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

School of Computer Science and Engineering

J Component Report

Programme : B.Tech

Course Title : IoT Domain Analyst

Course Code : ECE3502

Slot : G1

Title: Recipe Recommendation System

Team Members: **Akshaya Ramachandran | 20BCE1727**

Gowtham N | 20BCE1857

Raksha Aruloli | 20BCE1737

Faculty: Dr. Rekha D

ABSTRACT

The goal of this project is to create a smart recipe recommendation system that can take live pictures of the vegetables in a user's fridge using an ESP32 camera module. For the purpose of identifying and categorising the veggies on the tray, the system will utilise a deep learning CNN model, especially InceptionV3. The software will offer personalised meal ideas depending on the user's choices, such as vegan or gluten-free options, by analysing the contents of the fridge. Users can reduce food wastage and improve their meal planning by having dishes suggested based on ingredients already in the refrigerator. In order to produce an intuitive and user-friendly app that can revolutionise the way people cook and decrease food waste, the project will integrate hardware and software components.

INTRODUCTION

A fast-paced way of living in the modern world frequently leaves little time for tasks like food preparation and planning. As a result, people frequently turn to quick and simple meal selections that might not be sustainable or healthy. Additionally, food waste is an issue that is becoming more and more prevalent throughout the world, with a sizable portion of food wasted as a result of poor meal preparation or ignorance of the proper component usage.

This project intends to create a smart recipe recommendation system that makes use of deep learning and computer vision technologies to recognise and categorise veggies in a user's refrigerator in order to address these issues. To precisely identify and classify each vegetable, the system will employ an ESP32 camera module to take live photographs of the produce. These images will then be loaded into a deep learning CNN model named InceptionV3.

The app will offer recipe suggestions based on the user's tastes once the algorithm has identified the vegetables in the refrigerator. Within the app, users can tailor their nutritional needs, such as veganism, gluten-free eating, or other preferences. The software will suggest meals that fit the user's preferences based on the available vegetables, reducing food waste and enhancing meal preparation.

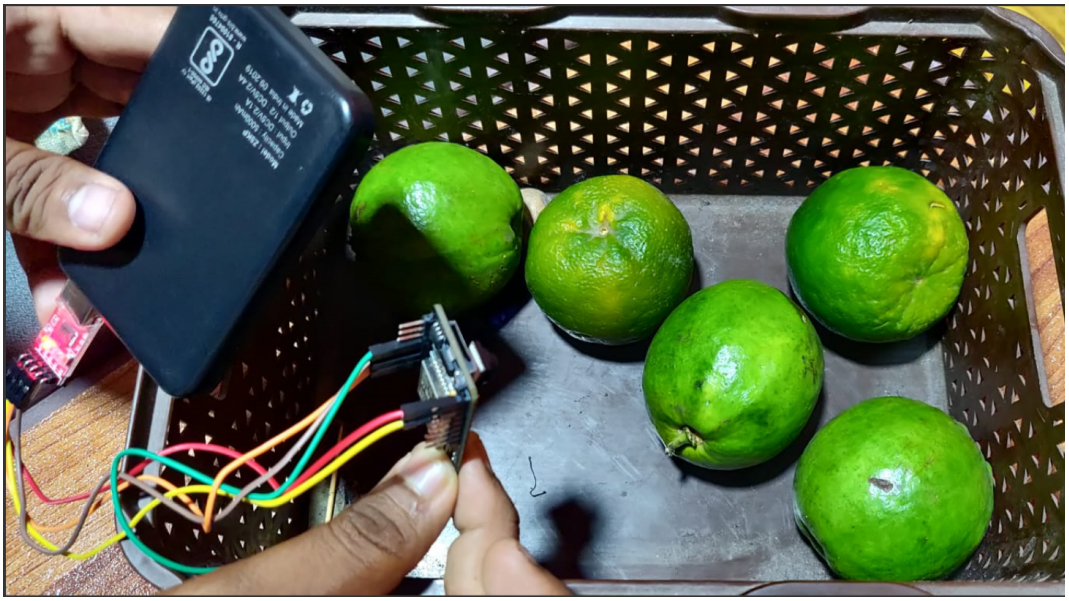
In order to create an intuitive and user-friendly app that can revolutionise the way people cook and decrease food waste, the project will integrate hardware and software development. The app will allow users to generate shopping lists based on the products they already have and the dishes they intend to make, in addition to offering personalised recipe ideas. This will not only minimise food waste but also help users save time and money by reducing unnecessary grocery trips.

Overall, this project aims to create a smart recipe recommendation system that can help people make informed and sustainable meal planning decisions while minimising food waste.

HARDWARE

The hardware components used in this project are:

1. ESP-32 CAM: This is the camera module used to capture the live image of the vegetables in the tray when the user clicks on “display current stock”. It serves as the primary source of collecting data.
2. Power Bank: This was used to power the ESP-32 CAM in the tray.
3. Jumper Wires



MODEL

Inceptionv3 is a deep learning architecture for image classification that was developed by Google. It is based on the concept of convolutional neural networks (CNNs) and is designed to identify patterns and features in images to make accurate predictions.

To predict a vegetable using Inceptionv3, the model would be trained on a large dataset of vegetable images, and the network would learn to recognize and classify different vegetables based on their unique features and characteristics.

The Inceptionv3 architecture uses a combination of convolutional layers, pooling layers, and fully connected layers to analyze and process the image data. The network starts by convolving the input image with a set of learned filters to extract low-level features such as edges and corners.

