PROJECT REPORT

on

Food Guide – Know Your Food

(CS VI Semester Mini project)

2018-19

Submitted to:

Mr. B. P. Dubey

Submitted by:

Mr. Ayush Dhanai

Mr. Akhil Tiwari

Guided by:

Mr. Avnish Sharma

DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

GRAPHIC ERA HILL UNVERSITY, DEHRADUN

CERTIFICATE

Certified that Mr. Ayush Dhanai (Roll No.- 1018047) and Mr. Akhil Tiwari (Roll No. – 1018016) have developed mini project on "Food Guide – Know Your Food" for the CS VI Semester Mini Project Lab (PCS604) in Graphic Era Hill University, Dehradun under my supervision. The project carried out by Students is their own work as best of my knowledge.

Date:

(Mr. Avnish Sharma)

Project Guide

Assistant Professor

Department of CS/IT

GEHU, Dehradun

ACKNOWLEDGMENT

We would like to express our gratitude to The Almighty Shiva, the most Beneficent and the most Merciful, for completion of project.

We wish to thank our parents for their continuing support and encouragement. We also wish to thank them for providing us with the opportunity to reach this far in our studies.

We would like to thank particularly our project Co-ordinator Mr. B. P. Dubey and our Project Guide Mr. Avnish Sharma for his patience, support and encouragement throughout the completion of this project and having faith in us.

We also acknowledge to all teachers specially Shashi Sharma Sir, Rishika ma'am, Avinish Sir and Amit Gupta Sir who gave valuable suggestions to us in developing the project.

At last but not the least We greatly indebted to all other persons who directly or indirectly helped us during this work.

Mr. Ayush Dhanai **(Roll No.- 1018047)**

Mr. Akhil Tiwari **(Roll No.- 1018016)**

TABLE OF CONTENTS

LIST OF SYMBOLS, ABBREVIATIONS

∪	Union
∩	Intersection
⊆	Subset or Same
∞	Infinite
---	----------
---	---------- etc as per your project.

DBMS	Data Base Management System
RDBMS	Relational Data Base Management System
SQL	Structure Query Language
E-R	Entity Relationship
NoSQL	Not Only SQL

CHAPTER 1

INTRODUCTION

1.1 ABOUT PROJECT

“Food guide – Know Your Food”, is an android application which provides its users with a capability to easily identify the food they eat in everyday life and even let them know the recipe of that particular dish. This app is targeted for the food enthusiast who are curious enough to know the food they are eating and people who like to learn different recipes. Food guide uses the power of deep neural networks, Convolutional neural networks in particular for identifying the dish. And after the dish has been identified it fetches the recipe of that particular dish from our database and present it to the user. Currently this application has the capability to classify 101 recipes ranging from Chinese cuisines like sushi to street foods like hotdog and momos. Database on which our classifier has been trained has very high diversity in terms of the food recipes it contains, this makes this application very apt for all the people irrespective of the region or part of the world they live in.

Along with the Food guide android application we have also created a REST api service, which allows other developer to use our finely tuned and intensely trained classifier and use its functionalities by creating other applications on top of it.

1.2 PYTHON

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aims to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library

1.2.1 Flask

Flask is a micro web framework written in Python. It is classified as a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common frameworks related tools. Extensions are updated far more regularly than the core Flask program.

Applications that use the Flask framework include Pinterest, LinkedIn, and the community web page for Flask itself.

1.2.2 PyTorch

PyTorch is a computer software, specifically a machine learning library for the programming language Python, based on the Torch library used for applications such as natural language processing. It is primarily developed by Facebook's artificial-intelligence research group, and Uber's *Pyro* probabilistic programming software is built on it. It is free and open-source software released under one of the BSD licenses. Facebook operates both *PyTorch* and *Convolutional Architecture for Fast Feature Embedding* (Caffe2), but incompatibility made it difficult to transform a PyTorch-defined model into Caffe2 or vice versa. The Open Neural Network Exchange (ONNX) project was created by Facebook and Microsoft in September 2017 for converting models between frameworks. Caffe2 was merged into PyTorch at the end of March 2018.

1.3 DBMS

A database is a collection of related data which represents some aspect of the real world. A database system is designed to be built and populated with data for a certain task.

Database Management System (also known as DBMS) is a software for storing and retrieving users' data by considering appropriate security measures. It allows users to create their own databases as per their requirement.

It consists of a group of programs which manipulate the database and provide an interface between the database. It includes the user of the database and other application programs.

The DBMS accepts the request for data from an application and instructs the operating system to provide the specific data. In large systems, a DBMS helps users and other third-party software to store and retrieve data.

1.3.1 NoSQL

A **NoSQL** originally referring to non SQL or non-relational is a database that provides a mechanism for storage and retrieval of data. This data is modeled in means other than the tabular relations used in relational databases. Such databases came into existence in the late 1960s, but did not obtain the NoSQL moniker until a surge of popularity in the early twenty-first century. NoSQL databases are used in real-time web applications and big data and their use are increasing over time. NoSQL systems are also sometimes called Not only SQL to emphasize the fact that they may support SQL-like query languages.

A NoSQL database includes simplicity of design, simpler horizontal scaling to clusters of machines and finer control over availability. The data structures used by NoSQL databases are different from those used by default in relational databases which makes some operations faster in NoSQL. The suitability of a given NoSQL database depends on the problem it should solve. Data structures used by NoSQL databases are sometimes also viewed as more flexible than relational database tables.

Many NoSQL stores compromise consistency in favor of availability, speed and partition tolerance. Barriers to the greater adoption of NoSQL stores include the use of low-level query languages, lack of standardized interfaces, and huge previous investments in existing relational databases.

1.3.2 Entity Relationship (ER Modelling)

The ER or (Entity Relational Model) is a high-level conceptual data model diagram. Entity-Relation model is based on the notion of real-world entities and the relationship between them.

ER modeling helps you to analyze data requirements systematically to produce a well-designed database. So, it is considered a best practice to complete ER modeling before implementing your database.

