A PROGRESS REPORT

ON

Food Recipe Generator System

Submitted by

Satyarth Ratnani (18IT444)

Kishan Thakkar (18IT458)

Shubham Patel (18IT463)

Keyur Patel (18IT464)

For Partial Fulfillment of the Requirements for Bachelor of Technology in Information Technology

GUIDED BY

Prof. Aneri Patel

May, 2022

Information Technology Department

Birla Vishvakarma Mahavidyalaya Engineering College

(An Autonomous Institution)

VallabhVidyanagar – 388120

Gujarat, INDIA

Birla Vishvakarma Mahavidyalaya Engineering College

(An Autonomous Institution)

Information Technology Department

AY: 2021-22, Semester II

CERTIFICATE

This is to certify that the project work entitled **Food recipe generator system** has been successfully carried out by **Satyarth Ratnani(18IT444), Kishan Thakkar(18IT458), Shubham Patel(18IT463), Keyur Patel(18IT464)** for the subject **Project II (4IT32)** during the academic year 2021-22, Semester-II for partial fulfilment of Bachelor of Technology in Information Technology. The work carried out during the semester is satisfactory.

Prof. Aneri Patel
IT Department
BVM

Dr. Keyur Brahmbhatt
Head, IT Department
BVM

Acknowledgement

We would like to express a special thanks to our guide Prof. Aneri Patel who gave us a golden opportunity to do this wonderful project, which also helped us in doing a lot of research and we came to know about so many new things. we are really thankful to them.

Secondly, we would like to thank Our course coordinators Dr. Zankhna Shah and Prof. Vishal Polara and also the head of IT department Dr. Keyur Brahmbhatt for guiding us for successful completion of the project.

However, it would not have been possible without the kind support and help of many individuals. We would like to extend our sincere thanks to all of them for their valuable Assistance. Also, thanks to some of our friends and family, who gave us moral support.

Abstract

People enjoy food photography because they appreciate food. Behind each meal there is a story described in a complex recipe and, unfortunately, by simply looking at a food image we do not have access to its preparation process. Therefore, we introduce an inverse cooking system that recreates cooking recipes from given food images. The system accurately detects the food item from image and show its recipe and it also provides recipe by search option, with an attractive website interface.

List of Figures

List of Tables

Table of Content

Chapter 1: Introduction

1.1 Brief overview of project

We aim to make a user aware of the various dishes which can be cooked from available set of ingredients or an input image given by a user.

Thus, systems like this never get their place in real life. We deciding to developed a Recipe Generation model by applying Neural Network on it. The tools used for developing the project are Python, React, CSS, JavaScript & Flask or Django.

- A dynamic website incorporating machine learning techniques.
- Search options:
- Search by Ingredients and name
- Search by food image

Output: self-generated recipes

It is a recipe generator system and not a recommender system!!!

1.2 Project Objective

Our objective is to create an application that will assist an end-user to explore a variety of recipes with the available ingredients in one's kitchen. The steps to be followed are:

- To explore various related datasets and perform data preprocessing in order to extract the desired attributes.
- To perform model training using Machine Learning or Deep Learning algorithms to find the best matching recipes corresponding to a given set of ingredients.
- To classify an input image of any dish and estimate its caloric value.

1.3 Project Scope

People get into a situation that they want to cook something delicious but are short on ingredients at home. Many times people see an image of a delicious looking dish, but they

don't know how to cook it. We aim to make a user aware of the various dishes which can be cooked from available set of ingredients or an input image given by a user.

1.4 Project modules

- Authentication
- preprocessing
- Recipe by Search
- Recipe by Image
- Comment
- Admin Panel

1.5 Project Hardware/Software Requirements:

1.5.1 Hardware Requirements:

- Computer Device
- RAM: 4 GB or above
- Processor: Intel i3 or higher
- Disk Space: 4 GB Minimum

1.5.2 Software Requirements

- Operating System: Microsoft Windows 7 or later, MacOS, Linux
- Programming Language: Python 3.x up to Python 3.7.x
- Software: PyCharm or any other IDE that supports Python 3

1.6 Technologies

- Back-end: Python, Django framework
- Front end: HTML, CSS, JavaScript, Bootstrap
- Database: MySQL
- Dataset creation : Image assistant batch extension

Chapter 2: Literature review

2.1 Extract Features, Visualize Filters and Feature Maps in VGG16 Models

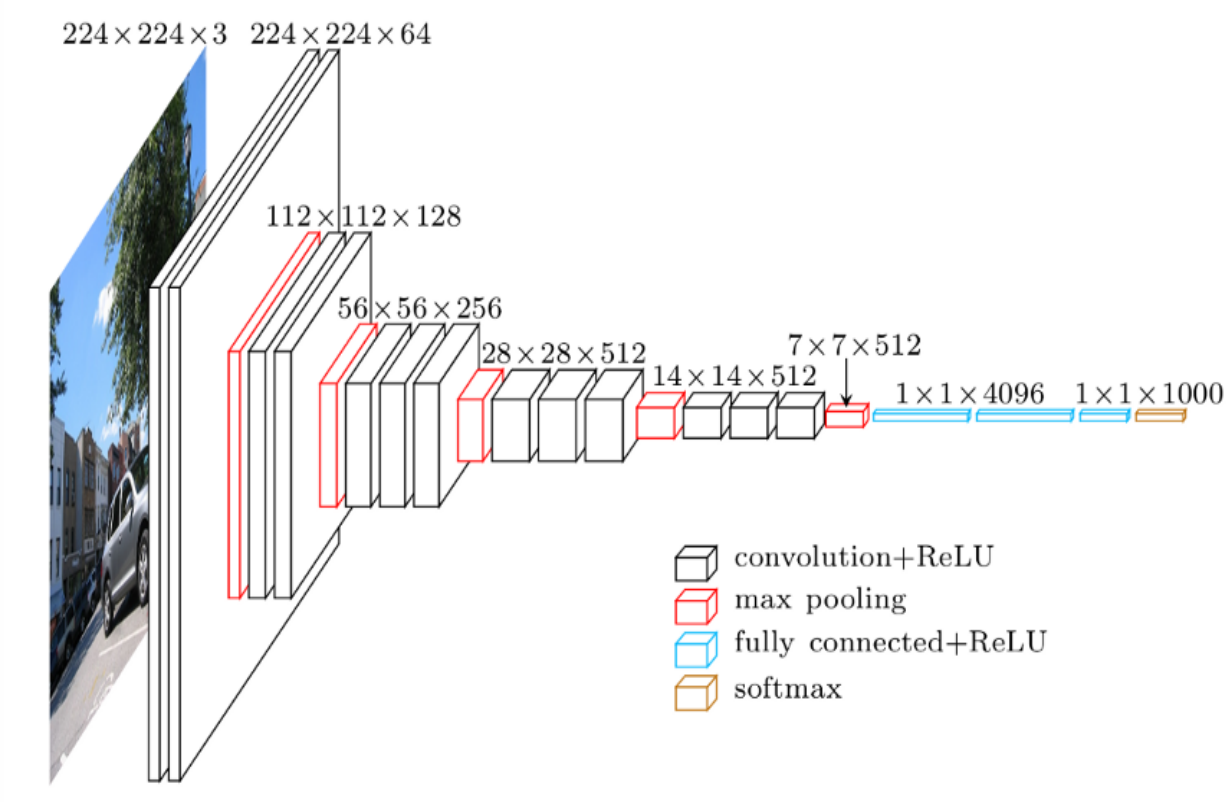


Figure 1. VGG16 Architecture

Keras provides a set of deep learning models that are made available alongside pre-trained weights on ImageNet dataset. These models can be used for prediction, feature extraction, and fine-tuning. Here I'm going to discuss how to extract features, visualize filters and feature maps for the pretrained models VGG16 and VGG19 for a given image.

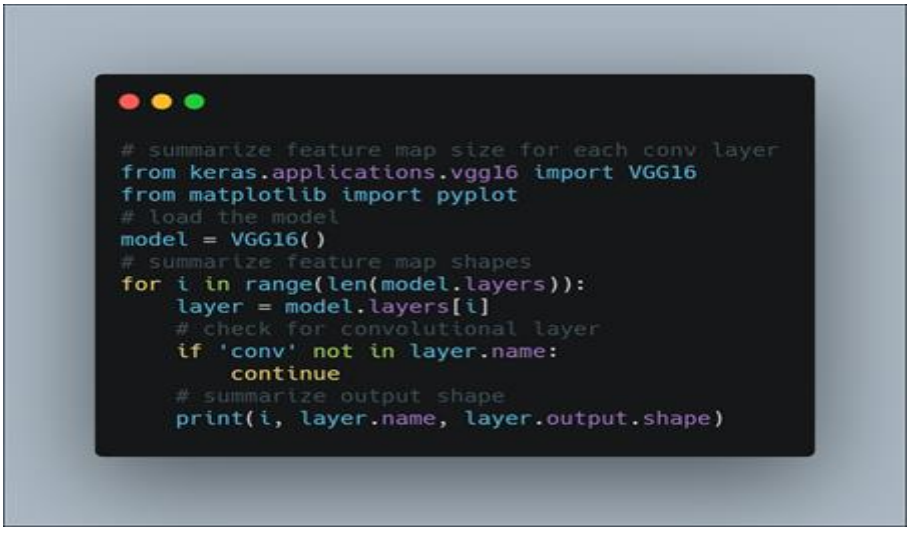
2.1.1 Extract Features with VGG16

Here we first import the VGG16 model from TensorFlow eras. The image module is imported to preprocess the image object and the preprocess input module is imported to scale pixel values appropriately for the VGG16 model. The NumPy module is imported for array-processing. Then the VGG16 model is loaded with the pretrained weights for the ImageNet

dataset. VGG16 model is a series of convolutional layers followed by one or a few dense (or fully connected) layers. Include top lets you select if you want the final dense layers or not. False indicates that the final dense layers are excluded when loading the model. From the input layer to the last max pooling layer (labeled by 7 x 7 x 512) is regarded as feature extraction part of the model, while the rest of the network is regarded as classification part of the model. After defining the model, we need to load the input image with the size expected by the model, in this case, 224×224. Next, the image PIL object needs to be converted to a NumPy array of pixel data and expanded from a 3D array to a 4D array with the dimensions of [*samples, rows, cols, channels*], where we only have one sample. The pixel values then need to be scaled appropriately for the VGG model. We are now ready to get the features.

