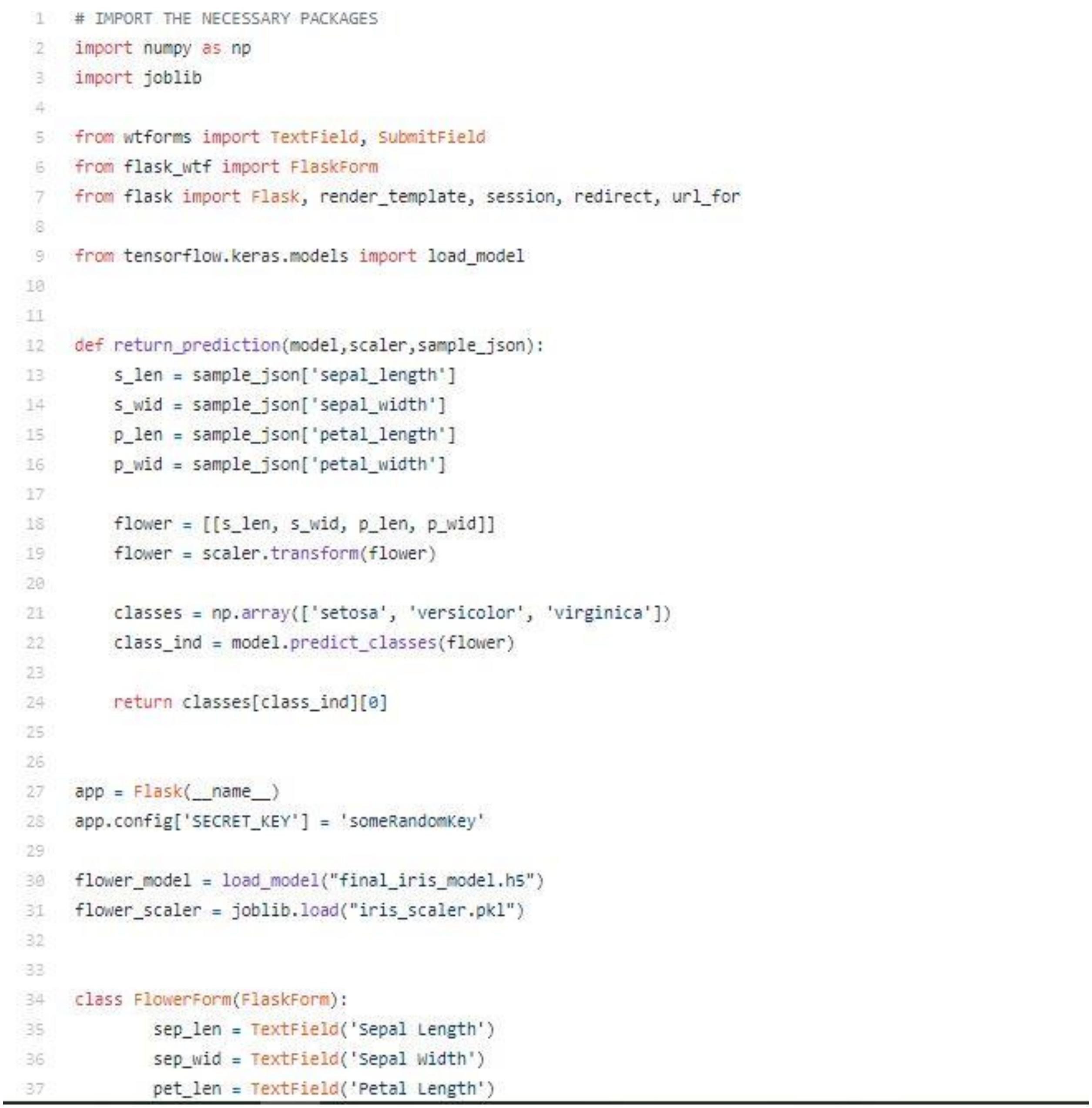
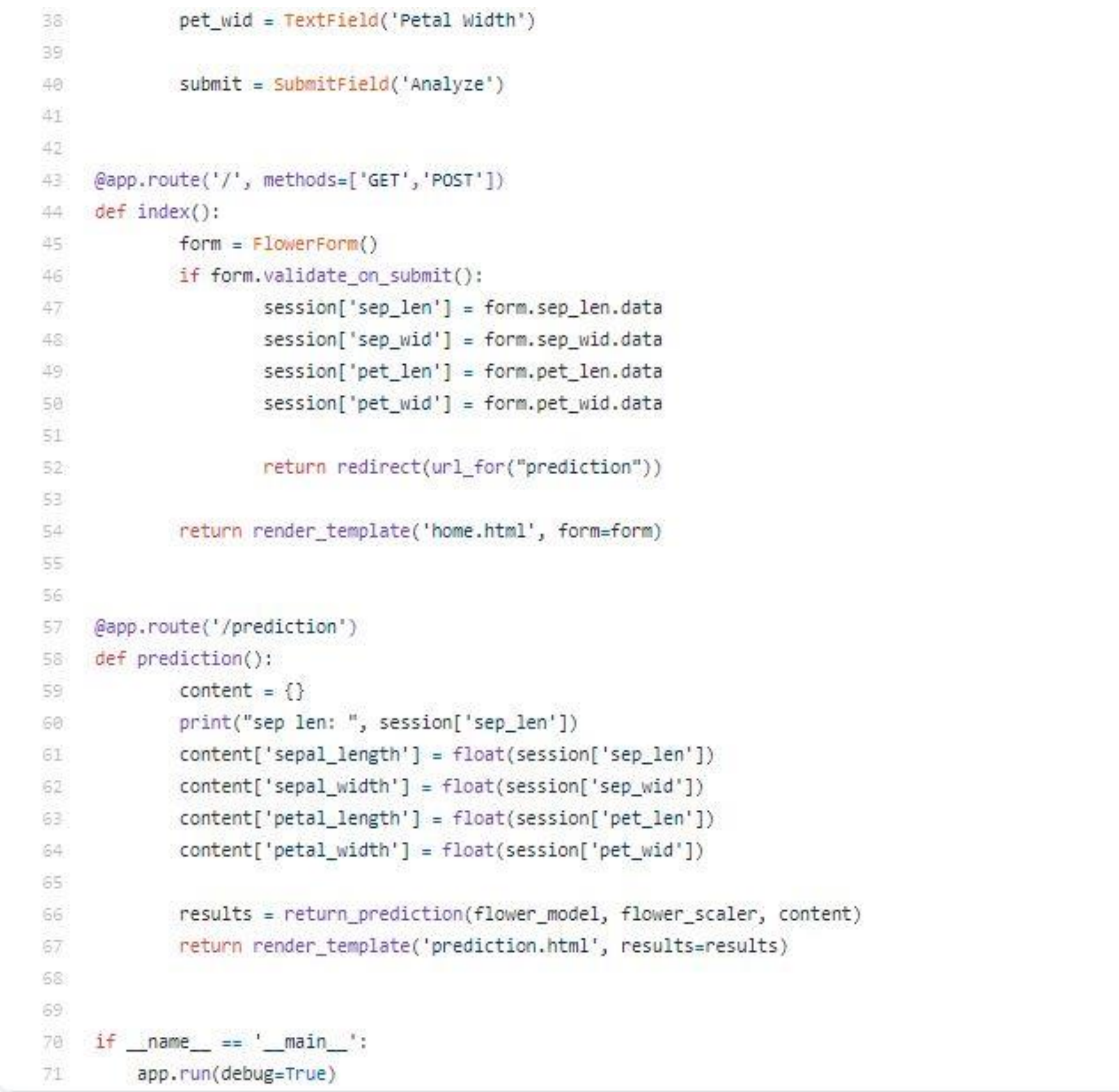
MODEL DEPLOYMENT ON FLASK

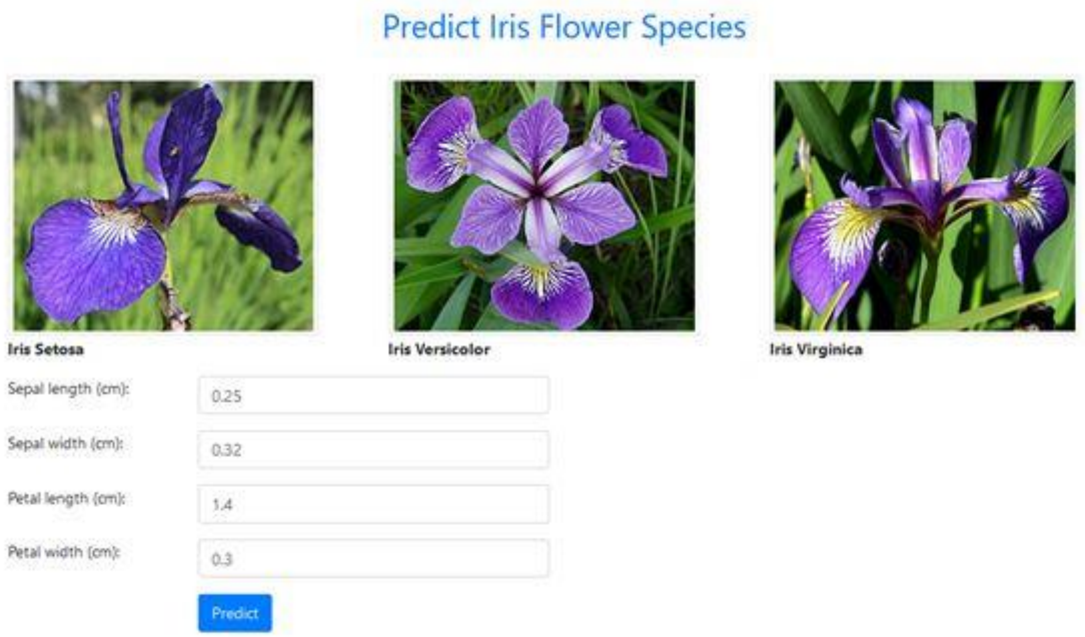
Name: Cyprian Nyamwaro
Internship Batch: LISUM01
Submission date: August 10, 2021
Storage location: https://github.com/cnyamwaro



```
# IMPORT THE NECESSARY PACKAGES
import numpy as np
import joblib

from wtforms import TextField, SubmitField
from flask_wtf import FlaskForm
from flask import Flask, render_template, session, redirect, url_for

from tensorflow.keras.models import load_model


def return_prediction(model,scaler,sample_json):
    s_len = sample_json['sepal_length']
    s_wid = sample_json['sepal_width']
    p_len = sample_json['petal_length']
    p_wid = sample_json['petal_width']

    flower = [[s_len, s_wid, p_len, p_wid]]
    flower = scaler.transform(flower)