Entity relationship diagram displays the relationships of entity set stored in a database. In other words, we can say that ER diagrams help you to explain the logical structure of databases. At first look, an ER diagram looks very similar to the flowchart. However, ER Diagram includes many specialized symbols, and its meanings make this model unique.

1.4 Android

Android is a software package and Linux based operating system for mobile devices such as tablet computers and smartphones. It is developed by Google and later the OHA (Open Handset Alliance). Java language is mainly used to write the android code even though other languages can be used. The goal of android project is to create a successful real-world product that improves the mobile experience for end users. There are many code names of android such as Lollipop, KitKat, Jelly Bean, Ice cream Sandwich, Froyo, Eclair, Donut etc.

Patterned after the Linux kernel, the Android also was released as open source code. Development for the Android may be done through Windows, Linux or Mac. Although primarily written in Java, there is no Java Development Machine (JDM) in the platform.

Instead of allowing Java programs to run through the JDM, Google developed Dalvik, a virtual machine specifically for the Android. Dalvik runs recompiled Java code and reads it as Dalvik bytecode and was designed to optimize battery power and maintain functionality in an environment with limited memory and CPU power, such as that of mobile phones, netbooks and tablet PCs.

One of the Android's selling points is an ability to break down application boundaries. Another advantage is that it is easily developed, not to mention its speed of app development.

1.5 Heroku

Heroku is a cloud platform as a service (PaaS) supporting several programming languages. Heroku, one of the first cloud platforms, has been in development since June 2007, when it supported only the Ruby programming language, but now supports Java, Node.js, Scala, Clojure, Python, PHP, and Go. For this reason, Heroku is said to be a polyglot platform as it has features for a developer to build, run and scale applications in a similar manner across most languages. Heroku was acquired by Salesforce.com in 2010 for $212 million.

Heroku was initially developed by James Lindenbaum, Adam Wiggins, and Orion Henry for supporting projects that were compatible with the Ruby programming platform known as Rack.The prototype development took around six months. Later on, Heroku faced drawbacks because of lack of proper market customers as many app developers used their own tools and environment. In Jan 2009 a new platform was launched which was built almost from scratch after a three-month effort. In October 2009, Byron Sebastian joined Heroku as CEO. On December 8, 2010, Salesforce.com acquired Heroku as a wholly owned subsidiary of Salesforce.com. On July 12, 2011, Yukihiro "Matz" Matsumoto, the chief designer of the Ruby programming language, joined the company as Chief Architect, Ruby. That same month, Heroku added support for Node.js and Clojure. On September 15, 2011, Heroku and Facebook introduced Heroku for Facebook. At present Heroku supports Redis databases in addition to its standard PostgreSQL. Application development on Heroku is primarily done through git. The application gets a new git remote typically named as Heroku along with its local git repository where the application was made. Hence to deploy heroku application is similar to using the git push command.

There are many other ways of deploying applications too. For example, developers can enable GitHub integration so that each new pull request is associated with its own new application, which enables all sorts of continuous integration scenarios. Dropbox Sync lets developers deploy the contents of Dropbox folders to Heroku, or the Heroku API can be used to build and release apps.

CHAPTER 2

PROJECT

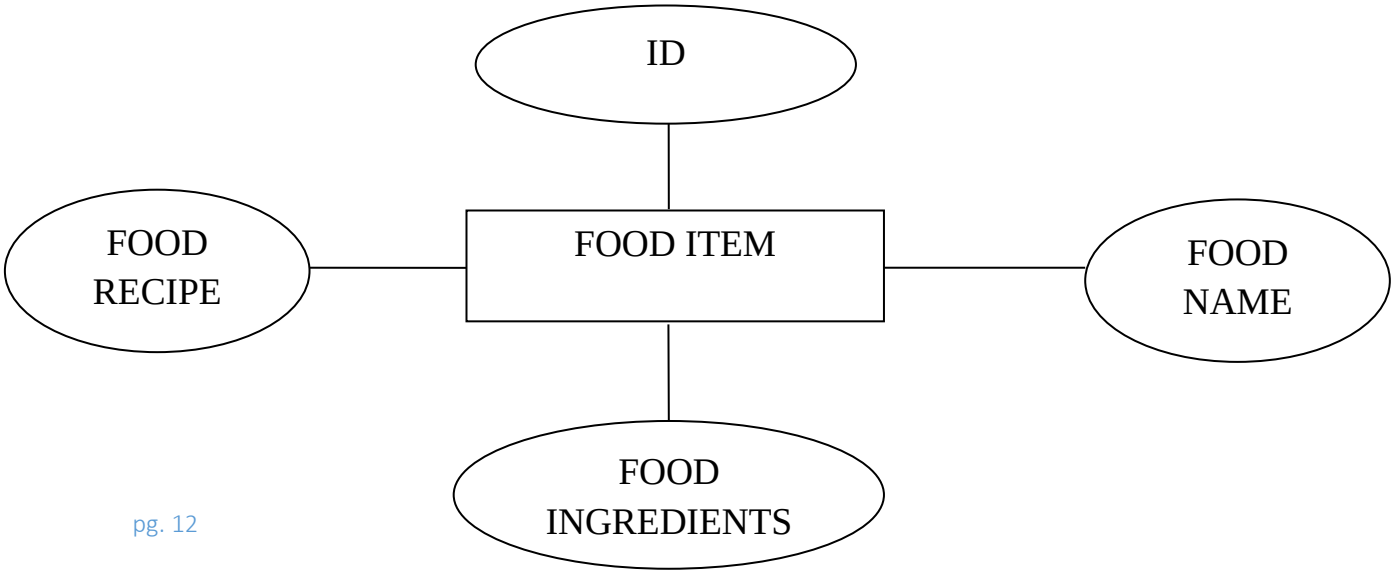
2.1 REQUIREMENT ANALYSIS

- Allow user to capture the and store the photos of their favourite dish.
- Allow to identify the dish.
- Allow user to know the recipe of the dish.
- Allow user to know the nutritional information of the dish.