2.1.2 Summarize Feature Map Size for Each Conv Layer

The activation maps, called feature maps, capture the result of applying the filters to input, such as the input image or another feature map. The idea of visualizing a feature map for a specific input image would be to understand what features of the input are detected or preserved in the feature maps. The expectation would be that the feature maps close to the input detect small or fine-grained detail, whereas feature maps close to the output of the model capture more general features. In order to explore the visualization of feature maps, we need input for the VGG16 model that can be used to create activations.



```
# summarize feature map size for each conv layer
from keras.applications.vgg16 import VGG16
from matplotlib import pyplot
# load the model
model = VGG16()
# summarize feature map shapes
for i in range(len(model.layers)):
    layer = model.layers[i]
    # check for convolutional layer
    if 'conv' not in layer.name:
        continue
    # summarize output shape
    print(i, layer.name, layer.output.shape)
```

Figure 2. VGG16 model

We need a clearer idea of the shape of the feature maps output by each of the convolutional layers and the layer index number. So we enumerate all layers in the model and print the output size or feature map size for each convolutional layer as well as the layer index in the model.

2.1.3 Train MobileNetV2 On a Custom Dataset

MobileNetV2 is a classification model developed by Google. It provides real-time classification capabilities under computing constraints in devices like smartphones. This implementation leverages transfer learning from ImageNet to your dataset. In this post, we will walk through how you can train MobileNetV2 to recognize image classification data for your custom use case.

We use a public flowers classification dataset for the purpose of this tutorial. However, you can import your own data into Roboflow and export it to train MobileNetV2 to fit your own needs. The MobileNetV2 notebook used for this tutorial can be downloaded here**.**

Thanks to the TensorFlow Team for publishing the base notebook that formed the foundation of our notebook.

To apply transfer learning to MobileNetV2, we take the following steps:

- Download data using Roboflow and convert it into a TensorflowImageFolder Format
- Load the pretrained model and stack the classification layers on top
- Train & Evaluate the model
- Fine Tune the model to increase accuracy after convergence
- Run an inference on a Sample Image

2.2 Innovations with MobileNetV2

The MobileNetV2 architecture utilizes an inverted residual structure where the input and output of the residual blocks are thin bottleneck layers. It also uses lightweight convolutions to filter features in the expansion layer. Finally, it removes non-linearities in the narrow layers. The overall architecture looks something like this:

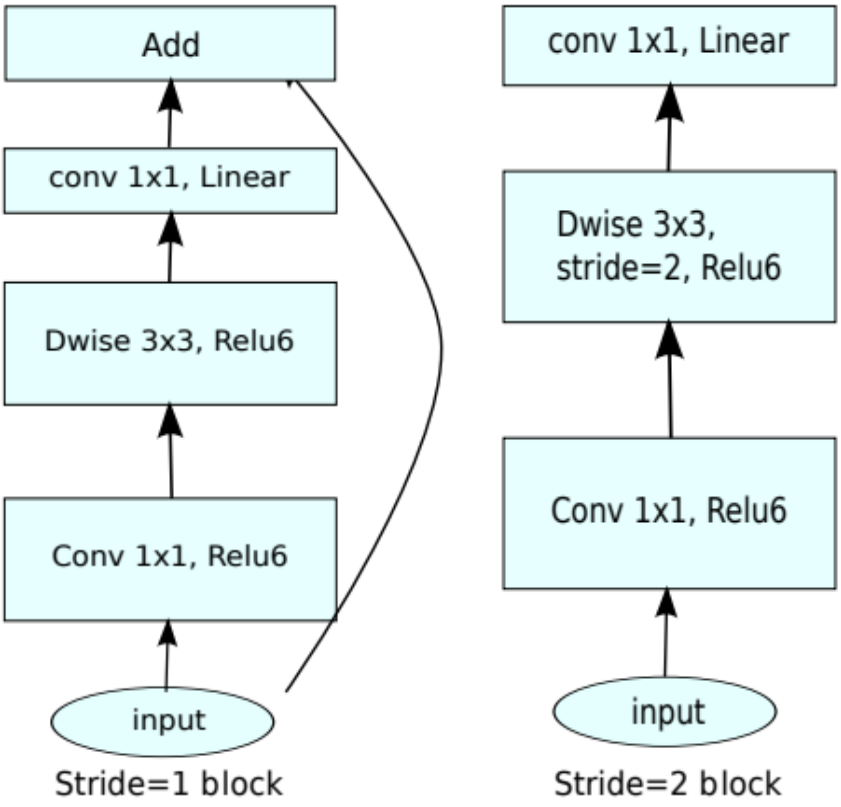


Figure 3. MobileNetV2 Flow chart

As a result, MobileNetV2 outperforms MobileNetV1 with higher accuracies and lower latencies:

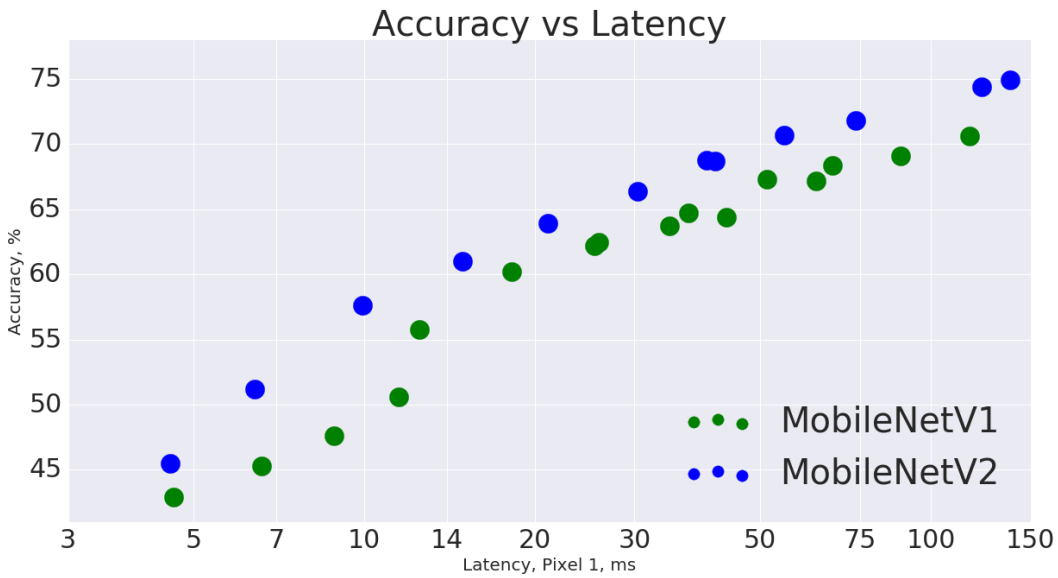


Figure 4. Accuracy vs Latency

MobileNetV2 is a powerful classification model that is able to reach state-of-the-art performance through transfer learning. In this tutorial we were able to:

- Download images to train MobileNetV2
- Construct the MobileNetV2 model
- Train the MobileNetV2 model for Binary Classification
- Improve performance post-convergence through fine tuning
- Run an inference on a sample image

2.3 Recipe Recommendation System

Recipe recommendation system that makes use of images of ingredients to recommend a recipe to a person who doesn't know anything about the contents and its proportion in a particular recipe. This system is also beneficial for health-conscious people. With the changing living manners, diet habits are changed and work load has increased which resulted in various diseases, such as diabetes, BP, problems related to heart and so on. All these diseases can be controlled by avoiding uneven and unhealthy food. So it's important to understand that what is the proper diet and in how much quantity it should be taken. Our main aim is to recommend recipes to maintain their health for people with disease and without disease which will satisfy

the needs of user. To give the recommendations, recommended system uses the user's profile, their favourite food and details of that food. Clients from various countries, belonging to various cultures are contributing in terms of a large number of new recipes on the web all over the world. Every recipe contains of so many distinct elements. Thus, user might not be able to identify all the ingredients or contents. In this paper we propose a recommendation system for recipe using Convolutional Neural Network (CNN) which is used for supervised learning in order to analyse the data. Recommended Recipe contains diverse ingredients, cooking procedure, categories and so on. The recipe which includes the ingredients mentioned by user with proper nutritional values will be a good recommendation.

2.3.1 System architecture

Following diagram is our system's architecture diagram:

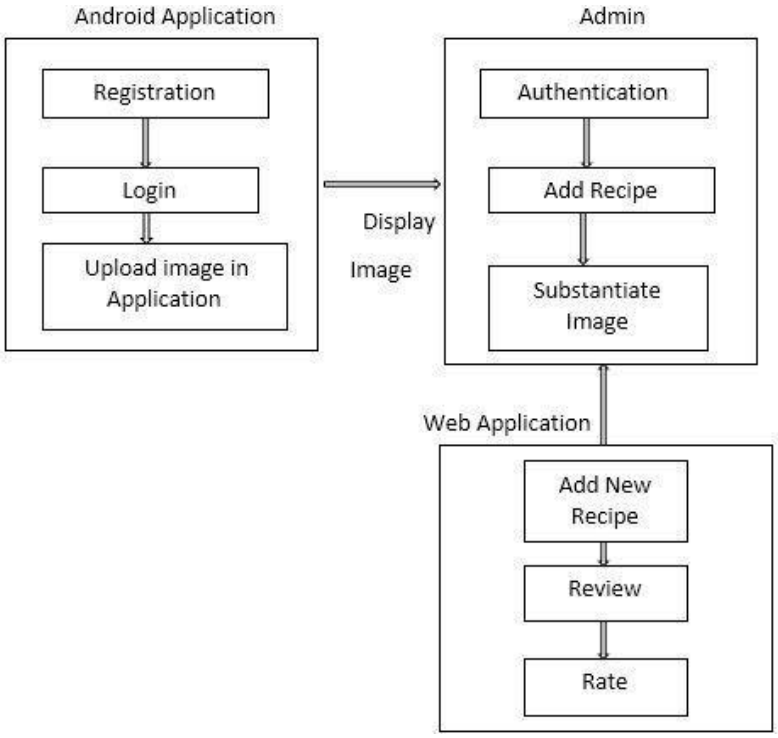


Figure 5. system architecture

In System architecture Admin and user have to register the application. As the registration is successful, user can login the application. Now the user will use this android application for detecting the images of ingredients and upload them. The uploaded images are compared with images in database. For developing this object detection app we are going

to use TensorFlow API. TensorFlow is an API used for object detection which uses pre-trained models. The matching image will be retrieved and displayed to user.

This gathered search information is passed to recommended module of proposed system. In this recommendation actual recommended recipe for user is present. Then this recipe will show on user profile as recommended recipe with its nutritional values. By looking at the nutrition details, user can decide his/her diet regarding their health issues.