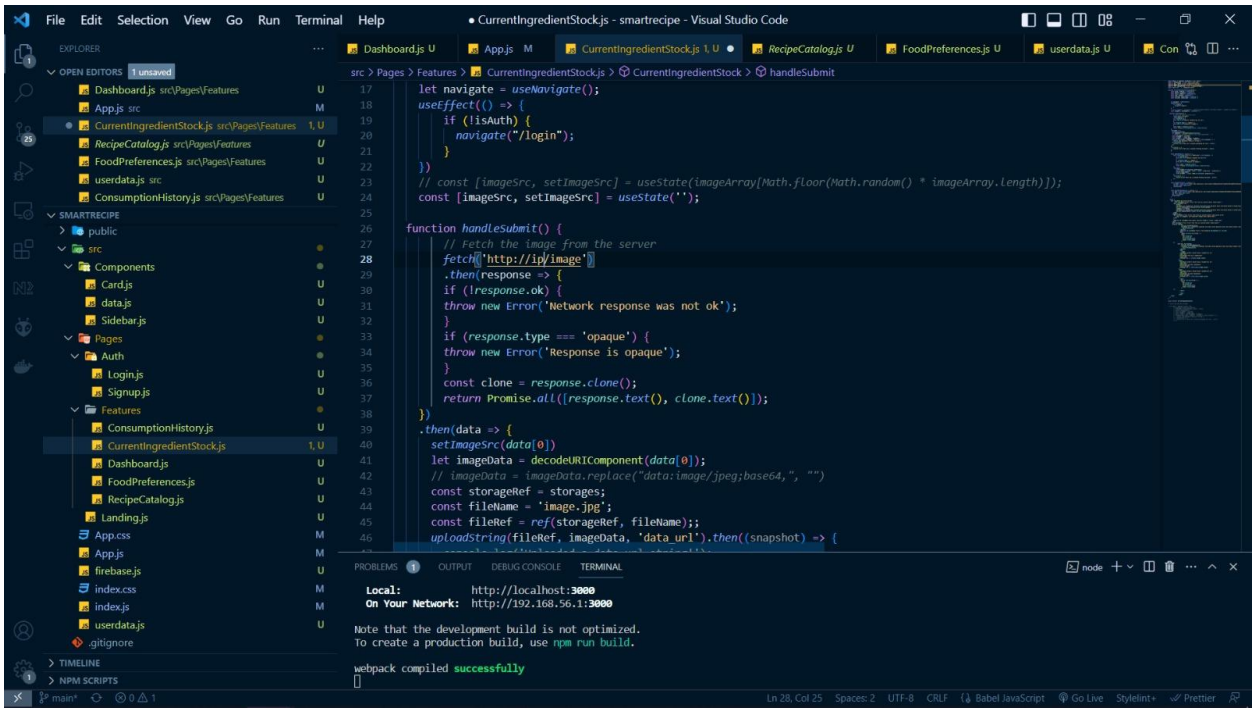
The output from each convolutional layer is then passed through a nonlinear activation function, such as the rectified linear unit (ReLU), which helps to introduce nonlinearity into the network and improve its ability to capture complex patterns and relationships.

As the image data is processed through the network, the features become more abstract and high-level, allowing the model to identify complex patterns and make accurate predictions about the image content.

In summary, Inceptionv3 is a deep learning architecture that uses convolutional neural networks to classify images, and it can be trained to accurately predict different vegetables based on their unique features and characteristics.

CODE

ReactJs:



File Edit Selection View Go Run Terminal Help • CurrentIngredientStock.js - smartrecipe - Visual Studio Code

src > Pages > Features > CurrentIngredientStock.js > CurrentIngredientStock > fetchRecommendations

```javascript
  }

  const fetchRecommendations = async() => {
    let data = await fetch(`https://api.spoonacular.com/recipes/complexSearch?apiKey=&cuisine=indian&diet=${veg}&includeIng
    let res = await data.json()
    res = res.results
    setRecs(res)
  }

return (
  <div className='px-0 mx-0 py-20'>
        <div className='h-full w-full flex flex-col justify-center items-center'>
      <div className='flex'>
          <button
          className='p-4 rounded-full bg-black text-white hover:bg-white hover:text-black border-2 border-black
          onClick={handleSubmit}>Display Current Stock</button>
          {imageSrc && (<button
          className='p-4 rounded-full bg-black text-white hover:bg-white hover:text-black border-2 border-black
          onClick={handlePredict}>Predict Current Stock</button>)}
      </div>
    {imageSrc && (
      <div className='h-full w-full flex flex-col justify-center items-center m-10'>
          <img src={imageSrc} alt="Captured" height={300} width={300}/>
      </div>
      )}
      {pred && <h1 className='text-center text-2xl'>Image in vision: {pred}</h1>}
    </div>
    <div className='h-full w-full flex flex-col justify-center items-center'>
    {pred && <button
                onClick={fetchPreferences}
                className='p-4 rounded-full bg-black text-white hover:bg-white hover:text-black border-2 borde
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

```
Local:            http://localhost:3000
On Your Network:  http://192.168.56.1:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

File Edit Selection View Go Run Terminal Help Login.js - smartrecipe - Visual Studio Code

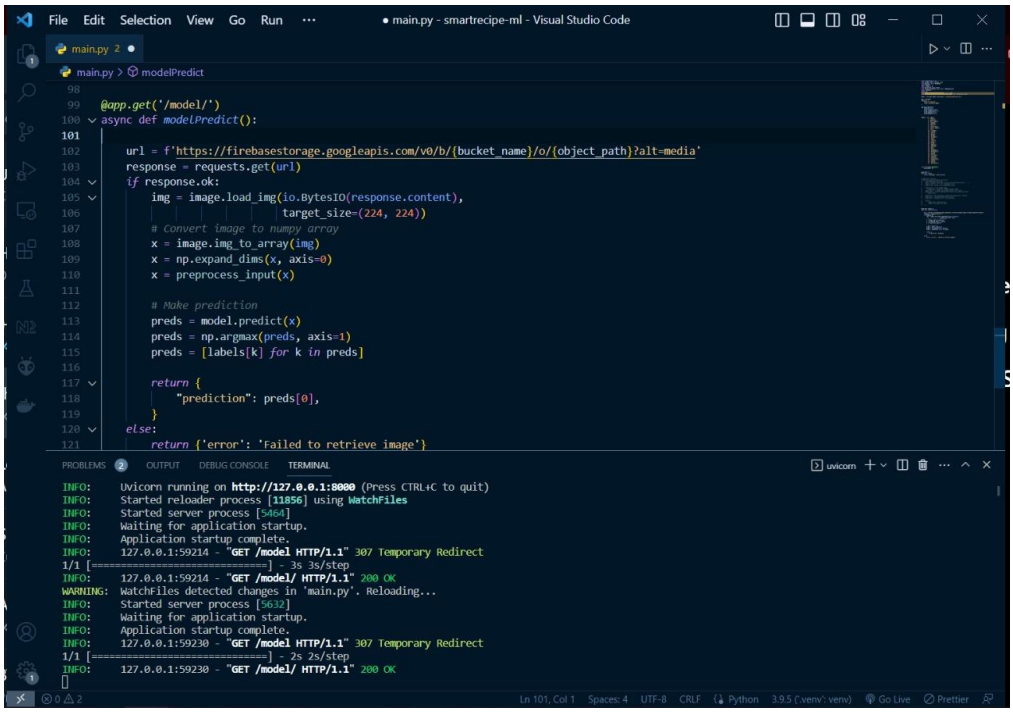
src > Pages > Auth > Login.js > Login > handleSubmit > then() callback

```javascript
import React, {useState, useEffect} from 'react';    'useEffect' is defined but never used.
import {auth, provider} from '../../firebase';    'provider' is defined but never used.
import {signInWithEmailAndPassword} from 'firebase/auth';
import { useNavigate } from 'react-router-dom';

const Login = ({setIsAuth}) => {

    const [name, setName] = useState('')    'name' is assigned a value but never used.
    const [email, setEmail] = useState('');
    const [password, setPassword] = useState('');

    let navigate = useNavigate();

    const handleSubmit = () => {
      if (email.match('^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\\.[A-Za-z]{2,}$') != null) {
        if (password.length > 5) {
          signInWithEmailAndPassword(auth, email, password)
          .then((userCredential) => {
            const user = userCredential.user;    'user' is assigned a value but never used.
            setIsAuth(true);
            navigate("/dashboard");
          })
          .catch((error) => {
            alert('Invalid credentials');
          });
        }
        else {
            setPassword('');
            console.log('Length of password should be more than 5.');
        }
```

