

# USER ACCEPTANCE TESTING

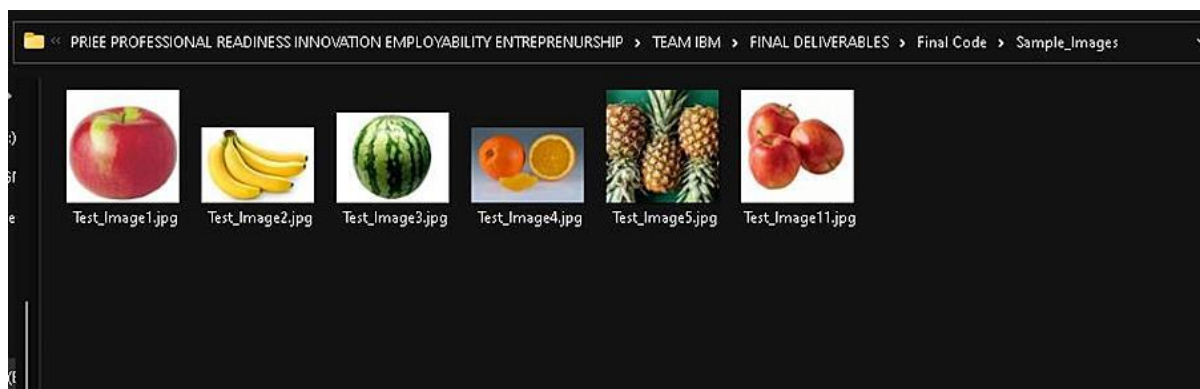
<b>DATE</b>	23 NOVEMBER 2022
<b>TEAM ID</b>	PNT2022TMID36987
<b>PROJECT TITLE</b>	AI POWERED NUTRITION ANALYST FOR FITNESS ENTHUSIASTS
<b>MAXIMUM MARKS</b>	10 MARKS

## 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

```

Flask - app.py
Project: app.py
Sample_Images
Test_Image1.jpg
Test_Image2.jpg
requirements.txt
0.html
main.css
style.css
main.js
app.py
home.html
image.html
imageprediction.html

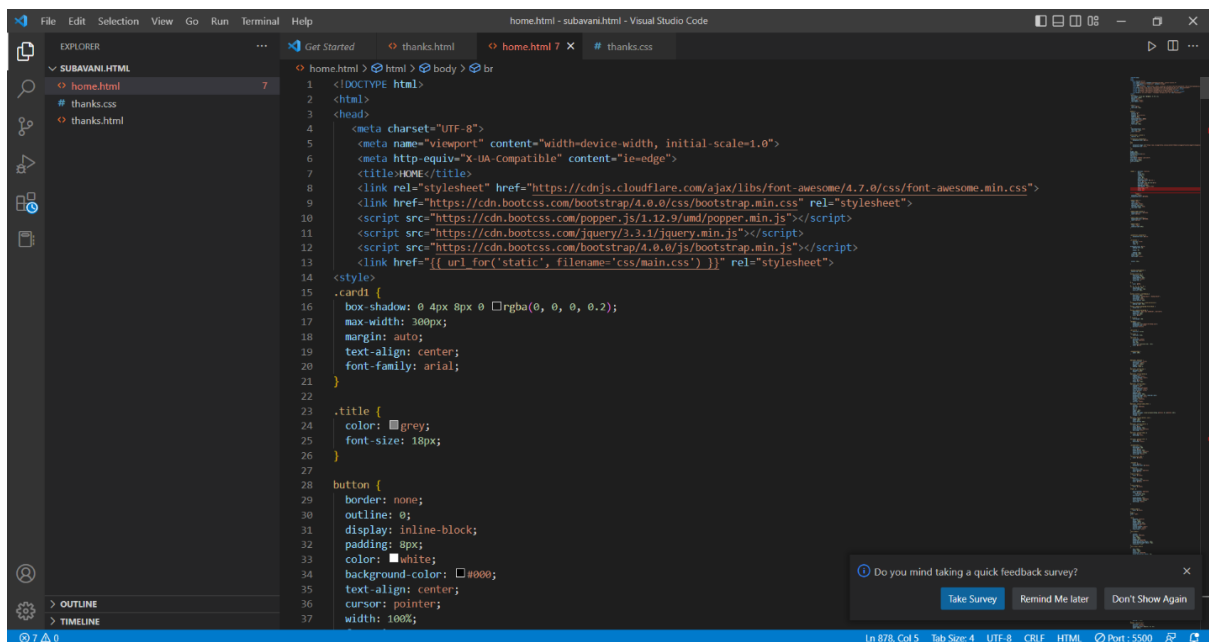
model.load_model('nutrition.h5')
print("Loaded model from disk")

To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
Loaded model from disk
* Restarting with stat
* Serving Flask app 'app' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
2022-11-13 14:47:13.521039: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll'; dLError: cuda
2022-11-13 14:47:13.823308: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dLError if you do not have a GPU set up on your machine.
Loaded model from disk
2022-11-13 15:03:52.074467: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'nvcuda.dll'; dLError: nvcuda.dll
2022-11-13 15:03:52.769818: W tensorflow/stream_executor/cuda/cuda_driver.cc:269] failed call to cuInit: UNKNOWN ERROR (303)
2022-11-13 15:03:54.596275: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:169] retrieving CUDA diagnostic information for host: LAPTOP-E5IM4603
2022-11-13 15:03:54.619299: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:176] hostname: LAPTOP-E5IM4603
2022-11-13 15:03:57.062699: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
* Debugger is active!
* Debugger PIN: 589-305-535
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
  
```

## Future Scope

- Offers payment convenience in personal training subscriptions.
- Useful articles.
- Video instructions.
- Diet Plans.
- Individual progress tracking.

## SCREENSHOTS:



## Data Collection

Download the dataset [here](#)

```
[ ] from google.colab import drive
    drive.mount('/content/drive')

Mounted at /content/drive

[ ] cd /content/drive/MyDrive/Colab Notebooks

/content/drive/MyDrive/Colab Notebooks

[ ] # Unzipping the dataset
    !unzip 'Dataset.zip'
```

## Image Preprocessing

```
[ ] from keras.preprocessing.image import ImageDataGenerator
```

## Image Data Augmentation

```
[ ] 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

```
▶ 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

```
[ ] 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

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

```
▶ classifier.summary()
```

Model: "sequential\_1"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 62, 62, 32)	896

### 5. Configure The Learning Process

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

### 6. Train The Model

```
[ ] 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

```
[ ] classifier.save('nutrition.h5')
```