2.3.2 Research Papers

https://drive.google.com/file/d/1hjfRhq2-lsC-nvK9ZwFA3A1P_x1uSCgX/view?usp=drivesdk

https://drive.google.com/file/d/1hlID4y4fztegTlr9oVIsugWuhM8XEUgz/view?usp=drivesdk

2.4 How to connect to an API in JavaScript?

An API or Application Programming Interface is an intermediary which carries request/response data between the endpoints of a channel. We can visualize an analogy of API to that of a waiter in a restaurant. A typical waiter in a restaurant would welcome you and ask for your order. He/She confirms the same and carries this message and posts it to the order queue/kitchen. As soon as your meal is ready, he/she also retrieves it from the kitchen to your table. As such, in a typical scenario, when a client endpoint requests a resource from a resource server endpoint, this request is carried via the API to the server. There are a various relevant information that are carried by the APIs which conform to certain specification of schema such as that provided by OpenAPI, GraphQL etc. This information may include endpoints URL, operations (GET, POST, PUT etc.), authentication methods, tokens, license and other operational parameters. APIs most commonly follow the JSON and XML format as its key mode of exchanging request/response while some also adhere to the YAM format.

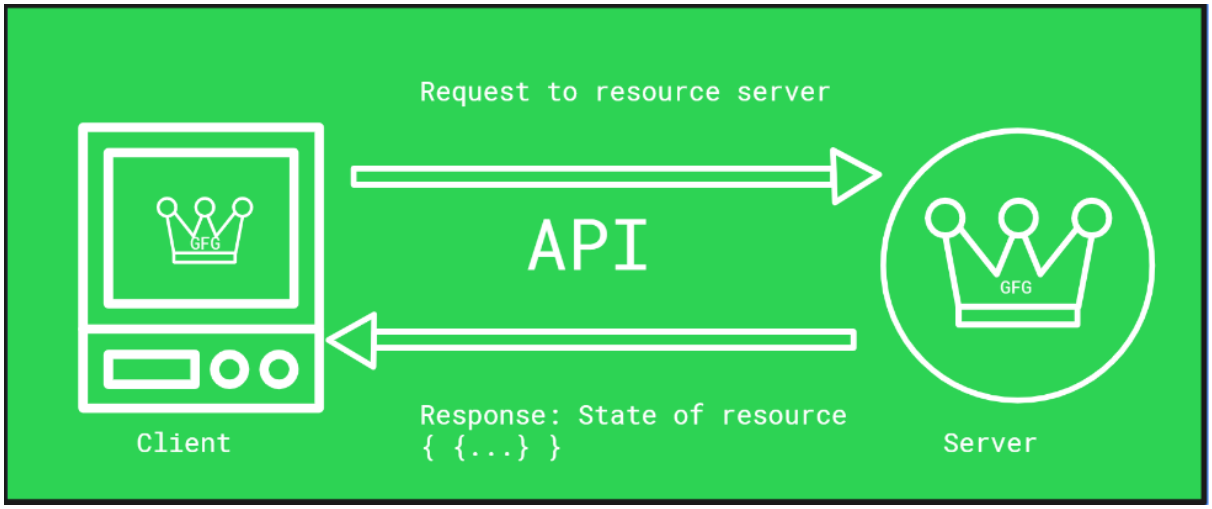


Figure 6. API request and response

Mostly the response generated by a resource server is a JSON schema which carries the state information of the resource requested at the other endpoint. As such these APIs have been named REST APIs where REST stands for Representational State Transfer. The state of the resources can be affected by the API operations. It should also be noted that there are System APIs as well which are used by the operating system to access kernel functions. A common example would include the Win32 API which is a windows platform API and acts as a bridge for system level operations such as File/Folder select, button styling etc. Most programming languages which have GUI libraries are wrapped upon this layer.

2.4.1 Sample Request of an API(Google Geolocation API) :

{

"homeMobileCountryCode": 310,

"homeMobileNetworkCode": 311,

"radioType": "gsm",

"carrier": "airtel",

"considerIp": "true",

```
"cellTowers": [
    {
        "cellId": 22,
        "locationAreaCode": 115,
        "mobileCountryCode": 310,
        "mobileNetworkCode": 311,
        "age": 0,
        "signalStrength": -40,
        "timingAdvance": 12
      }
],
"wifiAccessPoints": [
    {
    "macAddress": "00:25:9e:ff:jc:wc",
    "signalStrength": -33,
    "age": 0,
    "channel": 12,
    "signalToNoiseRatio": 0
    }
]
```

```
}
```

2.4.2 Sample response

```
{
   "location": {
     "lat": 41.1,
     "lng": -0.1
   },
   "accuracy": 1200.2
}
```

The schema, usage pricing, etc. of an API is dependent on the organization providing the API. There are freely available APIs such as Page CDN API, as well pay as you go pricing model based APIs such as Street View Static API. APIs enable a client/programmer to use the infrastructure and cloud services of an organization to get access to various resources over the internet. An API usually requires an API key(unique) along with a few optional credentials for it to authenticate a resource request made by a client. Web Programmers often rely on APIs for pulling off various awesome tricks such as displaying filtered tweets via twitter API on their homepage, converting HTML to pdf via HTML2PDF API, etc. For science students and enthusiasts, various APIs such as NASA APIs(Exoplanet, Insight, etc) provides content on demand. Mobile app developers also use APIs to a great extent such as weather APIs, Geolocation, Google Analytics API, etc.

2.4.3 Connect to an API using JavaScript

To make API calls using JavaScript, a reference can be made under the <script> tag to the

JavaScript library which contains functions and other configuration parameters pertaining to the API should be made. A good API always maintains appropriate documentation to its functions and parameters. In most cases, this js library is made available via the Content Delivery Network(CDN) in order to be responsive. JavaScript contains functions for serializing and de-serializing a JSON object and thus handling JSON responses and traversing/parsing the response is also managed within the same script.

The below snippet shows the simplest example of using google visualization API to construct a chart from data present in google sheets(spreadsheet).

A reference to the JS library containing necessary functions is made as follows:

```
<script type="text/javascript"
      src="https://www.gstatic.com/charts/loader.js">
</script>

google.load('visualization', '1',
          {
            'packages':['corechart', 'table', 'geomap']
          }
   );
   google.setOnLoadCallback(drawGID);
      function drawGID() {
        //var queryString = encodeURIComponent('SELECT A, B LIMIT 5');
```

```
    var query =
new google.visualization.Query(
    'https://docs.google.com/spreadsheets/d/spreadsheetId/gviz/tq?range=');
    query.send(handleQueryResponse);
  }
  var resp;
  function handleQueryResponse(response) {
    if (response.isError()) {
      alert('Error in query: ' + response.getMessage() +
          ' ' + response.getDetailedMessage());
      return;
    }
    var data = response.getDataTable();
    resp = response;
    var chart =
 new google.visualization.LineChart(
    document.getElementById('any_div_or_container'));
    chart.draw(data, { height: 400, curveType: 'function',
       legend: { position: 'bottom' }});
  }
```

The above code contains a callback function that fires when the window is loaded. A query string containing the spreadsheet ID and other parameters is passed to the server.

Here spreadsheetId needs to be replaced by the spreadsheet id of the concerned spreadsheet. The ‘any_div_or_container’ string is to be replaced by the DOM element on which we wish to display the results in our page. The response handler analyses the response and checks the content type after which it parses to the data to produce the desired output. The above code is run with a sample spreadsheet which generates a JSON response as shown below:

gvjs_rl {wva: "0.6", qX: "ok", hv: Array(0), Sw: Array(0), O2: "1651350667", …}

wva: "0.6"

qX: "ok"

hv: []

Sw: []

O2: "1651350667"

R: gvjs_L

$p: null

Ff: Array(2)

0: {id: "A", label: "", type: "datetime", pattern: "M/d/yyyy H:mm:ss", p: {…}}

1: {id: "B", label: "", type: "number", pattern: "General"}

length: 2

__proto__: Array(0)

eg: (98) [{…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, ...]

Fr: null

cache: []

version: "0.6"

__proto__: gvjs_$k

__proto__: Object

Chapter 3: System Analysis & Design

3.1 Comparison of Existing Applications with your Project with merits and demerits:

Some existing systems and their pros and cons are mentioned below

3.1.1. All Recipes

Taking the top position is All Recipes, a top-tier recipe website that is estimated to have over 25 million visitors each month. With a crisp and easily navigable site and backing of skilled and experienced culinary masters, this platform furnishes its visitors with thousands of recipes and meal preparation videos. It also features comprehensive how-to-do guides on absolutely everything food-related topics. The site also permits users to create profiles and share their secret recipes with others.

Pros

- Gives you access to an extensive recipe database
- Boasts advanced searching filters for targeted searches
- Provides some recipes with video instructions

Cons

- Recipes load-speeds are slow since they contain loads of information

3.1.2 The Food Network

Another outstanding player in the culinary game is The Food Network. Established in 1993, the network boasts a religious following since it reaches its followers via several mediums, including the website, television, radio, magazines, and podcasts. The site boasts an outstanding database of recipes and videos, as well as clips and episodes of past televised Food Network shows.

Pros

- Since it has been around for quite some time, it is one of the most trusted sources of culinary information
- Has a backing of experienced and talented gourmet experts
- Incorporates classic and modern recipes

Cons

- Loading speeds can be slower due to the tons of information on this website
- Some complain that the site doesn't update their content regularly

3.1.3 Yummly

Another excellent website to get top-notch recipes is Yummly. This platform boasts an appealing and easy to navigate interface, as well as plenty of filters to help you navigate easily in the database, which boasts over 2 million recipes. The site also features fantastic recipe videos that are often displayed on social media platforms such as Snapchat and Facebook. The site allows users with an account to save recipes, share their creations, and make shopping lists.

Pros

- Features outstanding videos
- Boasts a plethora of different filter options
- Clean, fun and easy to use website
- Ability to create meal plans based on diet restriction and allergies

Cons

- You need to sign up to share recipes
- Filters are only visible once you sign up

3.1.4 Epicurious

Epicurious was launched in 1998, and it has been providing top-quality recipes since. However, unlike other recipe websites that brag of the number of visitors they have, this platform takes pride in the quality of recipes and food-related content it provides. Don't expect to find millions of recipes on this site, but a handful of handpicked and trusted recipes. The platform sources filtered and tested meal preparation techniques from its partners Bon Appetit and Gourmet (RIP) magazines. They also provide instructions on some culinary basics and tips.