2.2 SOFTWARE SPECIFICATIONS

- Tools and IDE: Android studio, VS Code, Google Colab
- Programming Languages: Java and Python (3.7)
- Libraries and Packages: Flask (1.0.2), torchvision (0.2.1), numpy (1.15.4), Pillow (5.3.0), gunicorn (19.9.0), torch (1.0)

2.3 ENTITY RELATIONSHIP (ER) DIAGRAM

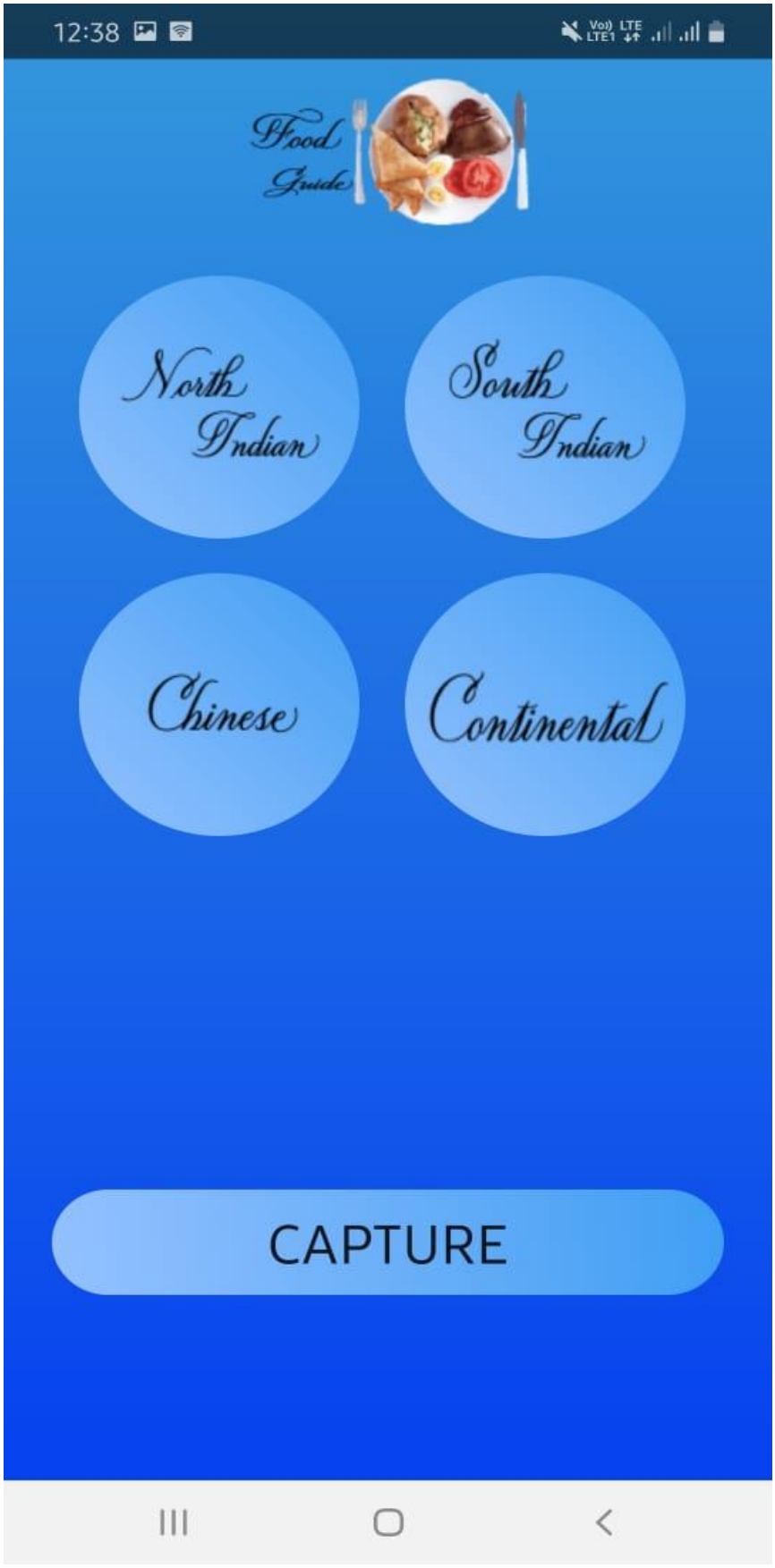


CHAPTER 3

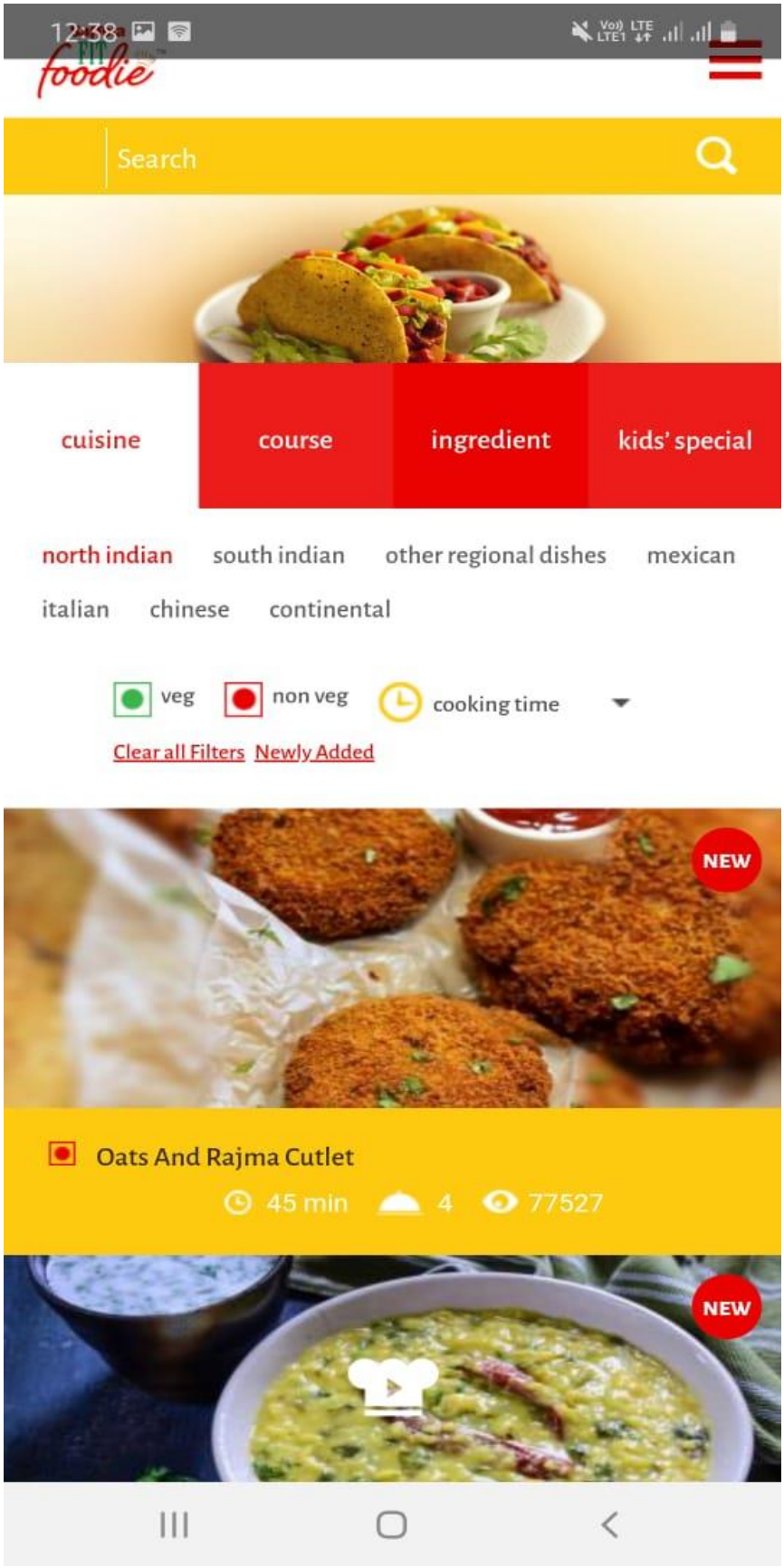
SNAPSHOT OF PROJECT

3.1 LAUNCHER ACTIVITY

3.2 HOME SCREEN

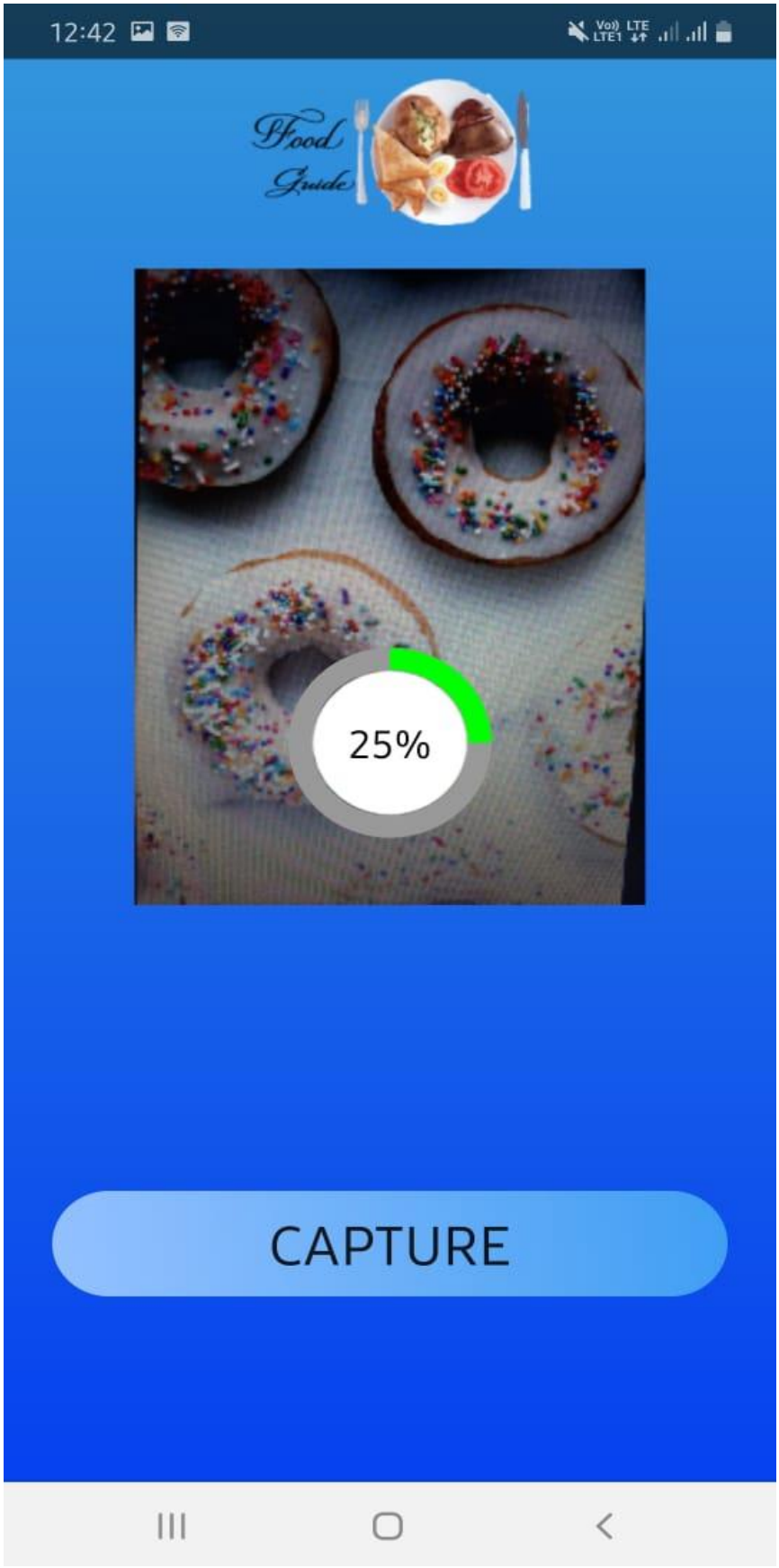


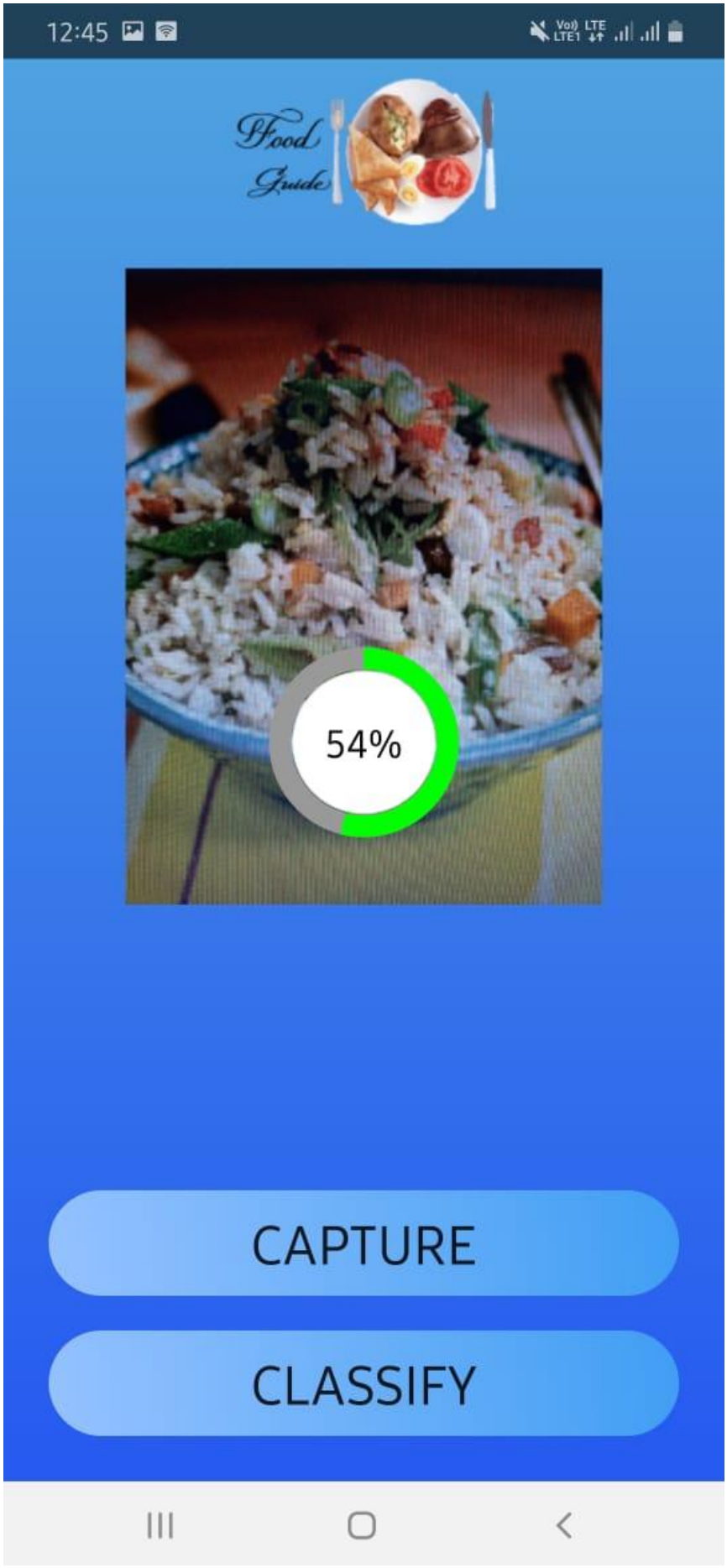
3.3 FOOD EXPLORER



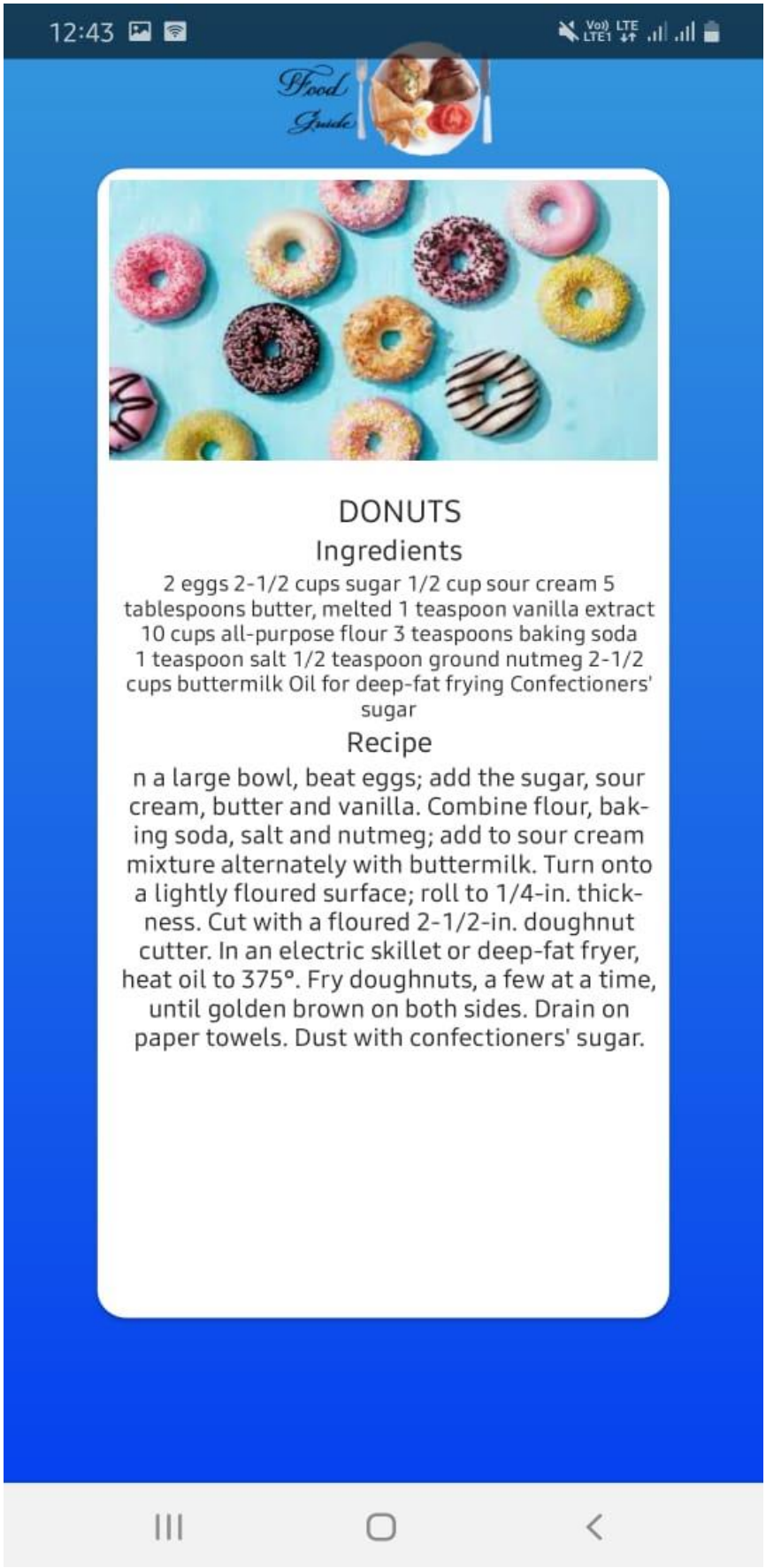