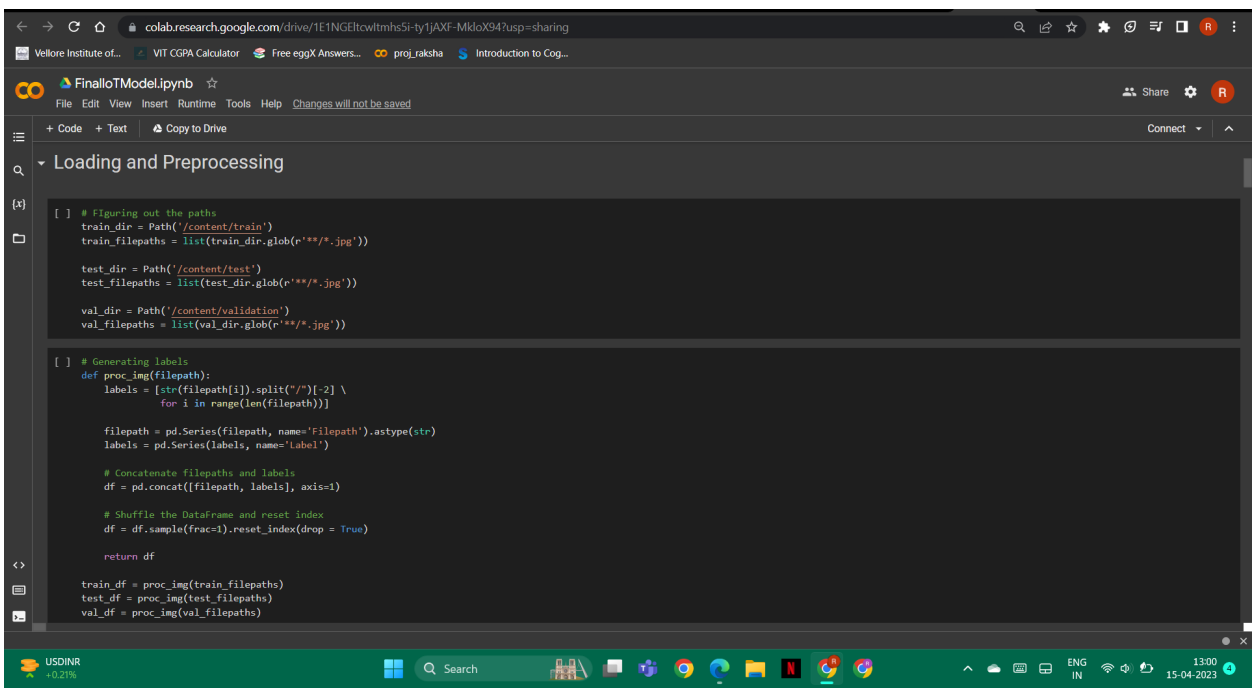
PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL

```
Compiling...
```

FastAPI:



ML model implementation:



colab.research.google.com/drive/1E1NGEltcwltmhs5i-ty1jAXF-MkloX94?usp=sharing

Vellore Institute of... | VIT CGPA Calculator | Free eggX Answers... | proj_raksha | Introduction to Cog...

FinalloTModel.ipynb

File Edit View Insert Runtime Tools Help Changes will not be saved

Share

+ Code + Text Copy to Drive

Connect

Loading and Augmentation

```
[ ] # Initializing the generators
    train_generator = tf.keras.preprocessing.image.ImageDataGenerator(
        preprocessing_function=tf.keras.applications.inception_v3.preprocess_input
    )

    test_generator = tf.keras.preprocessing.image.ImageDataGenerator(
        preprocessing_function=tf.keras.applications.inception_v3.preprocess_input
    )

    validation_generator = tf.keras.preprocessing.image.ImageDataGenerator(
        preprocessing_function=tf.keras.applications.inception_v3.preprocess_input
    )
```

```
# Prr-process the images
train_images = train_generator.flow_from_dataframe(
    dataframe=train_df,
    x_col='Filepath',
    y_col='Label',
    target_size=(224, 224),
    color_mode='rgb',
    class_mode='categorical',
    batch_size=32,
    shuffle=True,
    seed=0,
    rotation_range=30,
    zoom_range=0.15,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.15,
```

colab.research.google.com/drive/1E1NGEltcwltmhs5i-ty1jAXF-MkloX94?usp=sharing

Vellore Institute of... | VIT CGPA Calculator | Free eggX Answers... | proj_raksha | Introduction to Cog...

FinalloTModel.ipynb

File Edit View Insert Runtime Tools Help Changes will not be saved

Share

+ Code + Text Copy to Drive

Connect

```
    horizontal_flip=True,
    fill_mode="nearest"
)

val_images = validation_generator.flow_from_dataframe(
    dataframe=val_df,
    x_col='Filepath',
    y_col='Label',
    target_size=(224, 224),
    color_mode='rgb',
    class_mode='categorical',
    batch_size=32,
    shuffle=True,
    seed=0,
    rotation_range=30,
    zoom_range=0.15,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.15,
    horizontal_flip=True,
    fill_mode="nearest"
)

test_images = test_generator.flow_from_dataframe(
    dataframe=test_df,
    x_col='Filepath',
    y_col='Label',
    target_size=(224, 224),
    color_mode='rgb',
    class_mode='categorical',
    batch_size=32,
    shuffle=False
)
```

```
Found 2780 validated image filenames belonging to 36 classes.
```

colab.research.google.com/drive/1E1NGEltcwItmhs5i-ty1jAXF-MkloX94?usp=sharing

Vellore Institute of... | VIT CGPA Calculator | Free eggX Answers... | proj_raksha | Introduction to Cog...