Pros

- Handpicked recipes from renowned food magazines and individual chefs
- Allows users to upload their recipes
- Emphasizes on quality rather than quantity

Cons

- Fewer recipes when compared to other famous recipe websites

These are some famous recipe search websites. Our website will provide outstanding feature that is Upload an image of food item and get the recipe.

We are using Edamam recipe API which contains large range of recipes. And it takes very less time to fetch the recipe so user does not have to wait to load their favorite recipes.

3.2 Project Feasibility Study:

Feasibility study is an analysis of the viability idea. The studies provide thorough analysis of the system. The outcomes of the feasibility studies will indicate whether or not to develop the system.

3.2.1 Economic Feasibility:

The project is economically feasible as the only cost involved is having a computer with the minimum requirements mentioned earlier. For the users to access the application, the only cost involved will be in getting access to the Internet.

3.2.2 Technical Feasibility:

People want the procedure and ingredients of recipes; we are creating Recipe Generator System in which people would find the recipe by simply giving the photo of that recipe or name of that recipe in our system and system will give them out the procedure and ingredients of recipes, apart from that our system also suggest some videos of that recipe of YouTube podium in our system. It saves lots of time for every individual and one prominent thing is that you can learn the recipe procedure without getting ads and any kind of promotional thing which can make you tedious.

If we talk about requirements for a project, then we must have a system or website and also a photo or name of the recipe, which you want about. You just have to put the image into our system and our system will give you a recipe description .

The tools that will use for this system are:

Language: Python 3

Front-end: HTML, CSS, React-JS

Django will be used as a back-end framework

Database: SQLite

Library: TensorFlow, Keras

3.2.3 Operational Feasibility:

This feasibility study mainly concerned whether this system will be used if it is developed and implemented. If this system will meet the requirements and needs of customers, it can be proposed to them to be used in the future

3.3 Detailed Modules Description

- **Authentication**

  An authentication module is a plug-in that collects user information such as a user ID and password, and compares the information against entries in a database. User can create account, Login to the account, change the password.

- **Recipe by Search**

  User can search the recipe by name of the food item. It will use recipe search API to fetch the recipe and show it to our website.

- **Recipe by Image**

  In this module user can upload an image of food item. Name of the item is predicted by trained machine learning model. Based on that recipe will be fetched from API and show to user.

- **Comment**

  In this module user can add comment, share their opinion. User can read others comment also.

- **Admin Panel**

  This module allows administrator to manage the system. Admin can manage user profile, delete particular user. Admin can also manage the comments done by users.

3.4 Timeline chart

Figure 7. Timeline chart

3.5 Project SRS

3.5.1 Use-Case Diagram

The use case diagram starts with the doctor logging in and then entering the symptoms values.

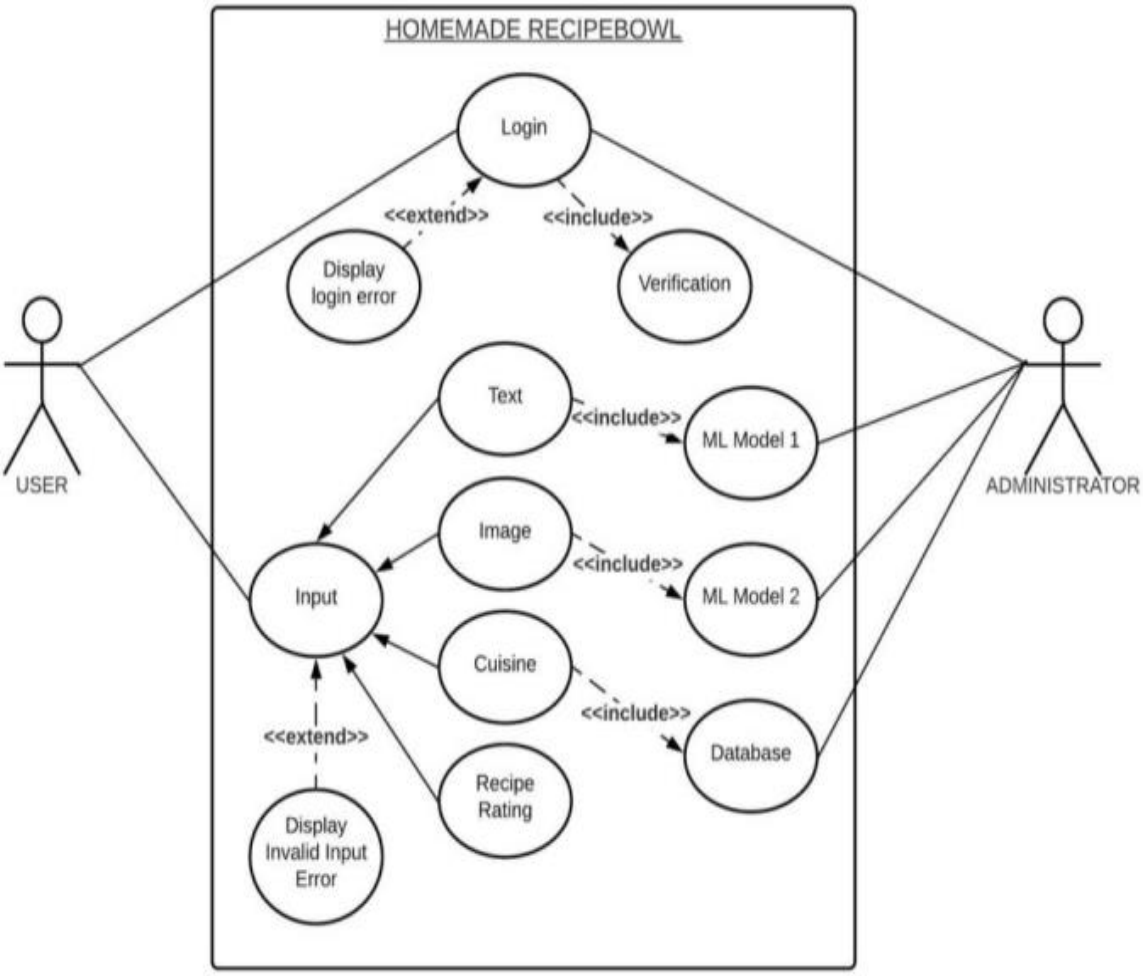


Figure 8. Use case diagram

3.5.2 Class Diagram

Shows the various components and interactions of various classes and their associated cardinality.

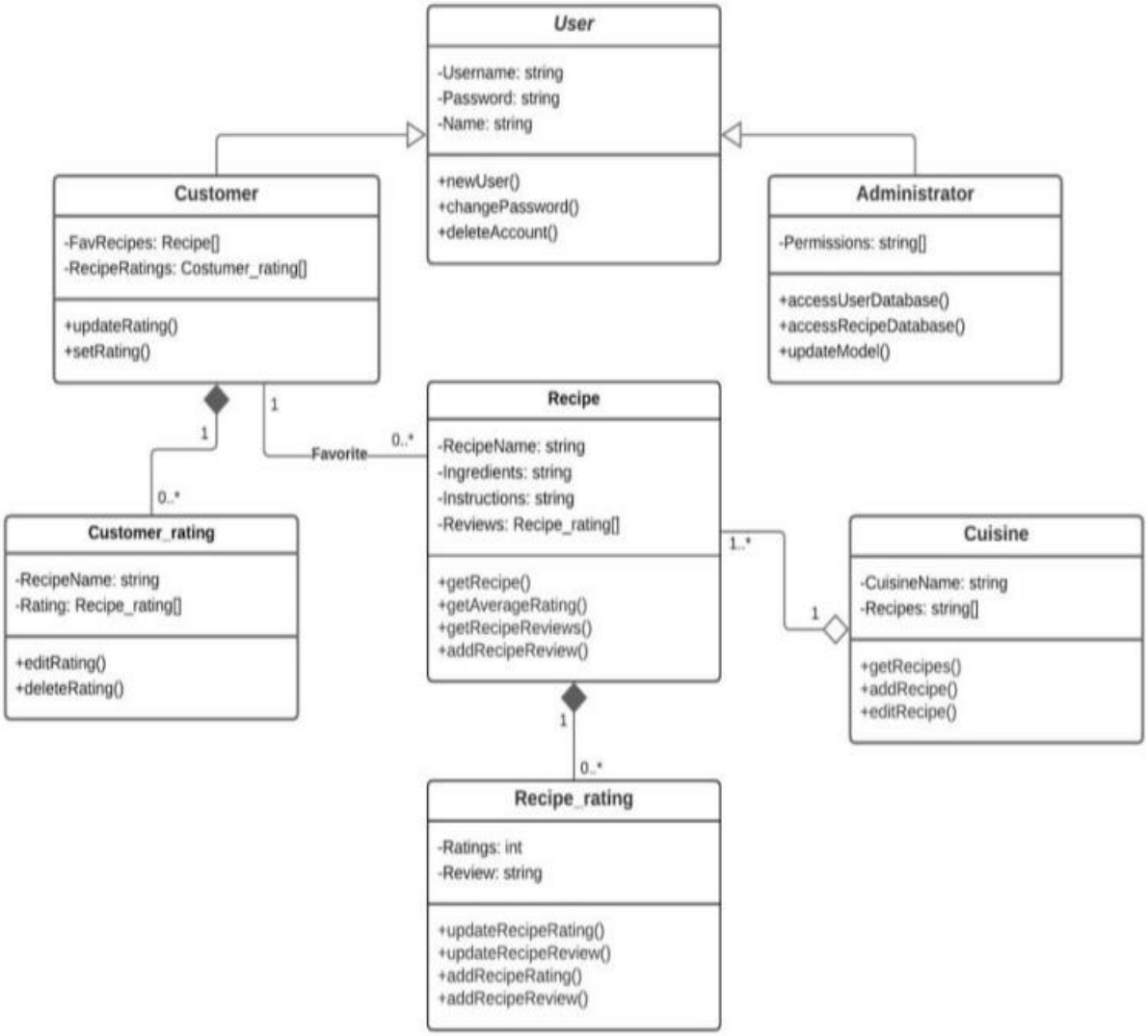


Figure 9. Class diagram

3.5.3 Sequence Diagram

A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together.