12:40
Cultural India
South Indian food includes the cuisines of the five states (Andhra Pradesh, Karnataka, Kerala, Tamil Nadu and Telangana) of south India.
Cultural India : Indian Food : South Indian Food
South Indian Food
South Indian food has earned much fame across the globe, particularly for scrumptious dishes like Dosa, Vada, Idli, Uttapam and Sambar. South Indian meals comprise cuisines of five South Indian states namely Tamil Nadu, Karnataka, Kerala, Andhra Pradesh and Telangana, along with several local cuisines within these states. The region offers a wide variety of vegetarian and non-vegetarian dishes with each state holding its own uniqueness and food habits. Some authentic and popular South Indian dishes that are sure to delight taste buds include Chakra Pongal, Sambar and Vadai from Tamil Nadu; Rava Idli from Karnataka; Kadala Curry and Appam from Kerala; and Kebabs and Biryanis from Andhra Pradesh.

3.4 INFERENCE




12:45
Food Guide
54%
CAPTURE
CLASSIFY

3.5 RESULT


12:43
Food Guide
DONUTS
Ingredients
2 eggs 2-1/2 cups sugar 1/2 cup sour cream 5 tablespoons butter, melted 1 teaspoon vanilla extract 10 cups all-purpose flour 3 teaspoons baking soda 1 teaspoon salt 1/2 teaspoon ground nutmeg 2-1/2 cups buttermilk Oil for deep-fat frying Confectioners' sugar
Recipe
n a large bowl, beat eggs; add the sugar, sour cream, butter and vanilla. Combine flour, baking soda, salt and nutmeg; add to sour cream mixture alternately with buttermilk. Turn onto a lightly floured surface; roll to 1/4-in. thickness. Cut with a floured 2-1/2-in. doughnut cutter. In an electric skillet or deep-fat fryer, heat oil to 375°. Fry doughnuts, a few at a time, until golden brown on both sides. Drain on paper towels. Dust with confectioners' sugar.