FinalloTModel.ipynb

File Edit View Insert Runtime Tools Help Changes will not be saved

Share | Connect

+ Code + Text | Copy to Drive

```
[ ] # Load the pretained model
    pretrained_model = tf.keras.applications.InceptionV3(
        input_shape=(224, 224, 3),
        include_top=False,
        weights='imagenet',
        pooling='avg'
    )
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/inception_v3/inception_v3_weights_tf_dim_ordering_tf_kernels_notop.h5
87910968/87910968 [==============================] - 0s 0us/step
```

Training Model

```
inputs = pretrained_model.input
x = tf.keras.layers.Dense(36, activation='relu')(pretrained_model.output)
outputs = tf.keras.layers.Dense(36, activation='softmax')(x)

model = tf.keras.Model(inputs=inputs, outputs=outputs)
model.compile(
    optimizer='adam',
    loss='categorical_crossentropy',
    metrics=['accuracy']
)
```

```
[ ] model.fit(x=train_images, validation_data=val_images, epochs=5, verbose=1)
```

```
Epoch 1/5
87/87 [==============================] - 1487s 17s/step - loss: 2.4238 - accuracy: 0.3583 - val_loss: 7.5940 - val_accuracy: 0.0808
Epoch 2/5
87/87 [==============================] - 1517s 17s/step - loss: 1.7708 - accuracy: 0.5094 - val_loss: 2.9936 - val_accuracy: 0.4371
```


```
[ ] model.fit(x=train_images, validation_data=val_images, epochs=5, verbose=1)

Epoch 1/5
87/87 [==============================] - 1487s 17s/step - loss: 2.4238 - accuracy: 0.3583 - val_loss: 7.5940 - val_accuracy: 0.0808
Epoch 2/5
87/87 [==============================] - 1517s 17s/step - loss: 1.7708 - accuracy: 0.5094 - val_loss: 2.9936 - val_accuracy: 0.4371
```


```
Epoch 3/5
87/87 [==============================] - 1473s 17s/step - loss: 1.2919 - accuracy: 0.6360 - val_loss: 2.4606 - val_accuracy: 0.4910
Epoch 4/5
87/87 [==============================] - 1466s 17s/step - loss: 1.0914 - accuracy: 0.6766 - val_loss: 1.6406 - val_accuracy: 0.5958
Epoch 5/5
87/87 [==============================] - 1492s 17s/step - loss: 0.8999 - accuracy: 0.7248 - val_loss: 1.5780 - val_accuracy: 0.6737
<keras.callbacks.History at 0x7fc64d9d7a30>
```

```
model.save("/content/gdrive/MyDrive/ingredientmodel/ingredientdetection.h5")
```

+ Code + Text

colab.research.google.com/drive/1E1NGEltcwltmhs5i-ty1jAXF-MkloX94?usp=sharing

Vellore Institute of... | VIT CGPA Calculator | Free eggX Answers... | proj_raksha | Introduction to Cog...

FinalloTModel.ipynb

File Edit View Insert Runtime Tools Help Changes will not be saved

+ Code + Text Copy to Drive

Share Connect

Evaluation

```
[ ] from sklearn.metrics import accuracy_score
    acc = accuracy_score(y_test, pred)
    print(f'Accuracy on the test set: {100*acc:.2f}%')
```

```
Accuracy on the test set: 67.37%
```

```
[ ] import numpy as np
    import tensorflow as tf
    from tensorflow import keras
    model = keras.models.load_model("/content/gdrive/MyDrive/ingredientmodel/ingredientdetection.h5")
```

```
import tensorflow as tf
import io
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.inception_v3 import preprocess_input, decode_predictions
import numpy as np
import requests
from PIL import Image

# Load pre-trained InceptionV3 model
# model = InceptionV3(weights='imagenet')

# Load image
img_path = 'https://firebasestorage.googleapis.com/v0/b/smartrecipe-c1cb2.appspot.com/o/image.jpg?alt=media&token=ec5bf5d4-8ac6-4380-b75f-c93756ab345c'
response = requests.get(img_path)
if response.ok:
  img = image.load_img(io.BytesIO(response.content), target_size=(224, 224))
  x = image.img_to_array(img)
```

colab.research.google.com/drive/1E1NGEltcwltmhs5i-ty1jAXF-MkloX94?usp=sharing

Vellore Institute of... | VIT CGPA Calculator | Free eggX Answers... | proj_raksha | Introduction to Cog...

