User Acceptance Testing

Date	24 November 2022
TEAM ID	PNT2022TMID37024
Project title	AI Powered Nutrition Analyst for Fitness Enthusiasts
Maximum Marks	10 Marks

Model Performance Testing:

Project team shall fill the following information in model performance testing template.

S.No	Parameter	Values	Screenshot
1	Model Summary	Total Params: 21,885,4485 Trainable Params: 1.024,005 Non-trainable Params: 20,861,480	Attached below
2	Accuracy	Training Accuracy – 72% Validation Accuracy – 59%	Attached below
3	Confidence Score	Class Detected – Nil Confidence Score - Nil	Nil

User Acceptance Testing

We are performing **White Box Testing** for select the package module.

Pseudocode for select the package module is-

1. Select the package() procedure begins

2. READ the package name, features, offers and duration from the package database

3. DISPLAY the package name, features, offers and duration

4. DO

5. GET the package name, features, offers and duration

6. STORE the package selected to the customer's database

7. PROCEED to payment screen //another module

8. WHILE select package is NULL

9. //End DO...WHILE

10. Procedure ends

FLOWGRAPH

CYCLOMATIC COMPLEXITY OF RESULTANT GRAPH

V (G) = Number of regions

= 2

V (G) = Edges-Nodes+2

= 8-8+2

=2

V (G) = Predicate nodes+1

= 1+1

= 2

LINEARLY INDEPENDENT PATHS FOR FLOW GRAPHS

Path 1: 1-2-3-4-5-6-7-8-9-10

Path 2: 1-2-3-4-5-6-7-8-4-5-6-7-8-9-10

| TEST ID	INPUT VALUES	ACTUAL OUTPUT	EXPECTED OUTPUT
1	PACKAGE IS SELECTED	TO BE OBSERVED AFTER EXECUTION	DISPLAY THE SELECTED PACKAGE
2	PACKAGE IS NOT SELECTED	TO BE OBSERVED AFTER EXECUTION	SHOW THE PACKAGES TO SELECT UNTIL ONE IS SELECTED

Performance Testing

Future Scope

- Offers payment convenience in personal training subscriptions.

- Useful articles.

- Video instructions.

- Diet Plans.

- Individual progress tracking.

SCREENSHOTS:
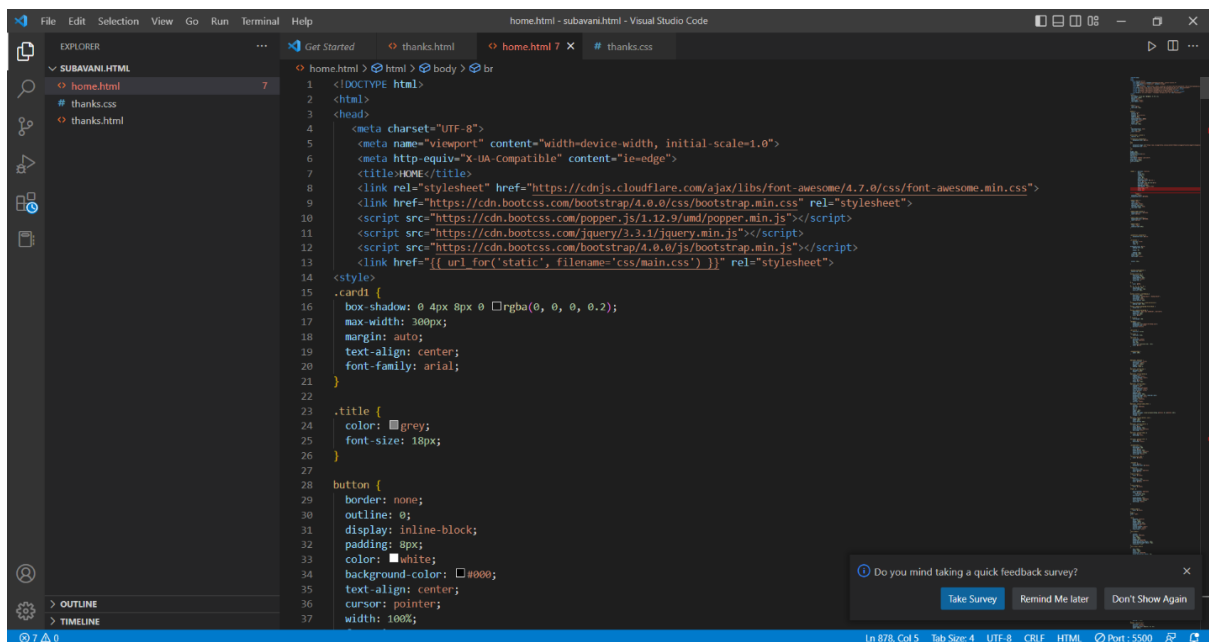
Image Preprocessing

```python
from keras.preprocessing.image import ImageDataGenerator
```

