proguard-rules.pro
  gradle
    wrapper
      gradle-wrapper.jar
      gradle-wrapper.properties
```

To see the actual file structure of the project including all files hidden from the Android view, select **Project** from the dropdown at the top of the **Project** window.

When you select **Project** view, you can see a lot more files and directories. The most important of which are the following:

module-name/

build/

Contains build outputs.

libs/

Contains private libraries.

src/

Contains all code and resource files for the module in the following subdirectories:

androidTest/

Contains code for instrumentation tests that run on an Android device. For more information, see the Android Test documentation.

main/

Contains the "main" sourceset files: the Android code and resources shared by all build variants (files for other build variants reside in sibling directories, such as src/debug/ for the debug build type).

AndroidManifest.xml

Describes the nature of the application and each of its components. For more information, see the AndroidManifest.xml documentation.

java/

Contains Java code sources.

jni/

Contains native code using the Java Native Interface (JNI). For more information, see the Android NDK documentation.

gen/

Contains the Java files generated by Android Studio, such as your R.java file and interfaces created from AIDL files.

res/

Contains application resources, such as drawable files, layout files, and UI string. See Application Resources for more information.

assets/

Contains file that should be compiled into an .apk file as-is. You can navigate this directory in the same way as a typical file system using URIs and read files as a stream of bytes using the AssetManager . For example, this is a good location for textures and game data.

test/

Contains code for local tests that run on your host JVM.

build.gradle **(module)**

This defines the module-specific build configurations.

build.gradle **(project)**

This defines your build configuration that apply to all modules. This file is integral to the project, so you should maintain them in revision control with all other source code.

For information about other build files, see Configure Your Build.

Project structure settings

To change various settings for your Android Studio project, open the **Project Structure** dialog by clicking **File > Project Structure**. It contains the following sections:

- **SDK Location:** Sets the location of the JDK, Android SDK, and Android NDK that your project uses.
- **Project:** Sets the version for Gradle and the Android plugin for Gradle, and the repository location name.
- **Developer Services:** Contains settings for Android Studio add-in components from Google or other third parties. See Developer Services, below.
- **Modules:** Allows you to edit module-specific build configurations, including the target and minimum SDK, the app signature, and library dependencies.

By using the android studio we develop an app called “grocer assist”.

Where we had written a code and Structured code modules allow you to divide your project into units of functionality that you can independently build, test, and debug…..

CODING AND IMPLEMENTATION

Database Code :-

```
 package com.kitty.Sanjanareddyjada.tuapp;
import   android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
 import android.database.sqlite.SQLiteDatabase;

 import android.database.sqlite.SQLiteOpenHelper;


 import org.jetbrains.annotations.Nullable;

public class DatabaseHelperextends SQLiteOpenHelper {

    public static final String DATABASE_NAME = "Student.db";
    public static final String TABLE_NAME = "student_table";
    public static final String COL_1 = "ID";
    public static final String COL_2 = "NAME";
    public static final String COL_3 = "SURNAME";
    public static final String COL_4 = "MARKS";

    public DatabaseHelper(@Nullable Context context) {

        super(context, DATABASE_NAME, null, 1);


    }
    @Override
    public void onCreate(SQLiteDatabasedb) {
        db.execSQL("create table " + TABLE_NAME +" (ID INTEGER PRIMARY KEY
,NAME TEXT,SURNAME TEXT,MARKS INTEGER)");

    }
    @Override
    public void onUpgrade(SQLiteDatabasedb, int i, int i1) {
        db.execSQL("DROP TABLE IF EXISTS "+TABLE_NAME);
        onCreate(db);
    }
    public booleaninsertData(String id,String name, String surname, String marks)