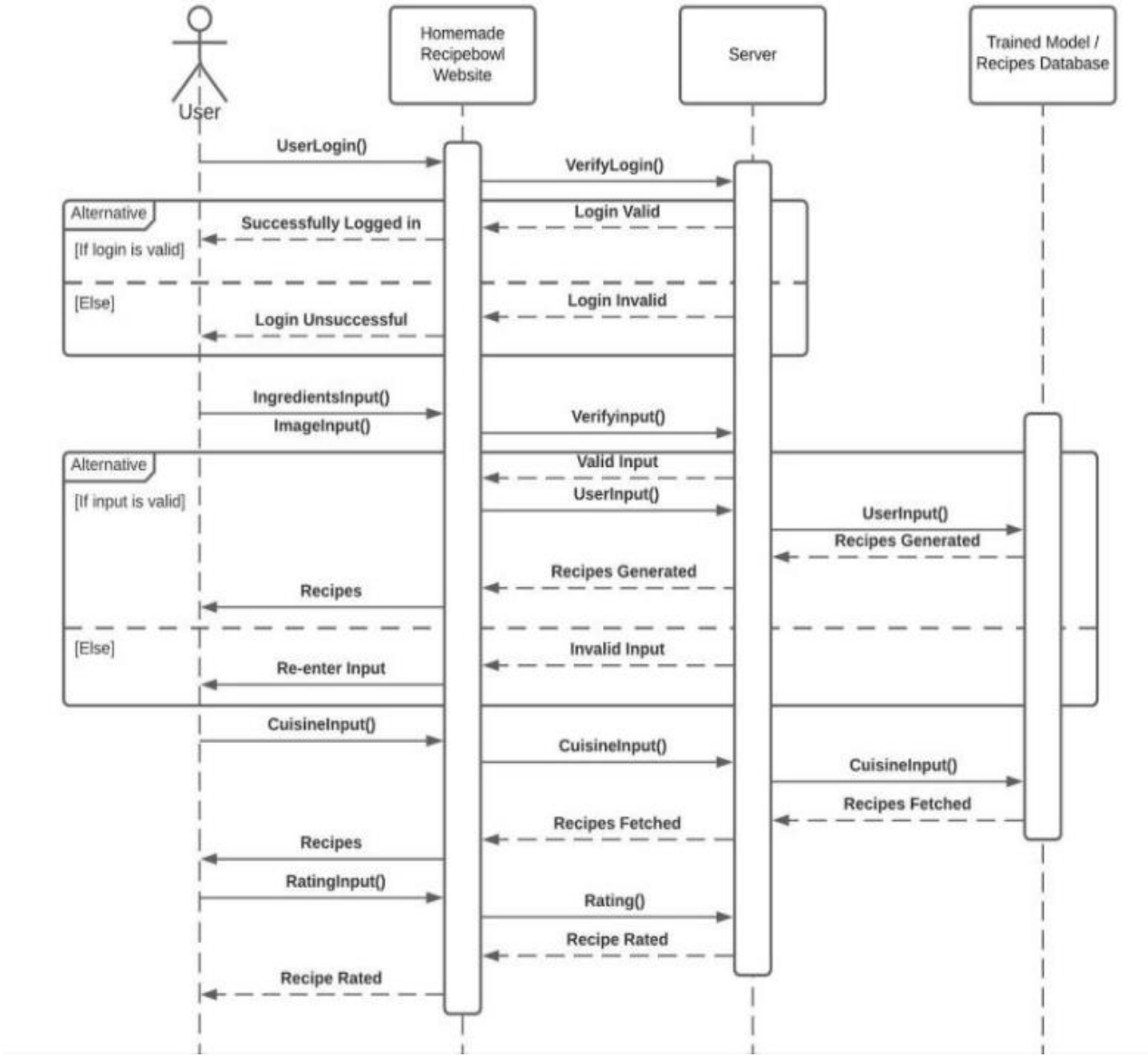


Figure 10. Sequence diagram

3.5.4 State Diagram

They are behavioral diagrams used to represent the conditions of the system or part of the system at finite instances of time.

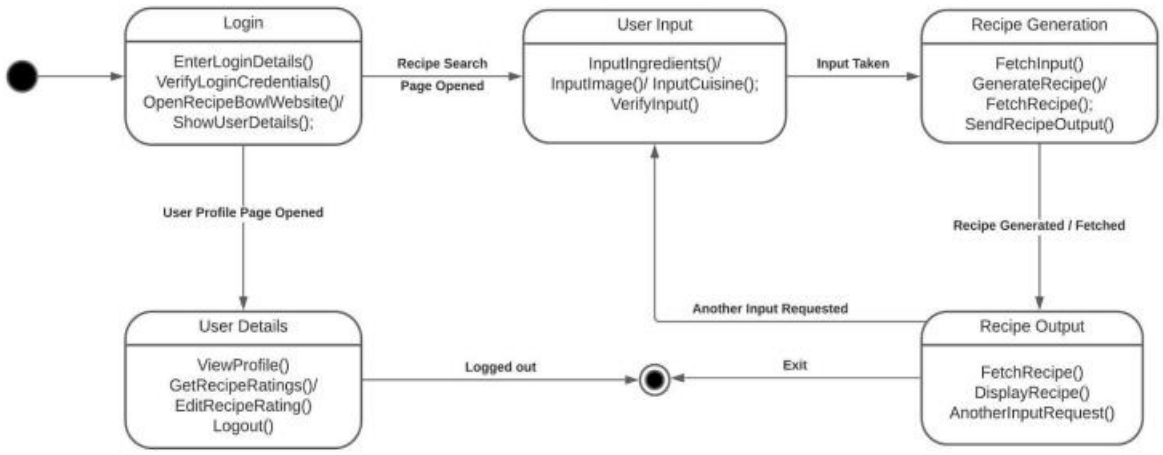


Figure 11. State diagram

3.5.5 Activity Diagram

An activity diagram is a behavioral diagram i.e. it depicts the behavior of a system

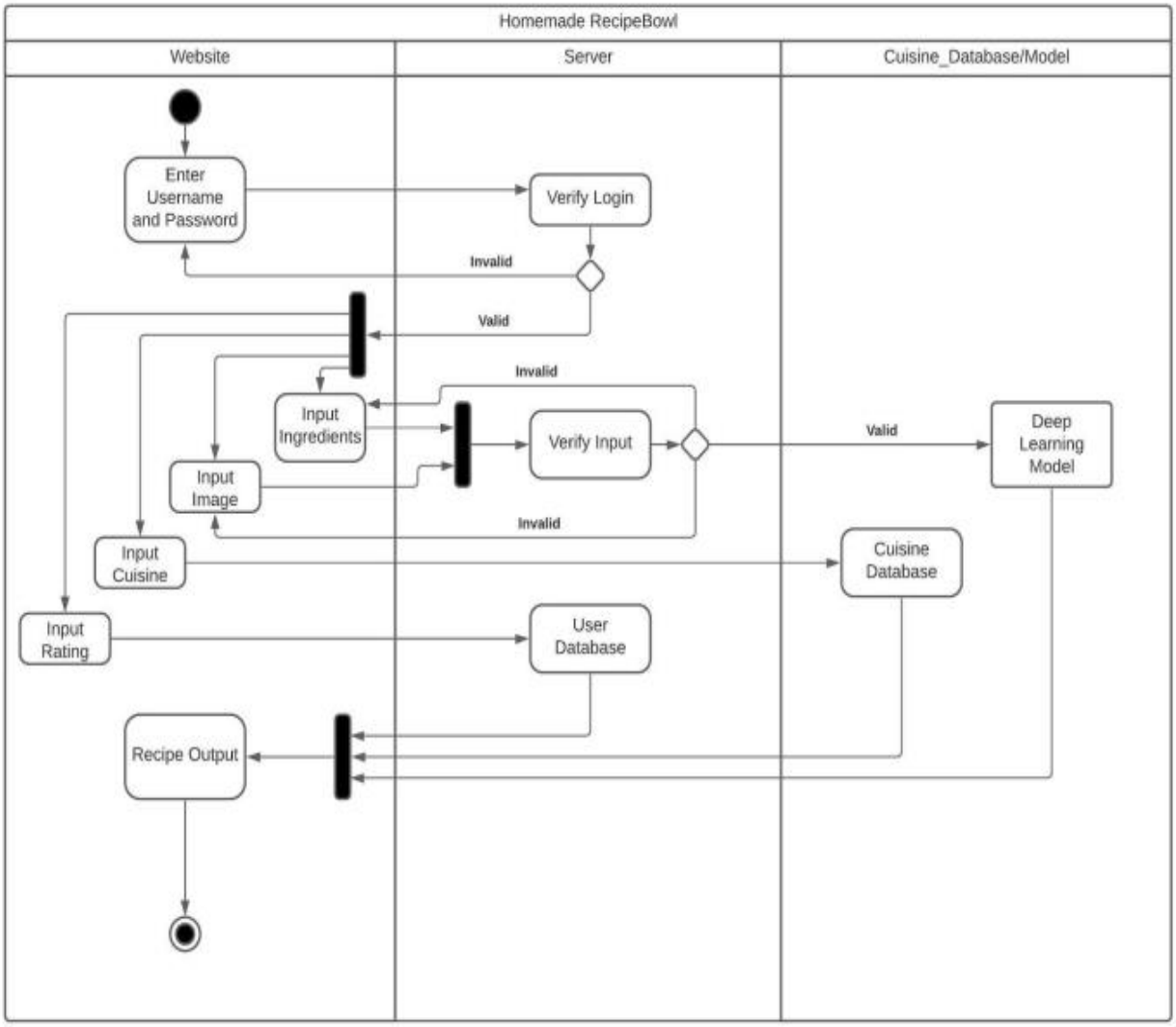


Figure 12. Activity diagram

3.6 Data Dictionary

3.6.1 User database

Table Name: Login table

Description: To Store the admin and user login details.

SrNo.	Name	Datatype	Constraint	Description
1	User_ID	Varchar(10)	Primary Key	To store unique user id
2	User_name	Varchar(10)	Not null	To store the user name
3	Email_ID	Varchar(10)	Not null	To store the user email
4	Password	Varchar(10)	Not null	To Store the Password

Table 1. User database

3.6.2 Comment database

Table Name: comment table

Description: To Store user's comments.

SrNo.	Name	Datatype	Constraint	Description
1	User_ID	Varchar(10)	Primary Key	To store unique user id
2	User_name	Varchar(10)	Not null	To store the user name
3	Comment	Varchar(10)	Not null	To store the comment

Table 2. Comment database

Chapter 4: Implementation and Testing

4.1 User Interface and Snapshot

4.1.1 Dataset Creation

First we did classification, training and testing on food 101 dataset which is open source dataset on Kaggle.

Later as we instructed our project guide, we Created our own dataset of Indian food items which contains 50 food items and in each we have 150 images of that food. So we created a Indian food dataset with 7500 images.

