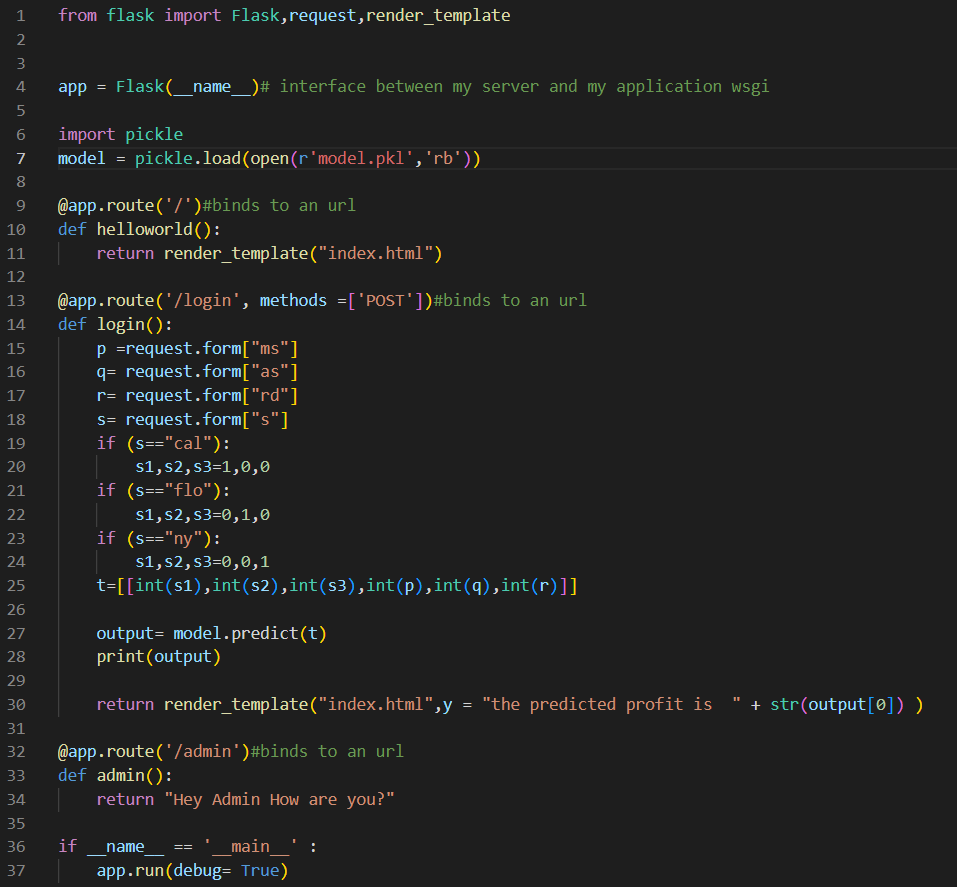
**Flask Document for ADS 50\_Startups Multi Linear Regression**

Create a file app.py(Python scripting file): Write the below code in app.py file to run web app for Startups profitability predictor project:

****

**Explanation:**

Line1: from flask import Flask,request,render\_template



Here, in this line we are importing necessary packages from module flask, flask is imported to create a web application using the Flask microframework. render\_template is a Flask function from the flask.templating package. The Flask request object contains the data that the client (eg a browser) has sent to your app - ie the URL parameters, any POST data, etc. The requests library is for your app to make HTTP request to other sites, usually APIs. It makes an outgoing request and returns the response from the external site.’

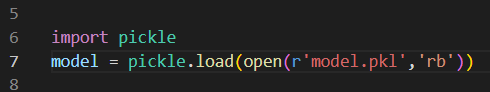
Line 2: app = Flask(\_\_name\_\_)



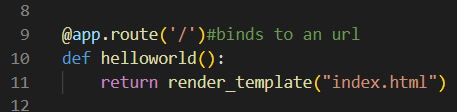
The above snippet code line initializes a Flask application by creating an instance of the Flask class with the name of the current module passed as the argument. The \_\_name\_\_ variable is a special variable in Python that holds the name of the current module, which is usually the name of the Python file that is being executed. When the script is run, \_\_name\_\_ will be equal to '\_\_main\_\_', indicating that the script is being run as the main program. By passing \_\_name\_\_ as the argument to the Flask constructor, the application's name will be set to the name of the current module. This allows Flask to locate the application's resources, such as templates and static files, relative to the module's directory.