    {
        SQLiteDatabasedb = this.getWritableDatabase();
        ContentValuescontentValues = new ContentValues();
        contentValues.put(COL_1,id);
        contentValues.put(COL_2,name);
        contentValues.put(COL_3,surname);
        contentValues.put(COL_4,marks);
        long result = db.insert(TABLE_NAME,null,contentValues);
        if(result == -1)
        {
            return false;
        }
        else
            {
            return true;
            }
    }


    public Cursor getAllData()
    {
        SQLiteDatabasedb = this.getWritableDatabase();
        Cursor res = db.rawQuery("select * from "+TABLE_NAME,null);
        return res;
    }


    public booleanupdateData(String id,Stringname,Stringsurname,String marks)
    {
        SQLiteDatabasedb = this.getWritableDatabase();
        ContentValuescontentValues = new ContentValues();
        contentValues.put(COL_1,id);
        contentValues.put(COL_2,name);
        contentValues.put(COL_3,surname);
        contentValues.put(COL_4,marks);
        db.update(TABLE_NAME,contentValues, "id = ?",new String[] { id });
        return true;
    }


    public Integer deleteData (String id)
    {
        SQLiteDatabasedb = this.getWritableDatabase();
        return db.delete(TABLE_NAME, "ID = ?",new String[] { id });


    }
```

```
 }
```

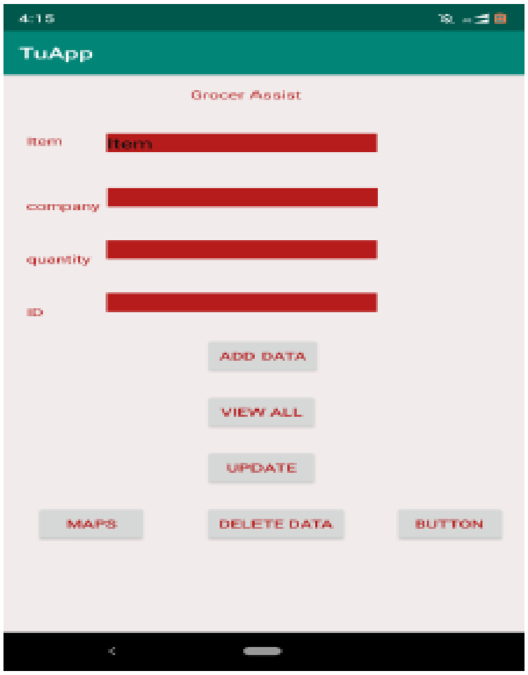
MainActivity Java code:-

```
 package com.kitty.jsanjanareddy.tuapp;



 import androidx.annotation.VisibleForTesting;

 import androidx.appcompat.app.AlertDialog;

 import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.database.Cursor;
import android.os.Bundle;
import android.os.Message;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
 public class MainActivityextends AppCompatActivity {

     private Button btnmove;
     private Button btnmessage;
     DatabaseHelpermyDB;
     EditTexteditName,editSurname,editMarks,editId;
     Button btnAddData;
     Button  btnviewAll;
     Button  btnupdate;
     Button btnDelete;

@Override
```

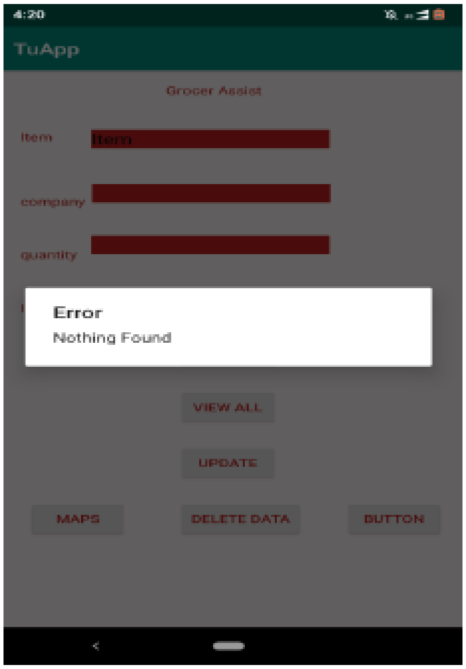
```
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        myDB= new DatabaseHelper(this);

editName= (EditText)findViewById(R.id.editText_Name);
editSurname= (EditText)findViewById(R.id.editText2_Surname);
editMarks= (EditText)findViewById(R.id.editText3_Marks); editId=
(EditText)findViewById(R.id.editText_Id);
        btnAddData= (Button)findViewById(R.id.button_add);
        btnviewAll= (Button)findViewById(R.id.button_viewAll);
        btnupdate= (Button)findViewById(R.id.button_update);
        btnDelete= (Button)findViewById(R.id.button_delete);
        btnmove= (Button)findViewById(R.id.button_maps);
        btnmove.setOnClickListener(new View.OnClickListener() {
            @Override

            public void onClick(View v) {
                moveToMaps() ;