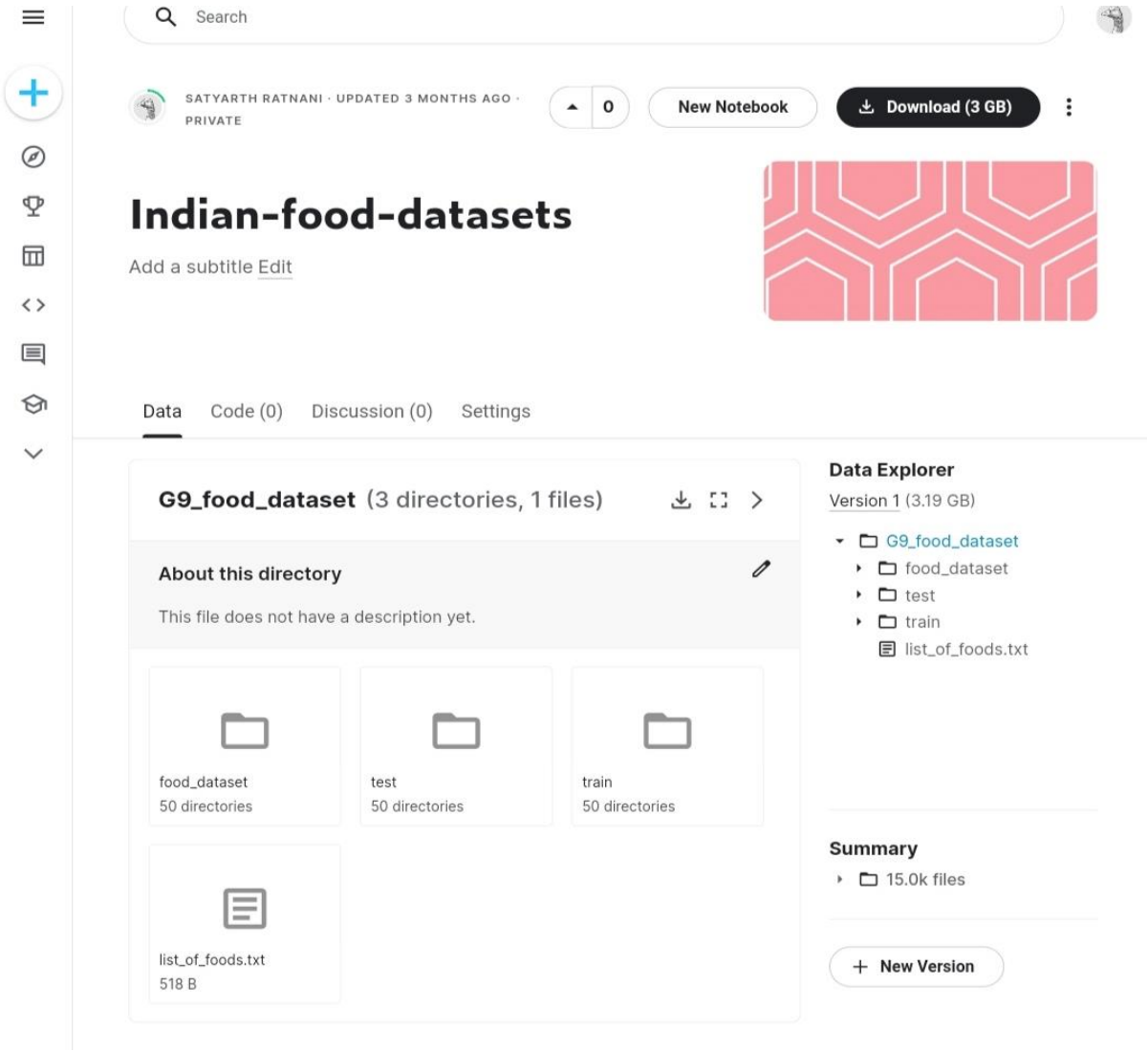


Figure 13. Indian food dataset

4.1.2 Training model

For training we are using 70% images to train set.

We trained our model using various pre trained models like mobilenetV2, InceptionV3, Resnet50.

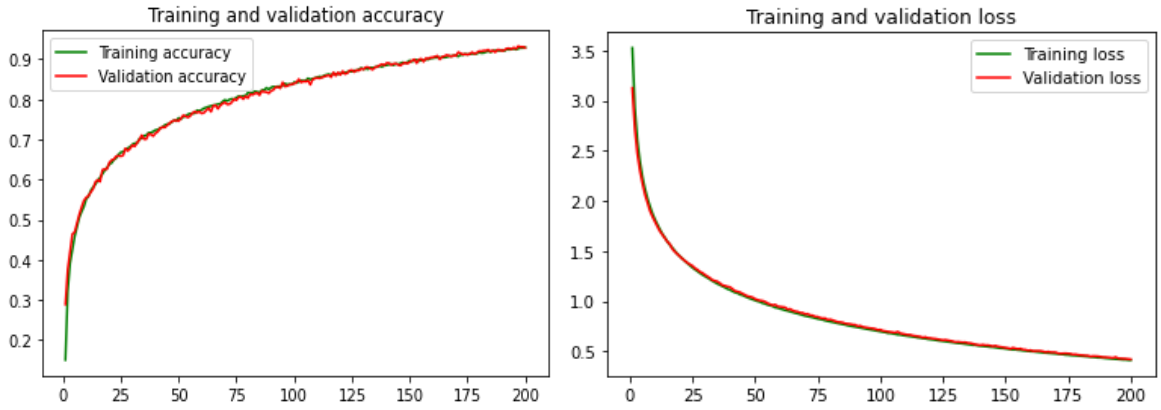


Figure 14. Accuracy chart

4.1.3 Testing model

For testing we used 30% images from dataset. By comparing all trained models we got maximum accuracy using mobilenetV2 which is around 92%.

```
Epoch 195/200
469/469 [==============================] - 3s 6ms/step - loss: 0.4244 - acc: 0.9264 - val_loss: 0.4317 - val_acc: 0.9289
Epoch 196/200
469/469 [==============================] - 3s 5ms/step - loss: 0.4206 - acc: 0.9240 - val_loss: 0.4305 - val_acc: 0.9258
Epoch 197/200
469/469 [==============================] - 3s 6ms/step - loss: 0.4178 - acc: 0.9264 - val_loss: 0.4282 - val_acc: 0.9324
Epoch 198/200
469/469 [==============================] - 3s 6ms/step - loss: 0.4181 - acc: 0.9281 - val_loss: 0.4276 - val_acc: 0.9302
Epoch 199/200
469/469 [==============================] - 3s 5ms/step - loss: 0.4144 - acc: 0.9280 - val_loss: 0.4246 - val_acc: 0.9307
Epoch 200/200
469/469 [==============================] - 3s 6ms/step - loss: 0.4134 - acc: 0.9293 - val_loss: 0.4219 - val_acc: 0.9284
```

Figure 15. Testing model

4.1.4 Prediction results

chhole_bhatore

alu_matar

mango_lassi

french_fries

Figure 16. Food prediction results

4.1.5 Website Interface

- **Login Page**

Whenever user open our website , they get prompted with interactive user login-page.

In which they have to first login by Username and Password if user have already created account on it, otherwise first they need to create an account.

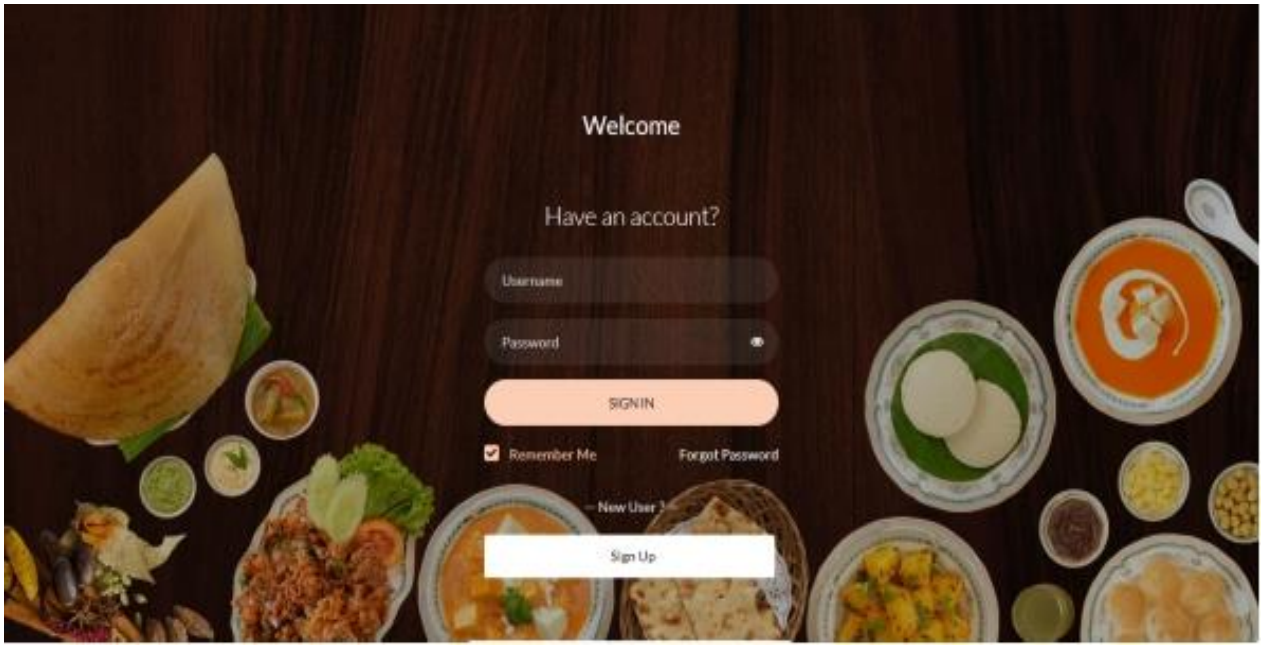


Figure 17. Login page

- **Forgot password**

If user forget the password then he can click on forgot password option. User have to provide his email id and he will get password through mail.