FinalloTModel.ipynb

File Edit View Insert Runtime Tools Help Changes will not be saved

+ Code + Text Copy to Drive

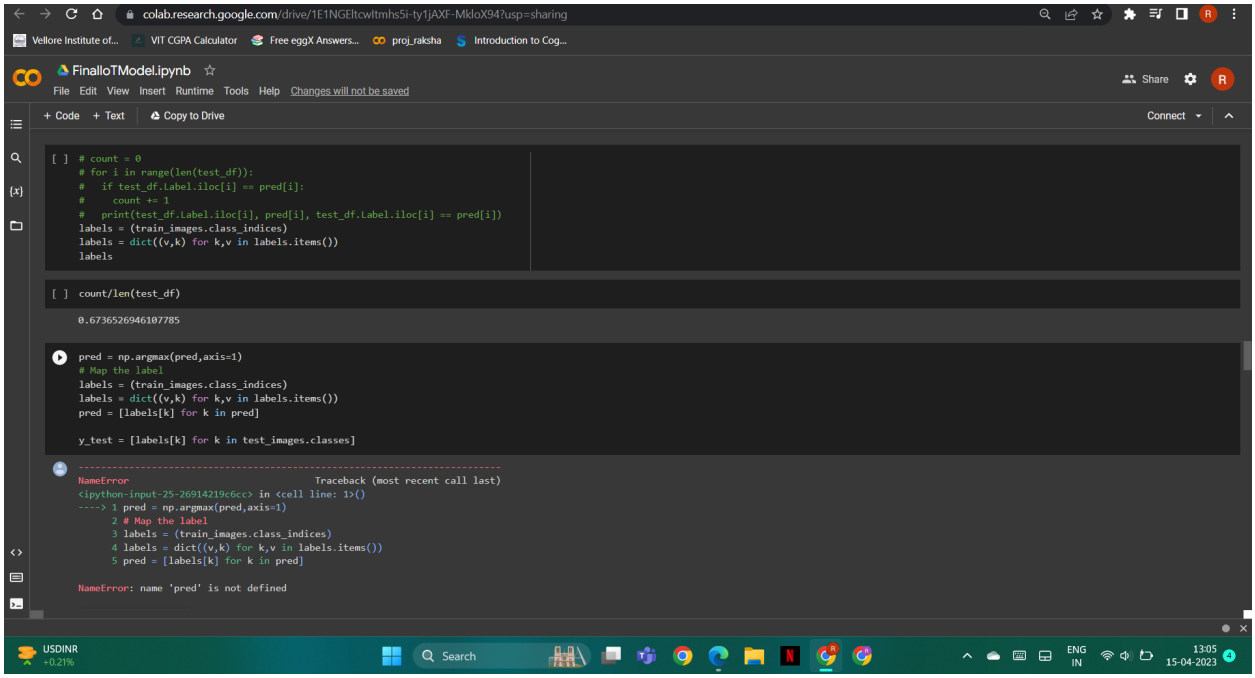
Share Connect

```
# Load image
img_path = 'https://firebasestorage.googleapis.com/v0/b/smartrecipe-c1cb2.appspot.com/o/image.jpg?alt=media&token=ec5bf5d4-8ac6-4380-b75f-c93756ab345c'
response = requests.get(img_path)
if response.ok:
  img = image.load_img(io.BytesIO(response.content), target_size=(224, 224))
  x = image.img_to_array(img)
  x = preprocess_input(x)
  # Convert image to numpy array
  x = image.img_to_array(img)
  x = np.expand_dims(x, axis=0)
  x = preprocess_input(x)

  # Make prediction
  preds = model.predict(x)

else:
  print("'error': 'Failed to retrieve image'")

# img = image.load_img(img_path, target_size=(224, 224))

# # Convert image to numpy array
# x = image.img_to_array(img)
# x = np.expand_dims(x, axis=0)
# x = preprocess_input(x)

# # Make prediction
# preds = model.predict(x)
```

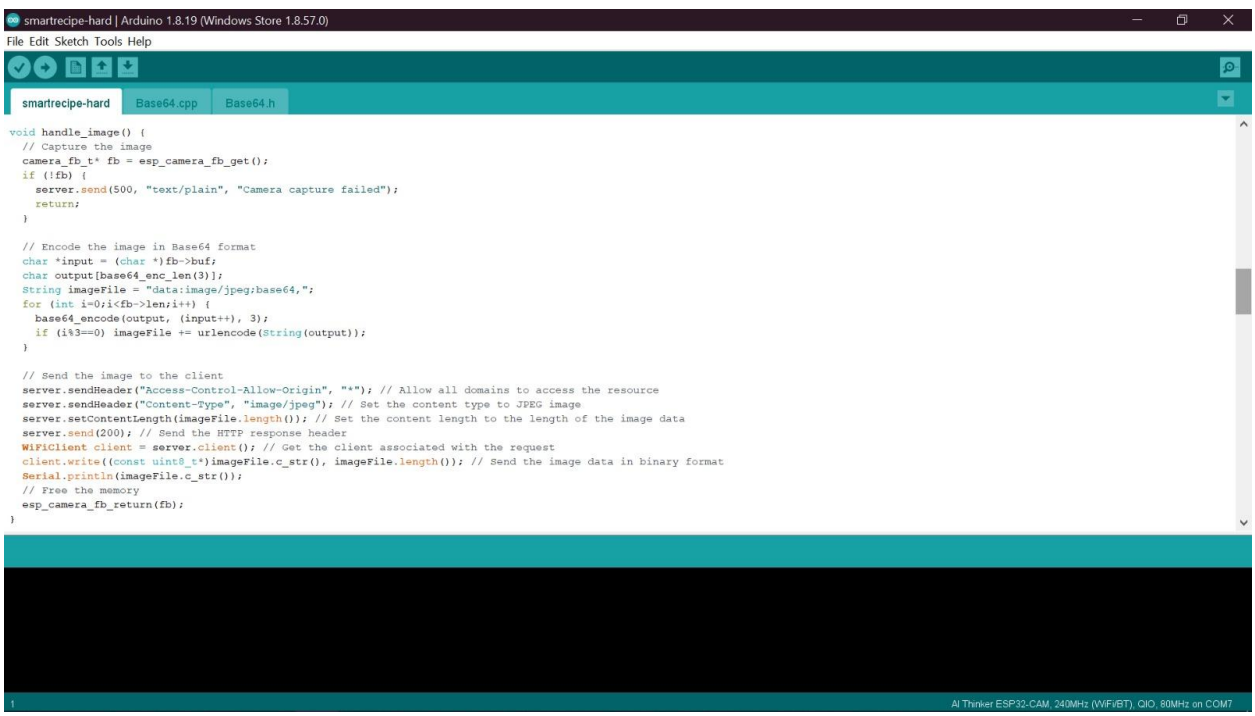
```
1/1 [==============================] - 2s 2s/step
```

```
[ ] np.argmax(preds, axis=1)
```

```
array([1])
```


colab.research.google.com/drive/1E1NGEltcwItmhs5i-ty1jAXF-MkloX94?usp=sharing
FinalloTModel.ipynb
File Edit View Insert Runtime Tools Help Changes will not be saved
+ Code + Text Copy to Drive
Connect
Share
[ ] # count = 0
# for i in range(len(test_df)):
# if test_df.Label.iloc[i] == pred[i]:
# count += 1
# print(test_df.Label.iloc[i], pred[i], test_df.Label.iloc[i] == pred[i])
labels = (train_images.class_indices)
labels = dict((v,k) for k,v in labels.items())
labels
[ ] count/len(test_df)
0.673652694610778 5
pred = np.argmax(pred,axis=1)
# Map the label
labels = (train_images.class_indices)
labels = dict((v,k) for k,v in labels.items())
pred = [labels[k] for k in pred]
y_test = [labels[k] for k in test_images.classes]
NameError Traceback (most recent call last)
<ipython-input-25-26914219c6cc> in <cell line: 1>()
----> 1 pred = np.argmax(pred,axis=1)
2 # Map the label
3 labels = (train_images.class_indices)
4 labels = dict((v,k) for k,v in labels.items())
5 pred = [labels[k] for k in pred]
NameError: name 'pred' is not defined
USDINR +0.21%
Search
ENG IN
13:05 15-04-2023