    classes = np.array(['setosa', 'versicolor', 'virginica'])
    class_ind = model.predict_classes(flower)

    return classes[class_ind][0]


app = Flask(__name__)
app.config['SECRET_KEY'] = 'someRandomKey'

flower_model = load_model("final_iris_model.h5")
flower_scaler = joblib.load("iris_scaler.pkl")


class FlowerForm(FlaskForm):
        sep_len = TextField('Sepal Length')
        sep_wid = TextField('Sepal Width')
        pet_len = TextField('Petal Length')
```



```
        pet_wid = TextField('Petal Width')

        submit = SubmitField('Analyze')


@app.route('/', methods=['GET','POST'])
def index():
        form = FlowerForm()
        if form.validate_on_submit():
                session['sep_len'] = form.sep_len.data
                session['sep_wid'] = form.sep_wid.data
                session['pet_len'] = form.pet_len.data
                session['pet_wid'] = form.pet_wid.data

                return redirect(url_for("prediction"))

        return render_template('home.html', form=form)


@app.route('/prediction')
def prediction():
        content = {}
        print("sep len: ", session['sep_len'])
        content['sepal_length'] = float(session['sep_len'])
        content['sepal_width'] = float(session['sep_wid'])
        content['petal_length'] = float(session['pet_len'])
        content['petal_width'] = float(session['pet_wid'])

        results = return_prediction(flower_model, flower_scaler, content)
        return render_template('prediction.html', results=results)


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



Predict Iris Flower Species

Iris Setosa

Iris Versicolor

Iris Virginica

Sepal length (cm): 0.25

Sepal width (cm): 0.32

Petal length (cm): 1.4

Petal width (cm): 0.3