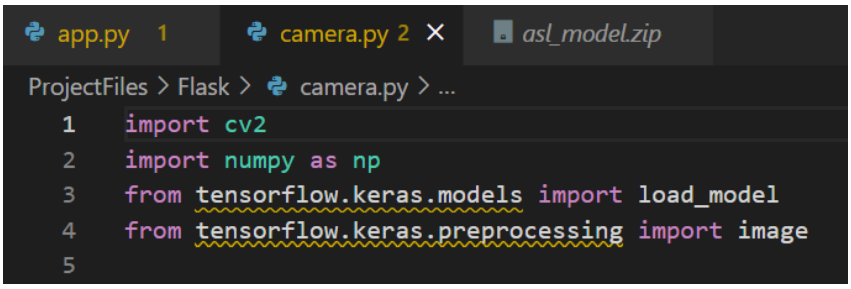
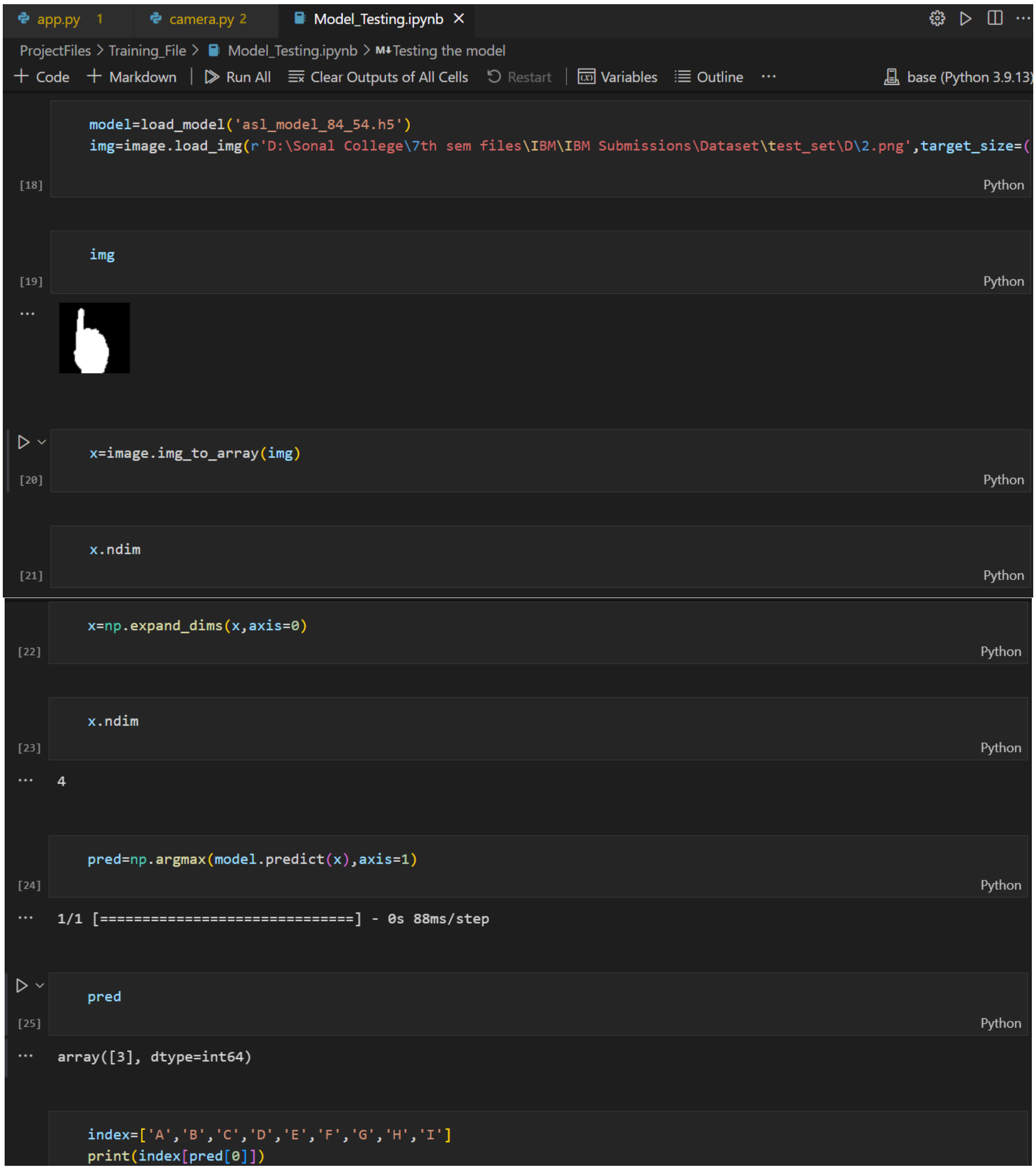
BUILD A FLASK APPLICATION