Arduino:


smartrecipe-hard | Arduino 1.8.19 (Windows Store 1.8.57.0)
File Edit Sketch Tools Help
smartrecipe-hard
Base64.cpp
Base64.h
void handle_image() {
// Capture the image
camera_fb_t* fb = esp_camera_fb_get();
if (!fb) {
server.send(500, "text/plain", "Camera capture failed");
return;
}
// Encode the image in Base64 format
char *input = (char *)fb->buf;
char output[base64_enc_len(3)];
String imageFile = "";
for (int i=0;i<fb->len;i++) {
base64_encode(output, (input++), 3);
if (i%3==0) imageFile += urlencode(String(output));
}
// Send the image to the client
server.sendHeader("Access-Control-Allow-Origin", "*"); // Allow all domains to access the resource
server.sendHeader("Content-Type", "image/jpeg"); // Set the content type to JPEG image
server.setContentLength(imageFile.length()); // Set the content length to the length of the image data
server.send(200); // Send the HTTP response header
WiFiClient client = server.client(); // Get the client associated with the request
client.write((const uint8_t*)imageFile.c_str(), imageFile.length()); // Send the image data in binary format
Serial.println(imageFile.c_str());
// Free the memory
esp_camera_fb_return(fb);
}
1
AI Thinker ESP32-CAM, 240MHz (WiFi/BT), QIO, 80MHz on COM7

smartrecipe-hard | Arduino 1.8.19 (Windows Store 1.8.57.0)

File Edit Sketch Tools Help

smartrecipe-hard | Base64.cpp | Base64.h

```
String urlencode(String str)
{
    String encodedString="";
    char c;
    char code0;
    char code1;
    char code2;
    for (int i =0; i < str.length(); i++){
      c=str.charAt(i);
      if (c == ' '){
        encodedString+= '+';
      } else if (isalnum(c)){
        encodedString+=c;
      } else{
        code1=(c & 0xf)+'0';
        if ((c & 0xf) >9){
            code1=(c & 0xf) - 10 + 'A';
        }
        c=(c>>4)&0xf;
        code0=c+'0';
        if (c > 9){
            code0=c - 10 + 'A';
        }
        code2='\0';
        encodedString+='%';
        encodedString+=code0;
        encodedString+=code1;
        //encodedString+=code2;
      }
      yield();
    }
```

1 AI Thinker ESP32-CAM, 240MHz (WiFi/BT), QIO, 80MHz on COM7

smartrecipe-hard | Arduino 1.8.19 (Windows Store 1.8.57.0)