APPENDIX

Code

Android Application

```
package ml.ayushdhanai.foodguide;

import android.Manifest;
import android.content.Context;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.content.res.Resources;
import android.database.Cursor;
import android.graphics.Bitmap;
import android.graphics.drawable.AnimationDrawable;
import android.graphics.drawable.Drawable;
import android.net.Uri;
import android.os.Handler;
import android.provider.MediaStore;
import android.support.annotation.NonNull;
import android.support.v4.app.ActivityCompat;
import android.support.v4.content.ContextCompat;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.GridLayout;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ProgressBar;
import android.widget.RelativeLayout;
import android.widget.TextView;
import android.widget.Toast;
```

```
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.RetryPolicy;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonObjectRequest;
import com.android.volley.toolbox.Volley;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.UploadTask;

import org.json.JSONException;
import org.json.JSONObject;

import java.io.ByteArrayOutputStream;
import java.io.File;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Date;
import java.util.List;
import java.util.Objects;

public class MainActivity extends AppCompatActivity {

   //Permission
   public static final int REQUEST_ID_MULTIPLE_PERMISSIONS = 1;


   //Progress bar code
   int pStatus = 0;
   private Handler handler = new Handler();
   TextView tv;
   ProgressBar mProgress;
   Drawable drawable;
   Resources res;
   ImageView progressImageview;
```

```
private static final int CAMERA_REQUEST = 1888;
ImageView imageView;
Bitmap mBitmap;
Button button;
GridLayout gridLayout;
RelativeLayout mylayout;
AnimationDrawable animationDrawable;
private StorageReference mStorageRef;
String BASE_URL = "https://foodguide-101.herokuapp.com/api?url=";
String apiResponse;
ArrayList<Uri> alldownloadUrls = new ArrayList<Uri>();


@Override
protected void onCreate(Bundle savedInstanceState) {
   super.onCreate(savedInstanceState);
   setContentView(R.layout.activity_main);

   mylayout = findViewById(R.id.mylayout);

   gridLayout=(GridLayout) findViewById(R.id.gridlayout1);
   animationDrawable = (AnimationDrawable) mylayout.getBackground();
   animationDrawable.setEnterFadeDuration(4500);
   animationDrawable.setExitFadeDuration(4500);
   animationDrawable.start();


   //Permisiions
   checkAndRequestPermissions();

   //Progress bar code
   res = getResources();
   drawable = res.getDrawable(R.drawable.circular);
   mProgress = (ProgressBar) findViewById(R.id.circularProgressbar);
   progressImageview=(ImageView) findViewById(R.id.progressimageview);
   mProgress.setProgress(0);   // Main Progress
   mProgress.setSecondaryProgress(100); // Secondary Progress
   mProgress.setMax(100); // Maximum Progress
   mProgress.setProgressDrawable(drawable);
```

```
    /*  ObjectAnimator animation = ObjectAnimator.ofInt(mProgress,
"progress", 0, 100);
      animation.setDuration(50000);
      animation.setInterpolator(new DecelerateInterpolator());
      animation.start();*/

      tv = (TextView) findViewById(R.id.tv);

   }



   @Override
   protected void onRestart() {
      super.onRestart();
      mStorageRef =
FirebaseStorage.getInstance().getReferenceFromUrl("gs://foodguide-
101.appspot.com");


   }

   //Camera
   Uri tempUri;
   File finalFile;String pathx;String s[];
   StorageReference storageReference = null;
   Uri downloadUrl;

   String url;
   JsonObjectRequest objectRequest;
   protected void onActivityResult(int requestCode, int resultCode, Intent data)
{
      try {
         if (requestCode == CAMERA_REQUEST) {
            Bitmap photo = (Bitmap)
Objects.requireNonNull(data.getExtras()).get("data");
            mBitmap = photo;
            imageView = findViewById(R.id.imageView);
            imageView.setVisibility(View.VISIBLE);
```

```
gridLayout.setVisibility(View.INVISIBLE);
imageView.setImageBitmap(photo);

//FireBase Connectivity


// CALL THIS METHOD TO GET THE URI FROM THE BITMAP
tempUri = getImageUri(getApplicationContext(), photo);


// CALL THIS METHOD TO GET THE ACTUAL PATH
finalFile = new File(getRealPathFromURI(tempUri));


//Extracting file name from file path
pathx = String.valueOf(finalFile);
s = pathx.split("/");
pathx = s[s.length - 1];

//progressbar start

mProgress.setVisibility(View.VISIBLE);
tv.setVisibility(View.VISIBLE);
progressImageview.setVisibility(View.VISIBLE);

pStatus=0;
new Thread(new Runnable() {

   @Override
   public void run() {
      // TODO Auto-generated method stub
      while (pStatus < 100) {
         pStatus += 1;

         handler.post(new Runnable() {

            @Override
            public void run() {
               // TODO Auto-generated method stub
               mProgress.setProgress(pStatus);
```

```
                        tv.setText(pStatus + "%");

                    }
                });
                try {
                    // Sleep for 200 milliseconds.
                    // Just to display the progress slowly
                    Thread.sleep(300); //thread will take approx 10 seconds to
finish
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
            if(pStatus==100) {
                mProgress.setVisibility(View.INVISIBLE);
                tv.setVisibility(View.INVISIBLE);
                progressImageview.setVisibility(View.INVISIBLE);

            }

        }
    }).start();


    //Progress bar code end

    storageReference = mStorageRef.child(pathx);

    storageReference.putFile(tempUri)
            .addOnSuccessListener(new
OnSuccessListener<UploadTask.TaskSnapshot>() {
                @Override
                public void onSuccess(UploadTask.TaskSnapshot taskSnapshot)
{



storageReference.getDownloadUrl().addOnSuccessListener(new
OnSuccessListener<Uri>() {
                    @Override
```

```
                    public void onSuccess(Uri uri) {
                        downloadUrl = null;
                        downloadUrl = uri;


                        alldownloadUrls.add(downloadUrl);
                        Log.d("MainActivity", "URL : " + downloadUrl);

                        mStorageRef =
FirebaseStorage.getInstance().getReferenceFromUrl(String.valueOf(download
Url));



                        // Calling Rest API


                        // Instantiate the RequestQueue.
                        RequestQueue requestQueue =
Volley.newRequestQueue(getBaseContext());
                        url = BASE_URL + downloadUrl;

                        objectRequest = new JsonObjectRequest(
                              Request.Method.GET,
                              url,
                              null,
                              new Response.Listener<JSONObject>() {
                                 @Override
                                 public void onResponse(JSONObject response) {
                                    apiResponse = " ";
                                    Log.e("Rest Response", response.toString());
                                    apiResponse = response.toString();
                                 }
                              },
                              new Response.ErrorListener() {
                                 @Override
                                 public void onErrorResponse(VolleyError error) {
                                    Log.e("Rest Response", error.toString());
                                 }
                              }
```

```
                        );
                        //For handling timeout error
                        objectRequest.setRetryPolicy(new RetryPolicy() {