            }
        });
        btnmessage= (Button)findViewById(R.id.button_message);
        btnmessage.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                movetoMessage();
            }
        });
        AddData();

        viewAll();

        UpdateData();

        DeleteData();
    }
```

TEST RESULTS

THE OUTPUT PAGE OF Grocer Assist



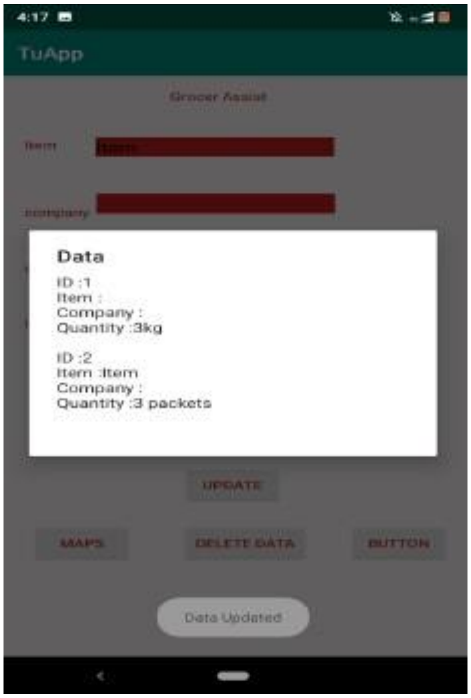
WHEN U UPDATE WITH NO INSERTION



Error

Nothing Found

INSERTION AND UPDATION(SHOWS LIST)



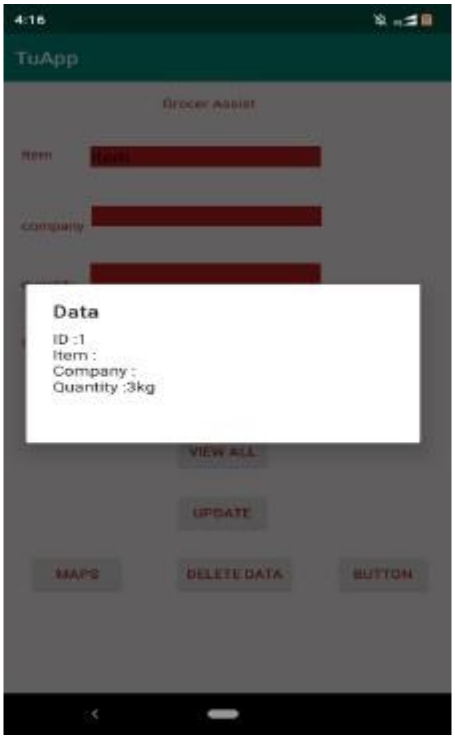
Data

ID :1
Item :
Company :
Quantity :3kg

ID :2
Item :Item
Company :
Quantity :3 packets

Data Updated

DELETION

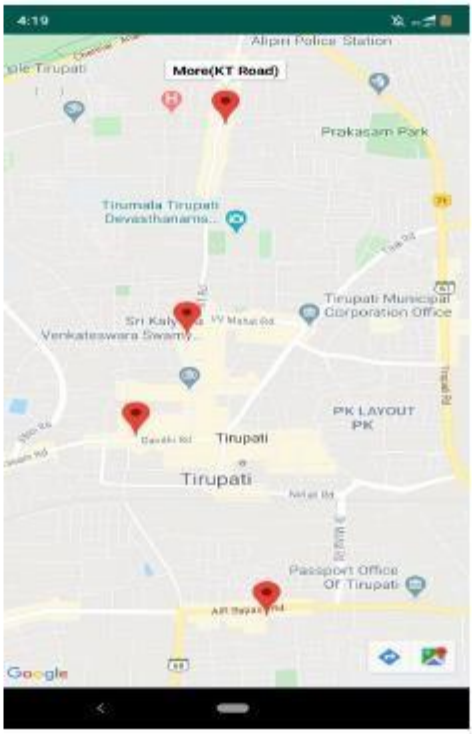


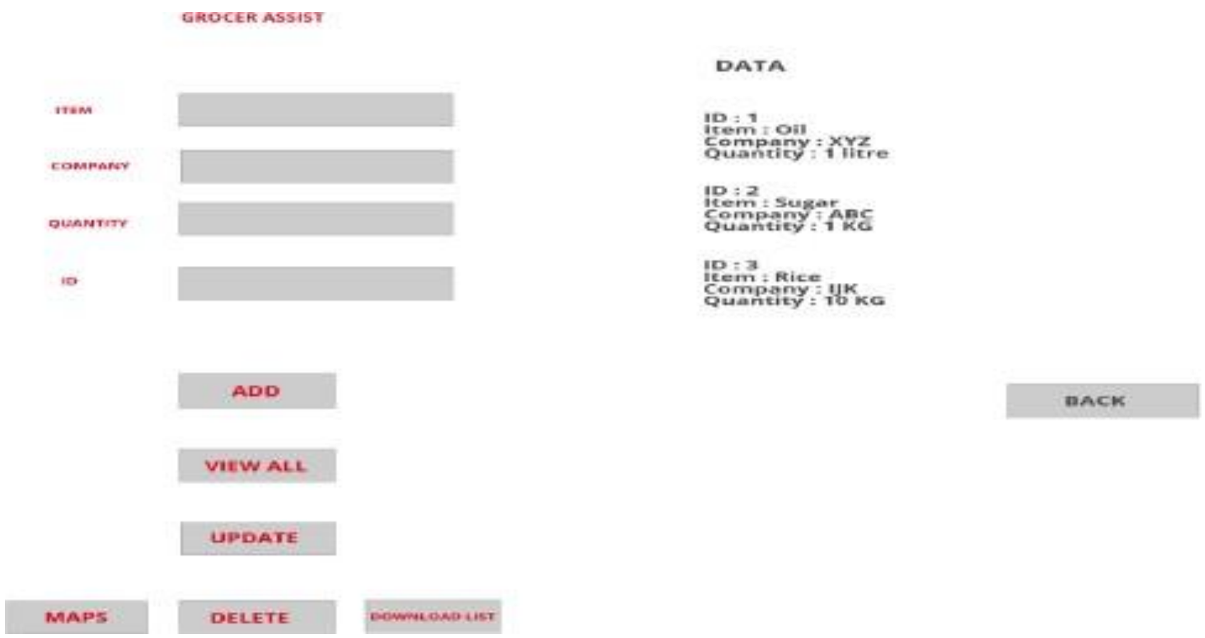
Data

ID :1
Item :
Company :
Quantity :3kg

MAPS

PROTOAPP:


4:19
Alipiri Police Station
More(KT Road)
Prakasam Park
Tirumala Tirupati Devasthanams...
Tirupati Municipal Corporation Office
Sri Kalyana Venkateswara Swamy...
PK LAYOUT PK
Tirupati
Tirupati
Passport Office Of Tirupati
Google



ID : 1
Item : Oil
Company : XYZ
Quantity : 1 litre

ID : 2
Item : Sugar
Company : ABC
Quantity : 1 KG

ID : 3
Item : Rice
Company : IJK
Quantity : 10 KG

CONCLUSION

This project can be used for grocery shopping which **makes out shopping easier by storing the list** and it also suggests the grocery stores in Google maps.The features include the storing of, list on the items that need to be purchased which is stored in a SQlite database and operations such as delete, update and view the list have been added. **Google maps API** is used to view the grocery stores in the user required area.

REFERENCES:

https://drive.google.com/file/d/1DqkO6RJs4dvmO_4olrClEm_NLPQ9Q4tr/view?usp=sharing