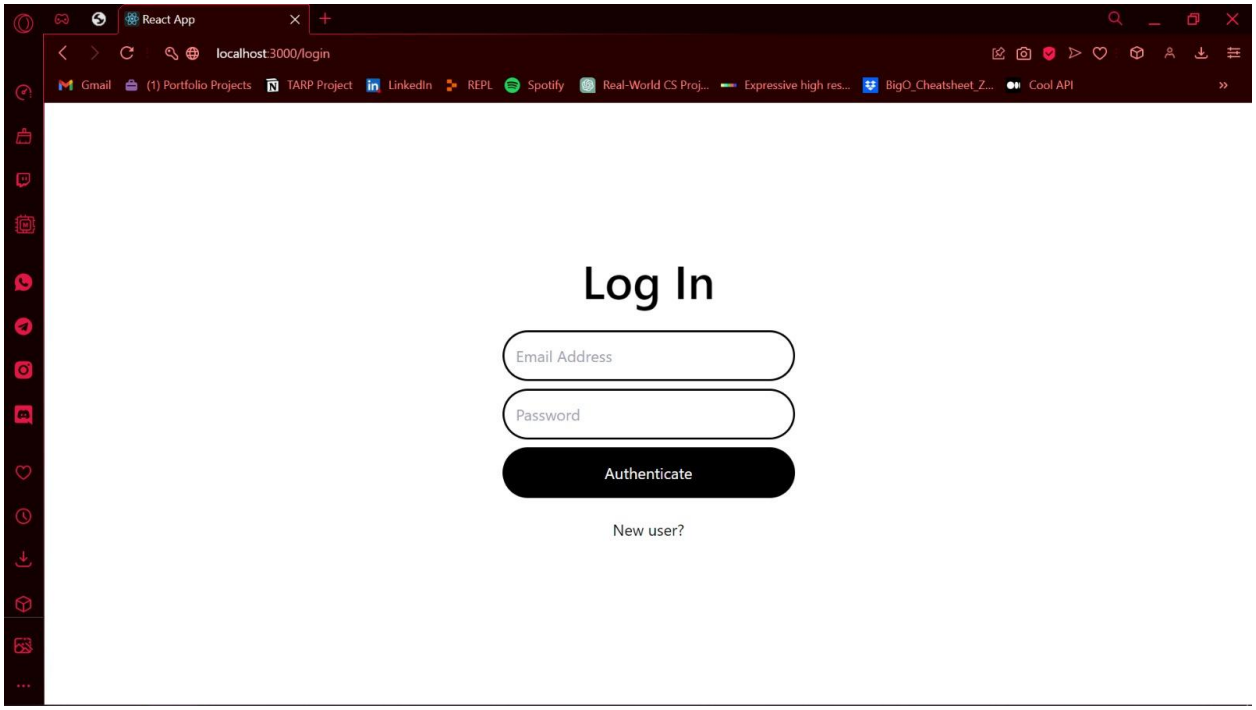
File Edit Sketch Tools Help

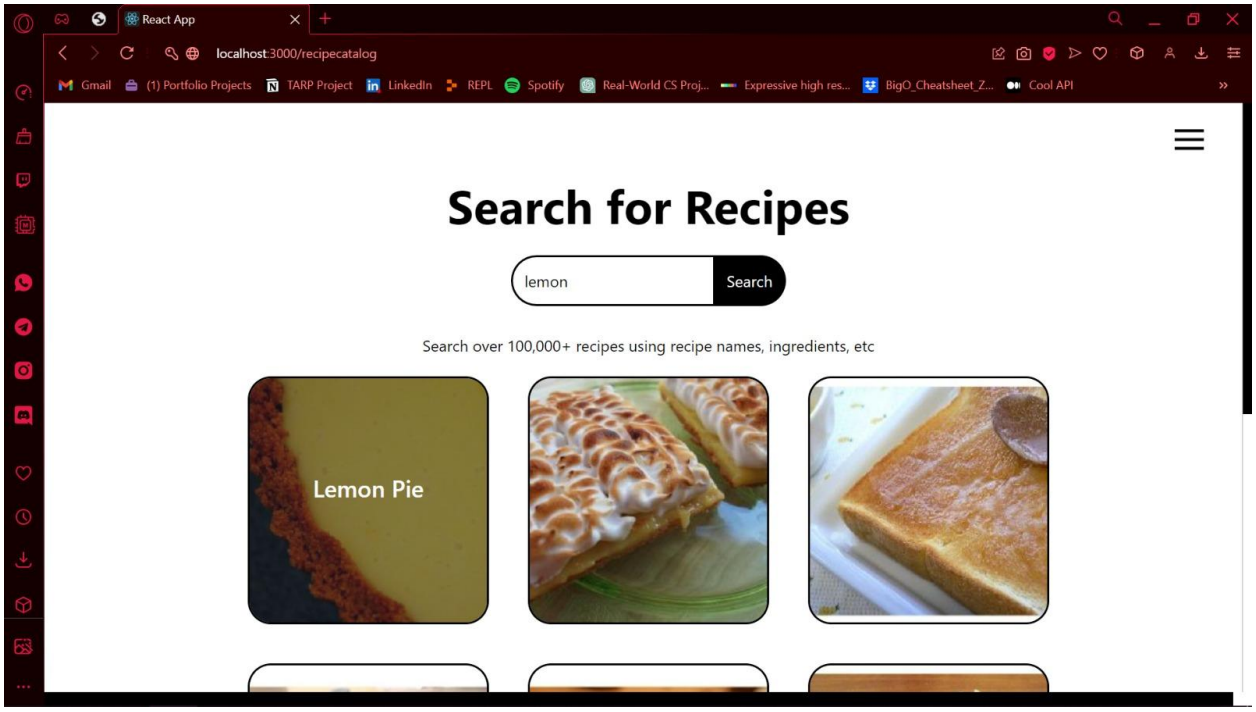
smartrecipe-hard | Base64.cpp | Base64.h

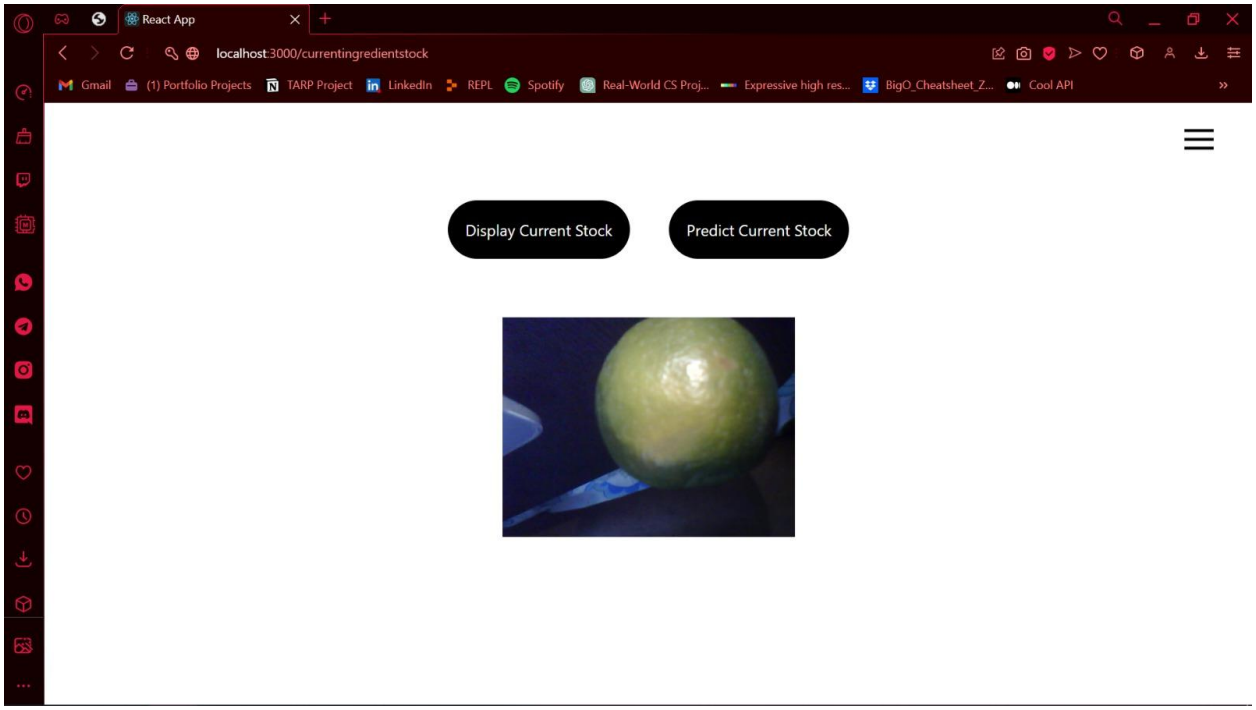
```
      yield();
    }
    return encodedString;
}
void setup() {
// Initialize the serial port
Serial.begin(115200);

// Connect to Wi-Fi
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
delay(1000);
Serial.println("Connecting to WiFi...");
}
Serial.println("Connected to WiFi");
Serial.println(WiFi.localIP());

// Initialize the camera
init_camera();

// Start the server
server.on("/", handle_root);
server.on("/image", handle_image);
server.begin();
Serial.println("Server listening");
}
```