Line 6: import pickle

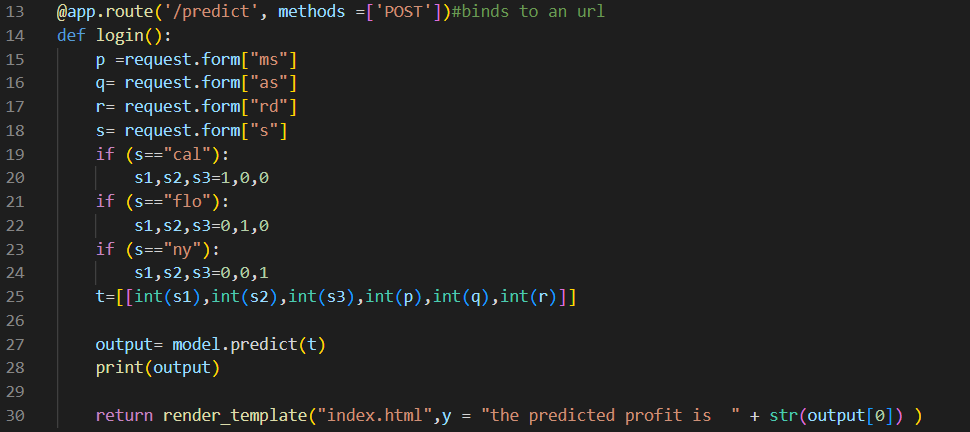
Line 7: model = pickle.load(open(r'model.pkl','rb'))



Python pickle module is used for serializing and de-serializing a Python object structure. Any object in Python can be pickled so that it can be saved on disk. What pickle does is that it “serializes” the object first before writing it to file. Pickling is a way to convert a python object (list, dict, etc.) into a character stream. Pickle.load(open(filename,’rb’)) is used to load the model saved as .pkl file. Here we are going to load model.pkl file as our machine Learning model and store it to a variable model.



The above snippet code defines a route for the root directory ('/') of the web application using the @app.route() decorator in Flask. The index function is called when the user visits the root directory. This function returns the result of the render\_template function, which generates an HTML page using a Jinja2 template called index.html. The render\_template function is a utility function provided by Flask that allows you to render a template with variables passed in as arguments. In this case, the index.html template is rendered without any variables, so the resulting HTML page will be static. Overall, this route is the default route for the web application and will display the index.html page when the user visits the root directory



Line 13: In the above snippet code defines another route for the ‘/predict’ endpoint of the web application. The methods argument specifies that the route should handle both GET and POST requests.

Line 14: login() function is called whenever user submits the form with user input.

Line 15: taking marketing spend input from user form using request and storing in variable p

Line 16: taking administration spend input from user form using request and storing in variable q

Line 15: taking R&D spend input from user form using request and storing in variable r

Line 15: taking State spend input from user form using request, here we’ll get input from the dropdown out of 3 states one will selected but while building the model we have applied encoding techniques.

State had 3 categories – California, Florida, New York. After one Label Encoding California is encode to ‘0’ , Florida is encoded to ‘1’ , New York is encoded to ‘2’.

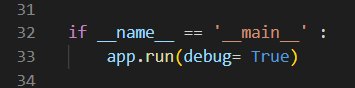
Line 19-24: Then after One hot encoding state becomes 3 different categorical columns, California is encode to ‘1,0,0’ , Florida is encoded to ‘0,1,0’ , New York is encoded to ‘0,0,1’. Here also if user selected California then state s is converted to 3 columns s1,s3,s3 having values 1,0,0 . also if user selected Florida then state s is converted to 3 columns s1,s3,s3 having values 0,1,0. also if user selected New York then state s is converted to 3 columns s1,s3,s3 having values 0,0 ,1.

Line25: passing the user inputs from the form i.e., p, q,r,s1,s2,s3 to a 2d array t.

Line 27: inputs from the user stored in t are then passed to the ML model that was loaded in model variable and storing the predicted output in the variable output.

Line 28: print the predicted output

Line 30: Finally, the text variable is returned to the user as the HTTP response. The return statement return the predicted output along with the text string as y and rendered onto the ‘index.html’ template using render\_html



Line 32: In the above snippet code block is used to run the Flask application when the script is executed as the main program. The if \_\_name\_\_ == '\_\_main\_\_': statement is a common Python idiom that checks if the script is being run as the main program. If it is, the code inside the if block is executed.

Finally the run() method of Flask class runs the application on the local development server.

Line 33: app.run(debug = True)

By default debug is False but here we are passing debug = True. If a Flask Application is having Debug Mode = False then, When you update some code, you need to restart the server for the changes to come upon the web page. This can be quite repetitive since we keep changing and updating our code. So to make coding easy, Flask gives us the Debug Mode. Hence, with the Debug Mode on, all the application code changes will get updated right away in the development stage, eliminating the need to restart the server.

**Run the application**

● Open the anaconda prompt from the start menu.