Image Data Augmentation

```python
train_datagen = ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1./255)
```

Applying Image DataGenerator Functionality To Trainset And Testset

```python
x_train = train_datagen.flow_from_directory(
    r'/content/drive/MyDrive/Colab Notebooks/Dataset/TRAIN_SET',
    target_size=(64, 64),batch_size=5,color_mode='rgb',class_mode='sparse')
x_test = test_datagen.flow_from_directory(
    r'/content/drive/MyDrive/Colab Notebooks/Dataset/TEST_SET',
    target_size=(64, 64),batch_size=5,color_mode='rgb',class_mode='sparse')
```

3. Adding CNN Layers

```python
classifier = Sequential()
classifier.add(Conv2D(32, (3, 3), input_shape=(64, 64, 3), activation='relu'))
classifier.add(MaxPooling2D(pool_size=(2, 2)))
classifier.add(Conv2D(32, (3, 3), activation='relu'))
classifier.add(MaxPooling2D(pool_size=(2, 2)))
classifier.add(Flatten())
```

4. Adding Dense Layers

```python
classifier.add(Dense(units=128, activation='relu'))
classifier.add(Dense(units=5, activation='softmax'))
```

```python
classifier.summary()
```

```
Model: "sequential_1"
_____
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 62, 62, 32)        896
```

5. Configure The Learning Process

```python
classifier.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
```

6. Train The Model

```python
classifier.fit_generator(generator=x_train,steps_per_epoch = len(x_train),epochs=20, validation_data=x_test,validation_steps = len(x_test))
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. P]

Epoch 1/20
494/824 [================>............] - ETA: 6:52 - loss: 0.7194 - accuracy: 0.7174
```

7. Saving The Model

```python
classifier.save('nutrition.h5')
```

```
190  <body>
191  <div class="header">
192  <div style="...">NUTRITION IMAGE ANALYSIS</div>
193    <div class="topnav-right"style="...">
194
195      <a href="{{ url_for('home')}}"><b>HOME</b></a>
196      <a  class="active" href="{{ url_for('image1')}}"><b>CLASSIFY</b></a>
197    </div>
198  </div>
199  <br>
200
201
202  </div>
203  <div class="container">
204    <center>
205  <div id="content" style="...">{% block content %}{% endblock %}</div></center>
206    </div>
207  </body>
208
209  <footer>
210    <script src="{{ url_for('static', filename='js/main.js') }}" type="text/javascript"></script>
211  </footer>
212
213  </html>
214
```

```
4   <style>
5
6   </style>
7        <meta charset="utf-8">
8        <title>NUTRITION IMAGE ANALYSIS</title>
9        <link rel="shortcut icon" href="{{ url_for('static', filename='diabetes-favicon.ico') }}">
10       <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='style.css') }}">
11       <script src="https://kit.fontawesome.com/5f3f547070.js" crossorigin="anonymous"></script>
12       <link href="https://fonts.googleapis.com/css2?family=Pacifico&display=swap" rel="stylesheet">
13   </head>
14
15       <div class="results">
16           <p style="..."><h4 style="...">IMAGE CLASSIFIED IS : <h4><b><h4 style="..."><u>{{showcase1}}<h4><br>
17
18       </div>
19       <br>
20       <br>
21
22   </div>
23       </body>
24   </html>
```