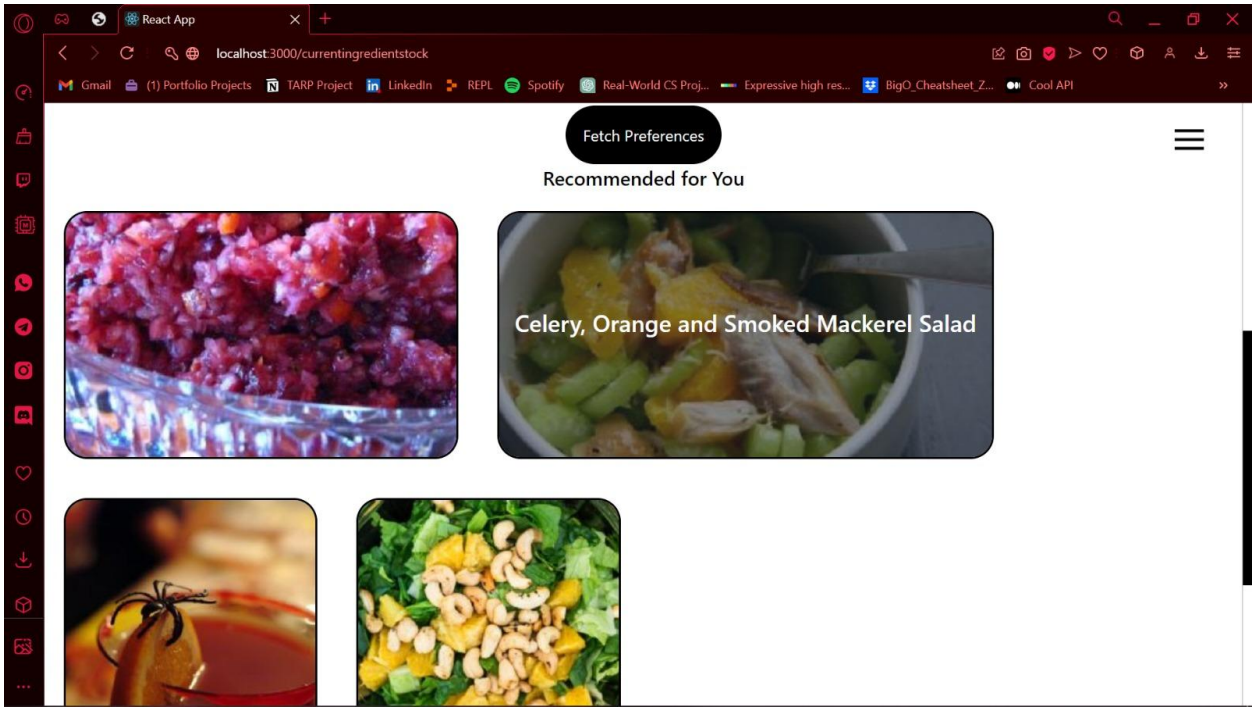
1 AI Thinker ESP32-CAM, 240MHz (WiFi/BT), QIO, 80MHz on COM7

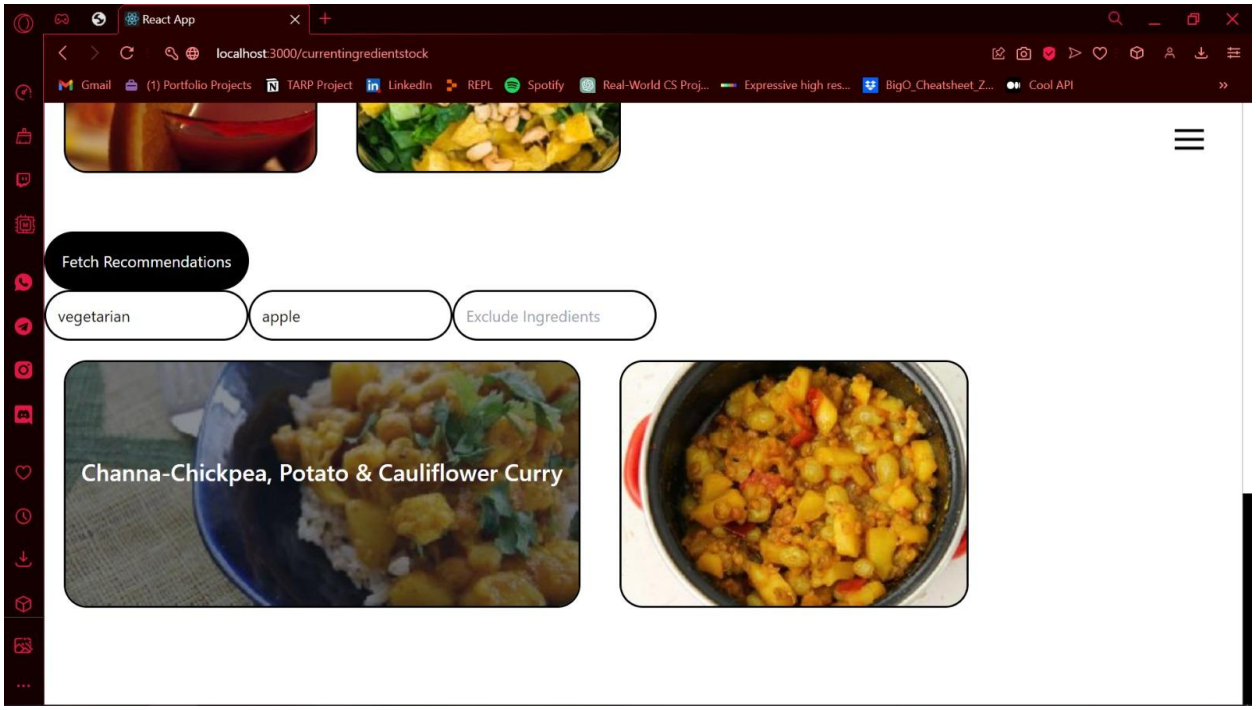
OUTPUT


React App
localhost:3000/login
Gmail
(1) Portfolio Projects
TARP Project
LinkedIn
REPL
Spotify
Real-World CS Proj...
Expressive high res...
BigO_Cheatsheet_Z...
Cool API
Log In
Email Address
Password
Authenticate
New user?


React App
localhost:3000/recipecatalog
Gmail
(1) Portfolio Projects
TARP Project
LinkedIn
REPL
Spotify
Real-World CS Proj...
Expressive high res...
BigO_Cheatsheet_Z...
Cool API
Search for Recipes
lemon
Search
Search over 100,000+ recipes using recipe names, ingredients, etc
Lemon Pie


React App
localhost:3000/currentingredientstock
Gmail
(1) Portfolio Projects
TARP Project
LinkedIn
REPL
Spotify
Real-World CS Proj...
Expressive high res...
BigO_Cheatsheet_Z...
Cool API
Display Current Stock
Predict Current Stock


React App
localhost:3000/currentingredientstock
Gmail
(1) Portfolio Projects
TARP Project
LinkedIn
REPL
Spotify
Real-World CS Proj...
Expressive high res...
BigO_Cheatsheet_Z...
Cool API
Fetch Preferences
Recommended for You
Celery, Orange and Smoked Mackerel Salad



CONCLUSION

In conclusion, a smart recipe recommender using an ESP32 camera module and deep learning CNN model called InceptionV3 can significantly improve meal planning and reduce food waste. By using live images of the vegetable tray in the fridge, the system can identify and classify the vegetables, then recommend recipes based on the user's preferences such as vegan, gluten-free, etc. This solution is not only convenient for users but also helps to reduce food waste by encouraging the use of ingredients already available at home. By providing an app interface for the user, it increases efficiency of use as well as flow of the system by guiding the user step by step on the process. With further development and refinement, this technology has the potential to transform the way we plan and prepare meals across different platforms.