Step 1: Load the required packages



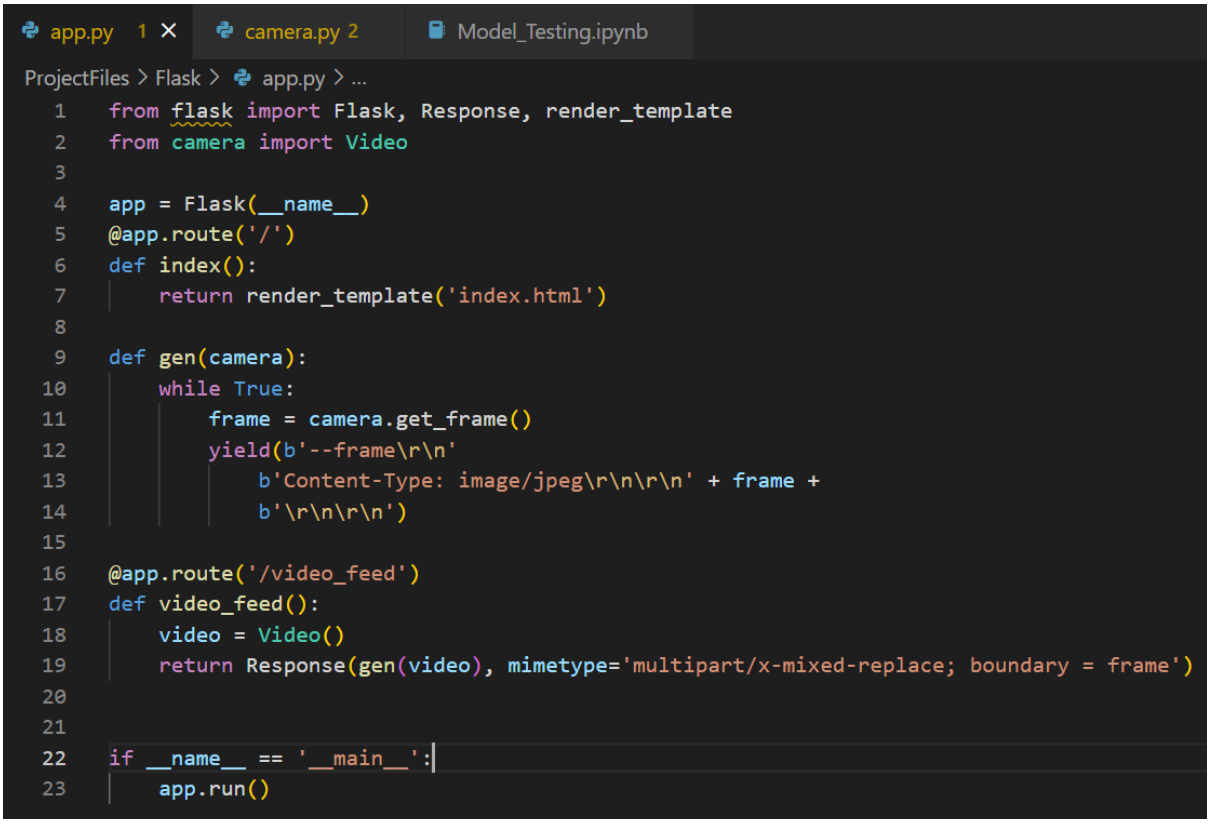
```python
import cv2
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
```

Step 2: Initialize graph, load the model, initialize the flask app and load the video graph element is required to work with tensorflow. So, graph element is created explicitly.



```
[18]
model=load_model('asl_model_84_54.h5')
img=image.load_img(r'D:\Sonal College\7th sem files\IBM\IBM Submissions\Dataset\test_set\D\2.png',target_size=(

[19]
img

[20]
x=image.img_to_array(img)

[21]
x.ndim

[22]
x=np.expand_dims(x,axis=0)

[23]
x.ndim

4

[24]
pred=np.argmax(model.predict(x),axis=1)

1/1 [==============================] - 0s 88ms/step

[25]
pred

array([3], dtype=int64)

index=['A','B','C','D','E','F','G','H','I']
print(index[pred[0]])
```

Step 3: Configure the home page



```python
from flask import Flask, Response, render_template
from camera import Video

app = Flask(__name__)
@app.route('/')
def index():
    return render_template('index.html')

def gen(camera):
    while True:
        frame = camera.get_frame()
        yield(b'--frame\r\n'
            b'Content-Type: image/jpeg\r\n\r\n' + frame +
            b'\r\n\r\n')

@app.route('/video_feed')
def video_feed():
    video = Video()
    return Response(gen(video), mimetype='multipart/x-mixed-replace; boundary = frame')


if __name__ == '__main__':
    app.run()
```