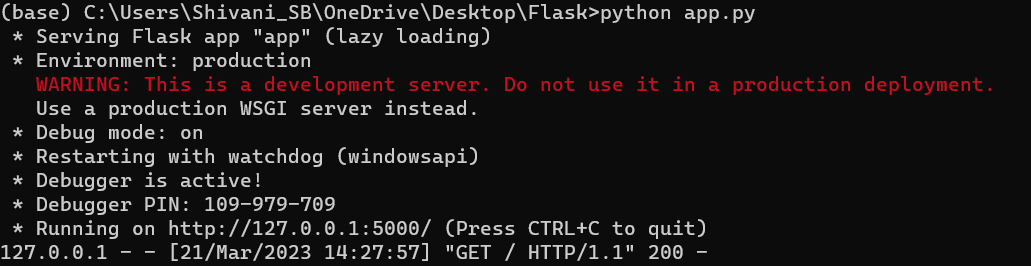
● Navigate to the folder where your app.py file is present.

● Now type “python app.py” command.

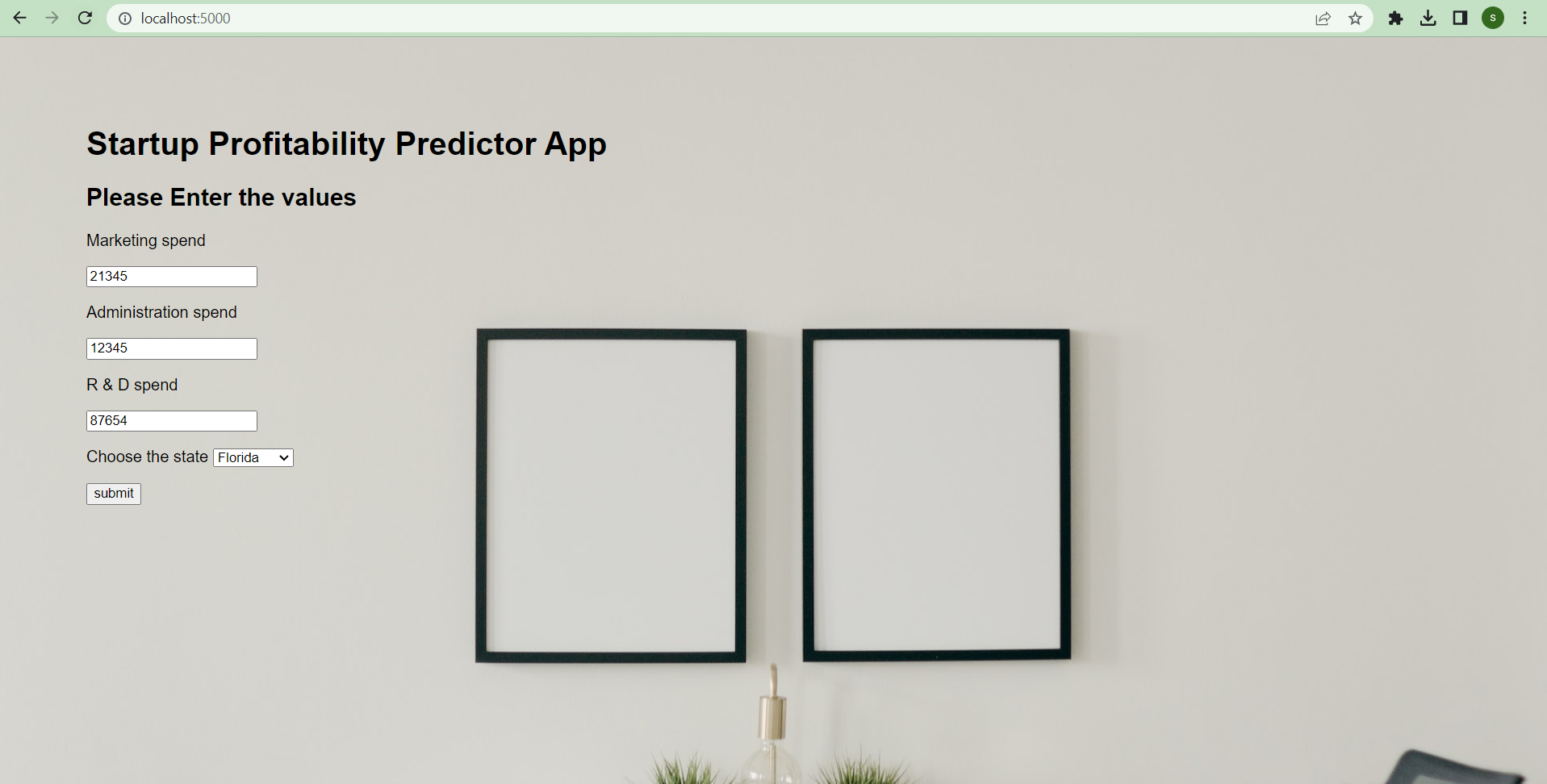
● It will show the local host where your app is running on http://127.0.0.1.5000/

● Copy that local host URL and open that URL in the browser.

Then it will run on localhost:5000



Navigate to the localhost (http://127.0.0.1:5000/) where you can view your web page. You need to give the input values.



Predicted output will be shown on the UI.