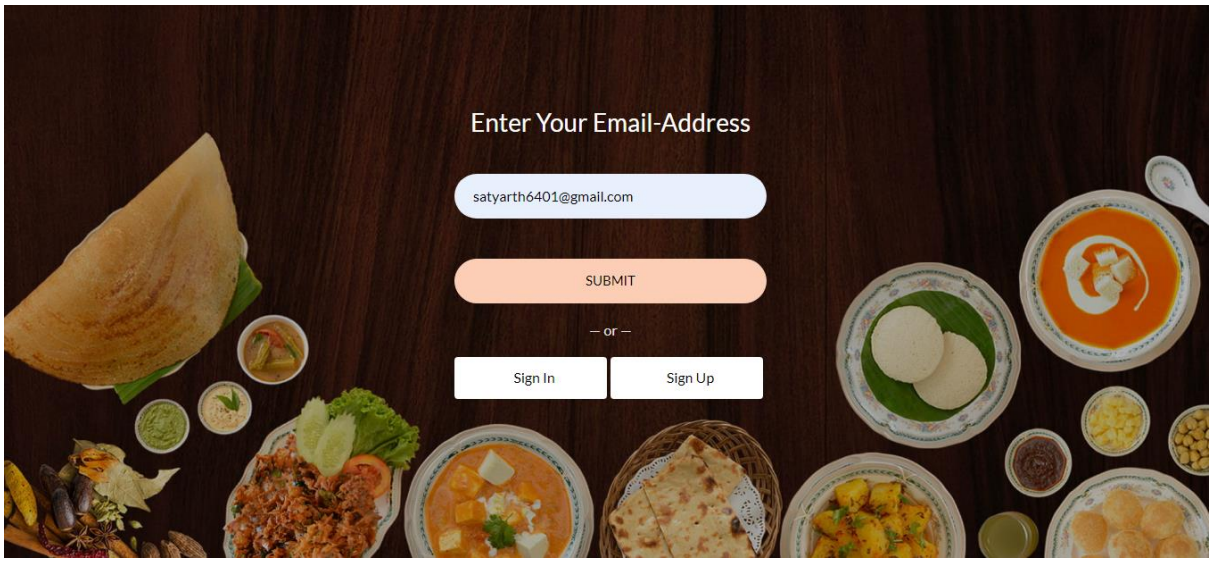


Figure 18. Forgot password page

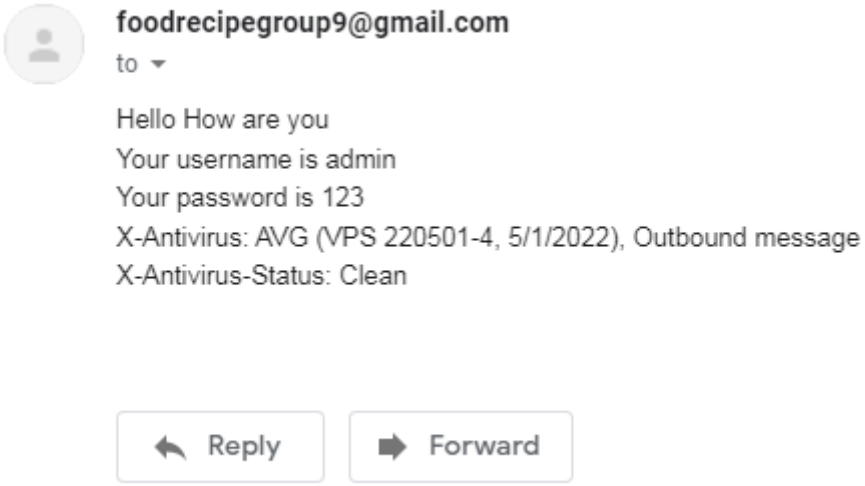


Figure 19. Password mail

- **Home Page**

After finishing authentication or login user is landed to easy to use and Beautiful Home Page, there they find By Image and By Search option to proceed further

Figure 20. Home page

- **Food detection**

After homepage individual able to find page, where two options are available,

1) Upload File: That means user have to upload the photo of cuisine that they want find recipe about and name of that recipe.

2) Detect Food: By clicking this user become able to find the recipe name and below that page our system also shows the recommendation videos of the same cuisine, by using that you would become able to find recipe making procedure.

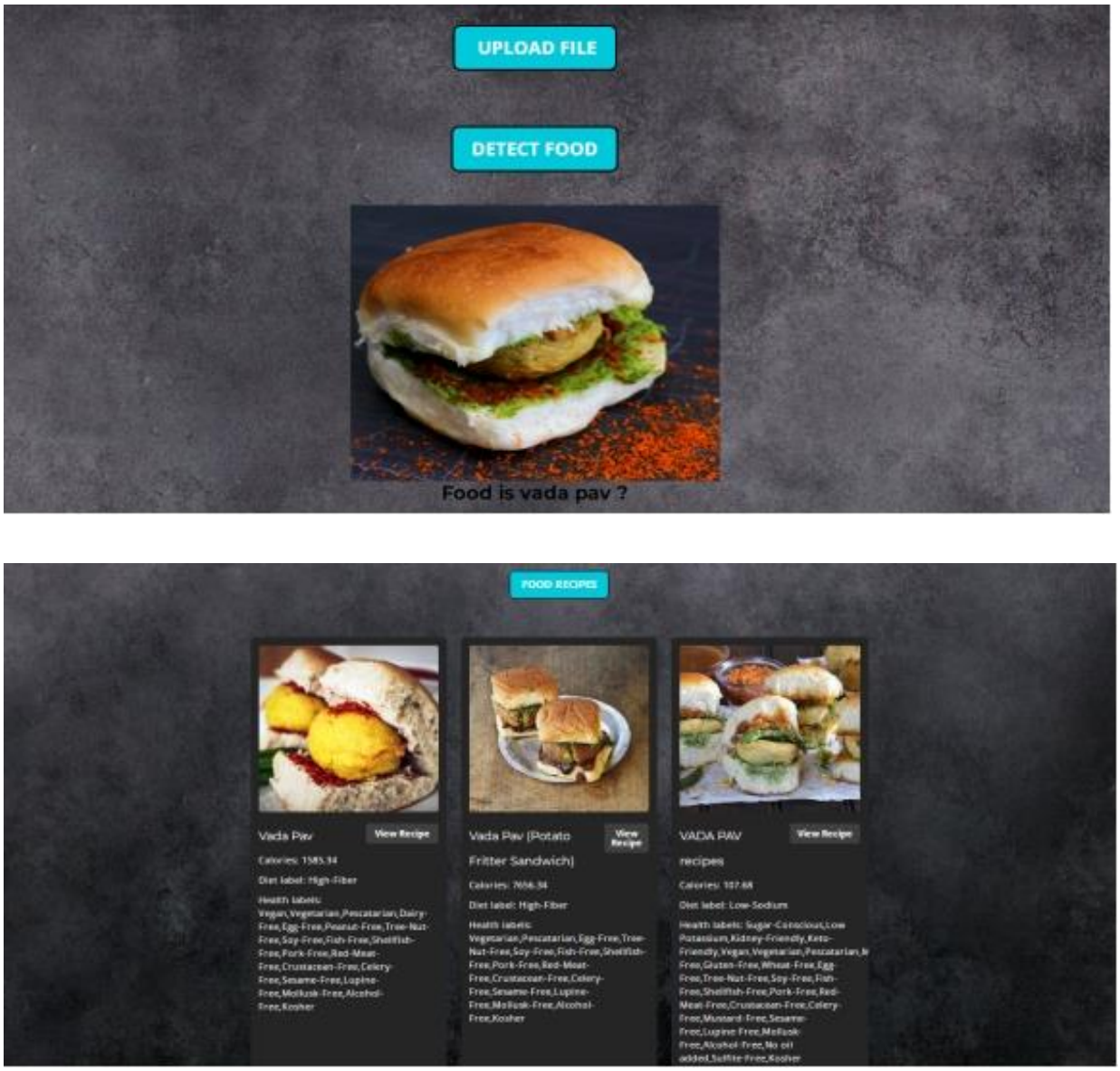


Figure 21. Food detection and recipe

- **Food Recipe by search**

In this page user can easily find various cuisine by just entering name of that recipes.

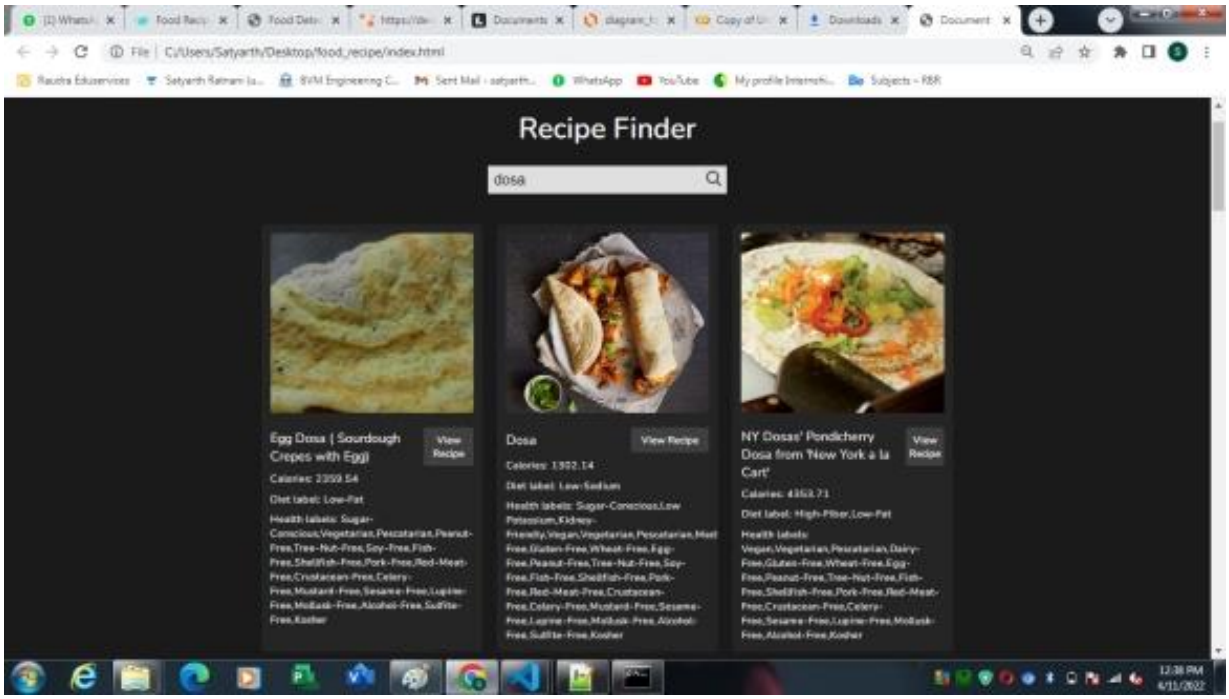


Figure 22. Recipe by search

- **Comment page**

User can add comment and share their opinion and read others comment also. based on that we can improve our website.

