











The screenshot shows an IDE window titled "Flask - home.py". The left sidebar displays a project structure for "Flask" located at "C:\Users\THALAPATHI GOPIN\Documents\Flask". The files listed include "app.py", "capp.py", "fruit.h5", "home.html", "home.py", "IBM-fruit.h5", "IBM-vegetable.h5", "ibmapp.py", "precautions - fruits.xlsx", "precautions - veg.xlsx", "predict.html", "static-20221111T065041Z-001.zip", "templates-20221111T064428Z-001.zip", "uploads-20221111T064449Z-001.zip", and "vegetable.h5". There are also sections for "External Libraries" and "Scratches and Consoles".

The main editor area shows the code for "home.py". The code defines a Flask application with a route for prediction. It handles POST requests by saving an uploaded image, loading it, and using a model to predict the class. The prediction results are then used to read data from an Excel file to provide a caution.

```
1 @app.route('prediction')
2 def prediction():
3     return render_template('predict.html')
4 @app.route('/predict', methods=['post'])
5 def predict():
6     if request.method == 'POST':
7         f=request.files['image']
8         basepath = os.path.dirname(__file__)
9         file_path = os.path.join(
10             basepath, 'uploads', secure_filename(f.filename))
11         f.save(file_path)
12         img = image.load_img(file_path, target_size=(128, 128))
13         x = image.img_to_array(img)
14         x = np.expand_dims(x, axis=0)
15         plant(plant)
16         if(plant=="vegetable"):
17             preds = model.predict_classes(x)
18             print(preds)
19             df=pd.read_excel('precautions - veg.xlsx')
20             print(df.iloc[preds[0]]['caution'])
21         else:
22             preds = model1.predict_classes(x)
23             df=pd.read_excel('precautions - fruits.xlsx')
24             print(df.iloc[preds[0]]['caution'])
```