                           @Override
                           public int getCurrentTimeout() {
                              return 50000;
                           }

                           @Override
                           public int getCurrentRetryCount() {
                              return 50000;
                           }

                           @Override
                           public void retry(VolleyError error) throws VolleyError
{


                           }
                        });
                        requestQueue.add(objectRequest);


                        //Api part end

                        try {
                           Thread.sleep(10000);
                           button.setVisibility(View.VISIBLE);
                        } catch (InterruptedException e) {
                           e.printStackTrace();
                        }


                     }
                  });
               }
            })
            .addOnFailureListener(new OnFailureListener() {
               @Override
               public void onFailure(@NonNull Exception exception) {
                  // Handle unsuccessful uploads
```

```
                    // ...
                }
            });




        }
    }catch (Exception e){
        onResume();
    }



}



ByteArrayOutputStream bytes;String path;

public Uri getImageUri(Context inContext, Bitmap inImage) {
    bytes = new ByteArrayOutputStream();
    //inImage.compress(Bitmap.CompressFormat.JPEG, 100, bytes);
    inImage = Bitmap.createScaledBitmap(inImage, (int) (inImage.getWidth() *
2), (int) (inImage.getHeight() * 2), true);
    path =
MediaStore.Images.Media.insertImage(inContext.getContentResolver(),
inImage, "Title", null);
    return Uri.parse(path);
}

int idx;String pathUri;Cursor cursor;

public String getRealPathFromURI(Uri uri) {
    pathUri = "";
    if (getContentResolver() != null) {
        cursor = getContentResolver().query(uri, null, null, null, null);
        if (cursor != null) {
            cursor.moveToFirst();
```

```
            idx =
cursor.getColumnIndex(MediaStore.Images.ImageColumns.DATA);
            pathUri = cursor.getString(idx);
            cursor.close();
         }
      }
      return path;
   }

   @Override
   protected void onDestroy() {
      super.onDestroy();
      for (Uri a : alldownloadUrls) {
         mStorageRef =
FirebaseStorage.getInstance().getReferenceFromUrl(String.valueOf(a));
         //Deleting the file from Storage
         mStorageRef.delete().addOnSuccessListener(new
OnSuccessListener<Void>() {
            @Override
            public void onSuccess(Void aVoid) {
               // File deleted successfully
               Log.d("Main Activity  : ", "onSuccess: deleted file");
            }
         }).addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception exception) {
               // Uh-oh, an error occurred!
               Log.d("Main Activity  : ", "onFailure: did not delete file");
            }
         });
      }

   }

   JSONObject reader;
   protected void classify(View view) {

     // do {
         try {
```

```
            //Json parsing

            if(pStatus==100) {
               reader = new JSONObject(apiResponse);

               String status = reader.getString("status");

               if (status.equals("200")) {
                  String food_name = reader.getString("food_name");
                  String food_ingredients = reader.getString("food_ingredients");
                  String food_recipe = reader.getString("food_recipe");
                  String category = reader.getString("category");
                  Intent i = new Intent(this, ClassifyMainActivity.class);
                  i.putExtra("title", food_name);
                  i.putExtra("food_ingredients", food_ingredients);
                  i.putExtra("food_recipe", food_recipe);
                  i.putExtra("category", category);
                  startActivity(i);
               } else {
                  Toast.makeText(this, "Sorry Unable to Classify Food! ",
Toast.LENGTH_SHORT).show();
                  onResume();
               }


            }
            else
            {
               Toast.makeText(this, "Please wait for a second or two!",
Toast.LENGTH_SHORT).show();
               onResume();
            }

         } catch (JSONException e) {
            Toast.makeText(this, "Response Not received Wait for a While",
Toast.LENGTH_SHORT).show();
            onResume();
         }

   }
```

```
    @Override
    protected void onResume() {
        super.onResume();

        mStorageRef =
FirebaseStorage.getInstance().getReferenceFromUrl("gs://foodguide-
101.appspot.com");

        Button photoButton = this.findViewById(R.id.button1);
        button = (Button) findViewById(R.id.button2);

        photoButton.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {

                button.setVisibility(View.INVISIBLE);
                Intent cameraIntent = new
Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE);
                startActivityForResult(cameraIntent, CAMERA_REQUEST);

            }
        });

    }

    public void northIndian(View view) {

        Intent i=new Intent(this,NorthActivity.class);
        startActivity(i);

    }

    public void southIndian(View view) {
        Intent i=new Intent(this,SouthActivity.class);
        startActivity(i);
    }

    public void chinese(View view) {
```

```
    Intent i=new Intent(this,ChineseActivity.class);
    startActivity(i);
  }

  public void continental(View view) {
    Intent i=new Intent(this,ContinentalActivity.class);
    startActivity(i);
  }

  private  boolean checkAndRequestPermissions() {
    int camera = ContextCompat.checkSelfPermission(this,
android.Manifest.permission.CAMERA);
    int storage = ContextCompat.checkSelfPermission(this,
android.Manifest.permission.WRITE_EXTERNAL_STORAGE);
    int loc = ContextCompat.checkSelfPermission(this,
android.Manifest.permission.READ_EXTERNAL_STORAGE);
    int loc2 = ContextCompat.checkSelfPermission(this,
android.Manifest.permission.INTERNET);
    List<String> listPermissionsNeeded = new ArrayList<>();

    if (camera != PackageManager.PERMISSION_GRANTED) {
      listPermissionsNeeded.add(android.Manifest.permission.CAMERA);
    }
    if (storage != PackageManager.PERMISSION_GRANTED) {

listPermissionsNeeded.add(android.Manifest.permission.WRITE_EXTERNAL_ST
ORAGE);
    }
    if (loc2 != PackageManager.PERMISSION_GRANTED) {

listPermissionsNeeded.add(android.Manifest.permission.READ_EXTERNAL_ST
ORAGE);
    }
    if (loc != PackageManager.PERMISSION_GRANTED) {
      listPermissionsNeeded.add(android.Manifest.permission.INTERNET);
    }
    if (!listPermissionsNeeded.isEmpty())
    {
      ActivityCompat.requestPermissions(this,listPermissionsNeeded.toArray
```

```
                (new
String[listPermissionsNeeded.size()]),REQUEST_ID_MULTIPLE_PERMISSIONS);
            return false;
        }
        return true;
    }


}


/*
//This will run if image is correctly classified




 */
```

REFERENCE

1. Android Programming for Beginners" by John Horton
2. "Android Programming in a Day! The Power Guide for Beginners in Android App Programming" by Sam Key
3. "Android Programming: Mastering Course for Beginners Quick Start to Develop Your Own App" by Mitchell Schule
4. Mastering Firebase for Android Development by Ashok Kumar S
5. A definitive Guide to firebase by H Wells
6. Building a RESTful python web service by Gaston C. Hillar
7. Python API tutorials – www.realpython.com
8. Flask tutorial for beginners – CS50 YouTube channel
9. REST API design rule book by Mark Masse
10. Deploy PyTorch to production on Heroku – Avinash S