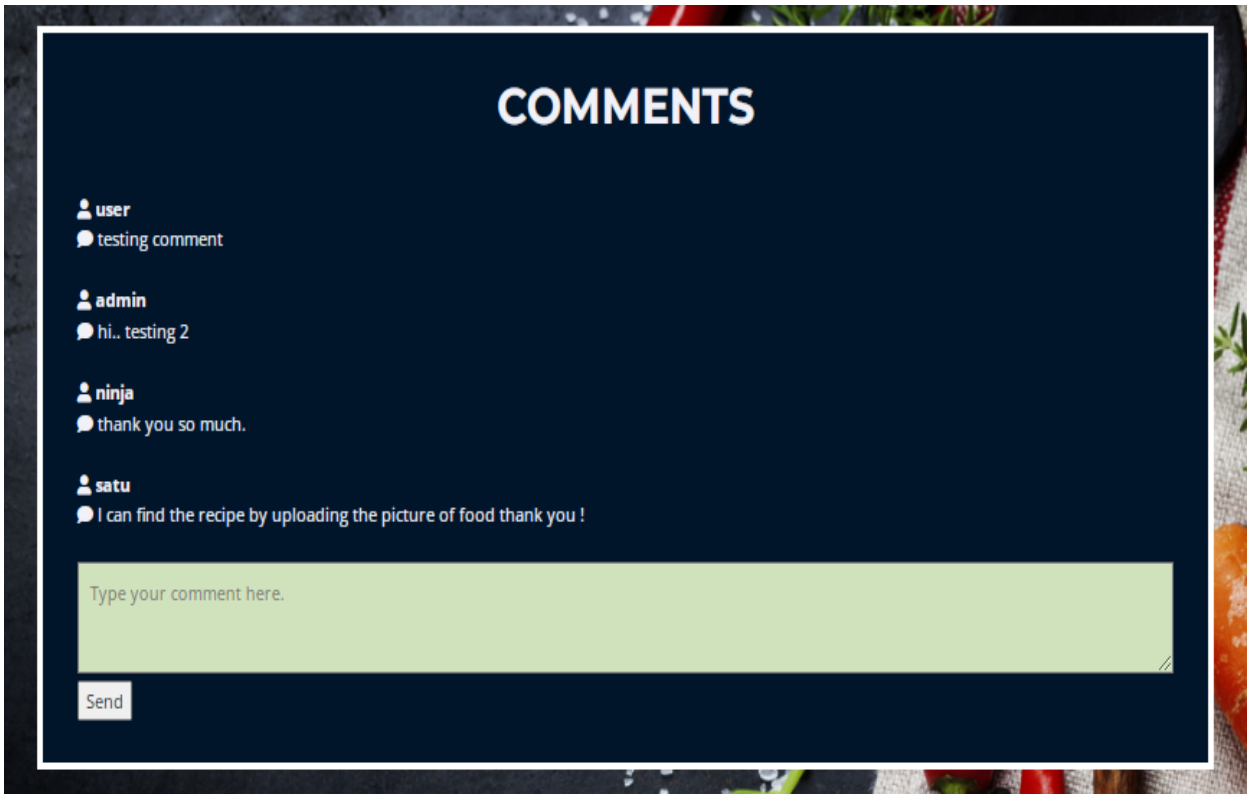


Figure 23. Comment page

- **Administrator**

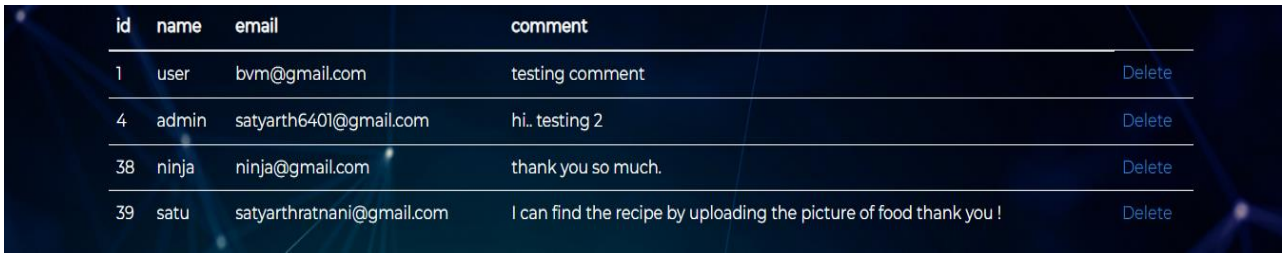
Figure 24. Admin panel

- Admin can see the details of all registered users block any particular user and delete his/her account.

id	name	email	password	
6	user	bvm@gmail.com	bvm	Delete
7	admin	satyarth6401@gmail.com	123	Delete
9	shubham	shubham@gmail.com	123	Delete
11	ninja	ninja@gmail.com	ninja@123	Delete
14	visvaraj	visvarajsinh.rathod0203@gmail.com	bapu	Delete
15	shubham1	shubhampatel9869@gmail.com	123	Delete
16	yash	yja8585@gmail.com	beluga	Delete
19	satu	satyarthratnani@gmail.com	1234	Delete

Figure 25. User details to admin

- Admin can see all the comments and delete particular comment.



id	name	email	comment	
1	user	bvm@gmail.com	testing comment	Delete
4	admin	satyarth6401@gmail.com	hi.. testing 2	Delete
38	ninja	ninja@gmail.com	thank you so much.	Delete
39	satu	satyarthratnani@gmail.com	I can find the recipe by uploading the picture of food thank you !	Delete

Figure 26. Comment details to admin

4.1.6 Recipe Search API

For recipe search from predicted food item we are using Edamam's API.

Edamam's Recipe Search API lets you integrate recipes and faceted recipe search into your websites or mobile applications.

There are other APIs available for food recipe search like Spoonacular, Mycookbook.io, Tasty but the reason of using Edamam's API is, it provide more number of API calls in it's developer option and latency is also comparatively less.

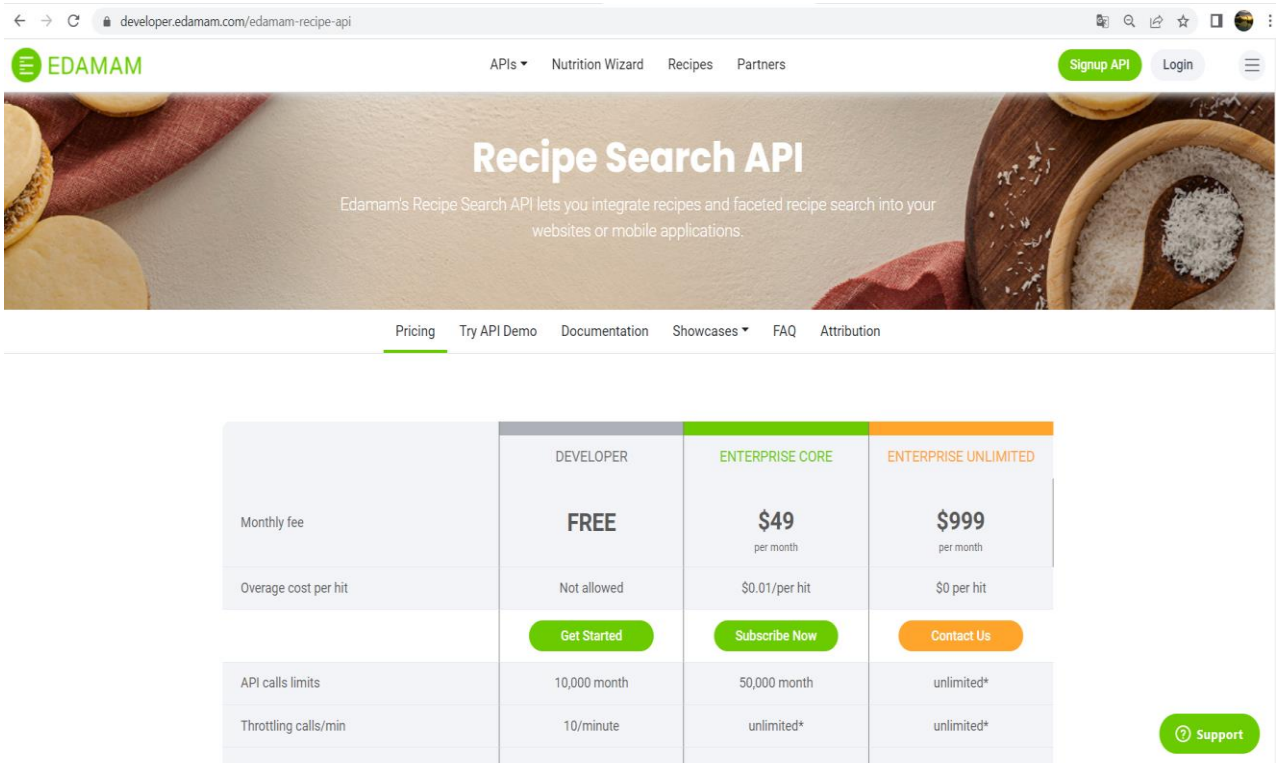


Figure 27. Edamam's recipe search API

4.2 Testing using Use Cases

Test	Expected outcome	Result
Login	Validation checking	Pass
Forgot password	Receiving password through mail	Pass
Upload image	User should able to upload image	Pass
prediction	Food item should be predicted accurately	Pass
Fetch recipes	Recipes should be shown to user	Pass
Recipe search	User can search recipe	Pass
Comment	User can comment	Pass
Admin panel	Admin can see user details, comments, delete any user or comment	Pass

Table 3. Testing using Cases

Chapter 5: Conclusion and Future Work

5.1 Conclusion

- As we planned and discussed to explore various related datasets and perform data preprocessing in order to extract the desired attributes and also perform model training using Machine Learning or Deep Learning algorithms to find the food name and its recipe through Edamam's API.

5.2 Future work

- In future we will add more images in our dataset to get more accuracy. Also add some feature like favorite recipe, comment on post, rating of recipe.

Chapter 6: References

Dataset

https://www.kaggle.com/datasets/kmader/food41

Backend

https://pyimagesearch.com/2019/05/27/keras-feature-extraction-on-large-datasets-with-deep-learning/

https://www.youtube.com/watch?v=35iRdqY01co

https://www.youtube.com/watch?v=PaSEVY9d4RI

Frontend

https://developer.edamam.com/edamam-recipe-api

https://hackanons.com/2019/04/run-python-script-on-clicking-html-button-map-output-on-html-page.html

https://www.geeksforgeeks.org/how-to-connect-to-an-api-in-javascript/