

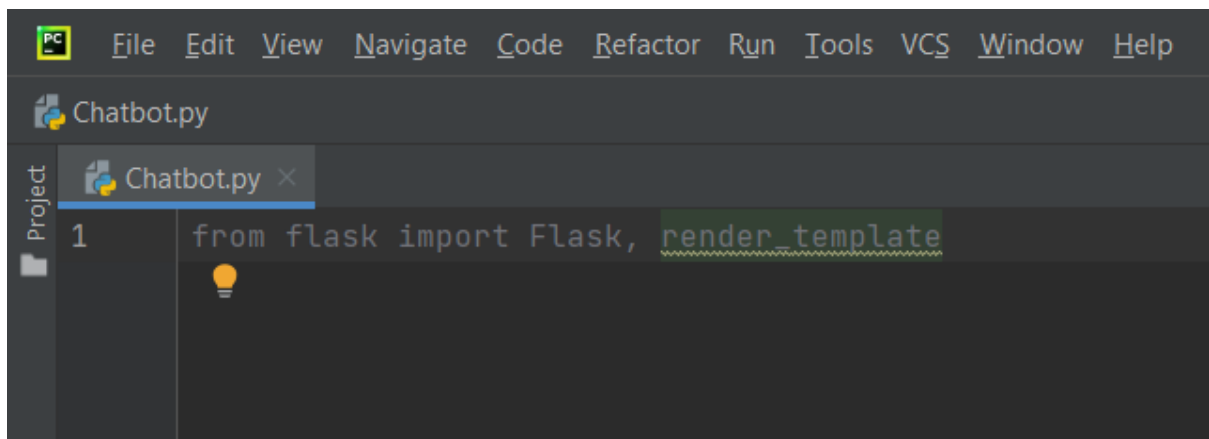
| | |
|--------------|---|
| Team ID | PNT2022TMID39294 |
| Project Name | AI-BASED DISCOURSE FOR BANKING INDUSTRY |

Creating Assistant & Integrate With Flask Web Page

Build Python Code

Progress 1: Importing Libraries

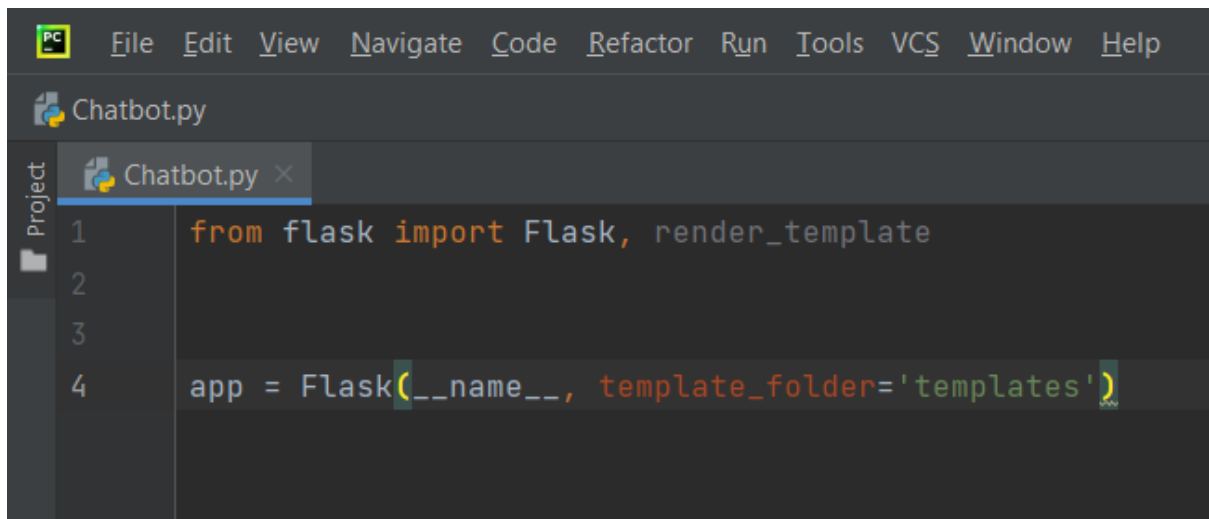
The first step is usually importing the libraries that will be needed in the program.



```
File Edit View Navigate Code Refactor Run Tools VCS Window Help
Chatbot.py
Project Chatbot.py x
1 from flask import Flask, render_template
```

Progress 2: Creating our flask application and loading

Importing the flask module into the project is mandatory. An object of the Flask class is our WSGI application. The Flask constructor takes the name of the current module (`__name__`).

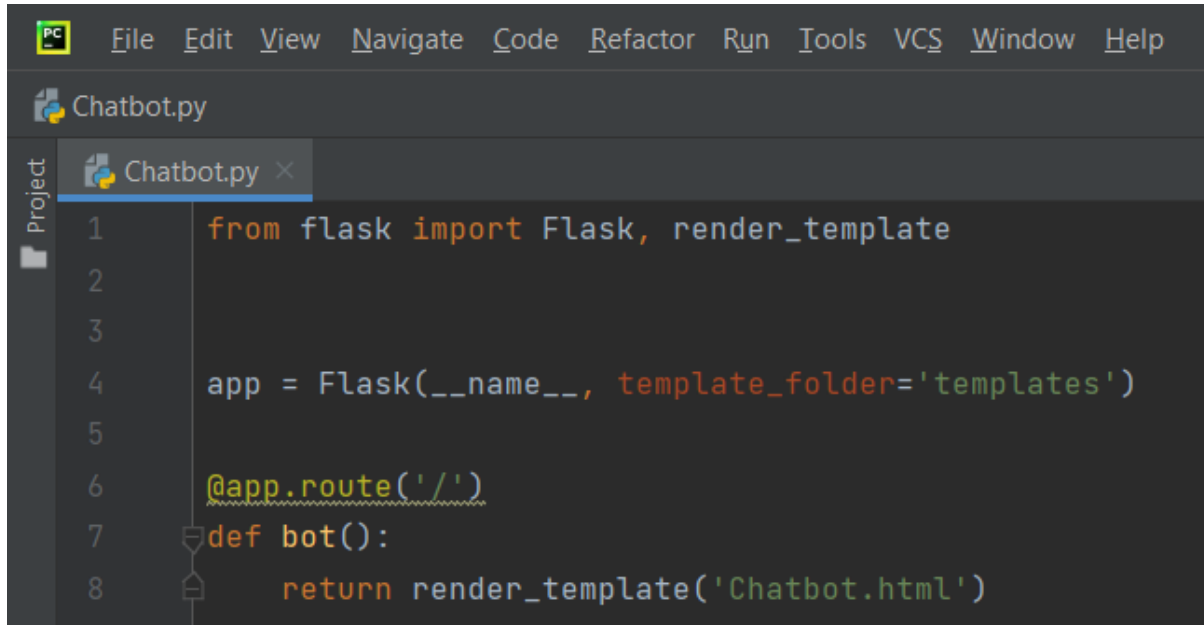


```
File Edit View Navigate Code Refactor Run Tools VCS Window Help
Chatbot.py
Project Chatbot.py x
1 from flask import Flask, render_template
2
3
4 app = Flask(__name__, template_folder='templates')
```

Progress 3: Routing to the Html Page

Here, the declared constructor is used to route to the HTML page created earlier.

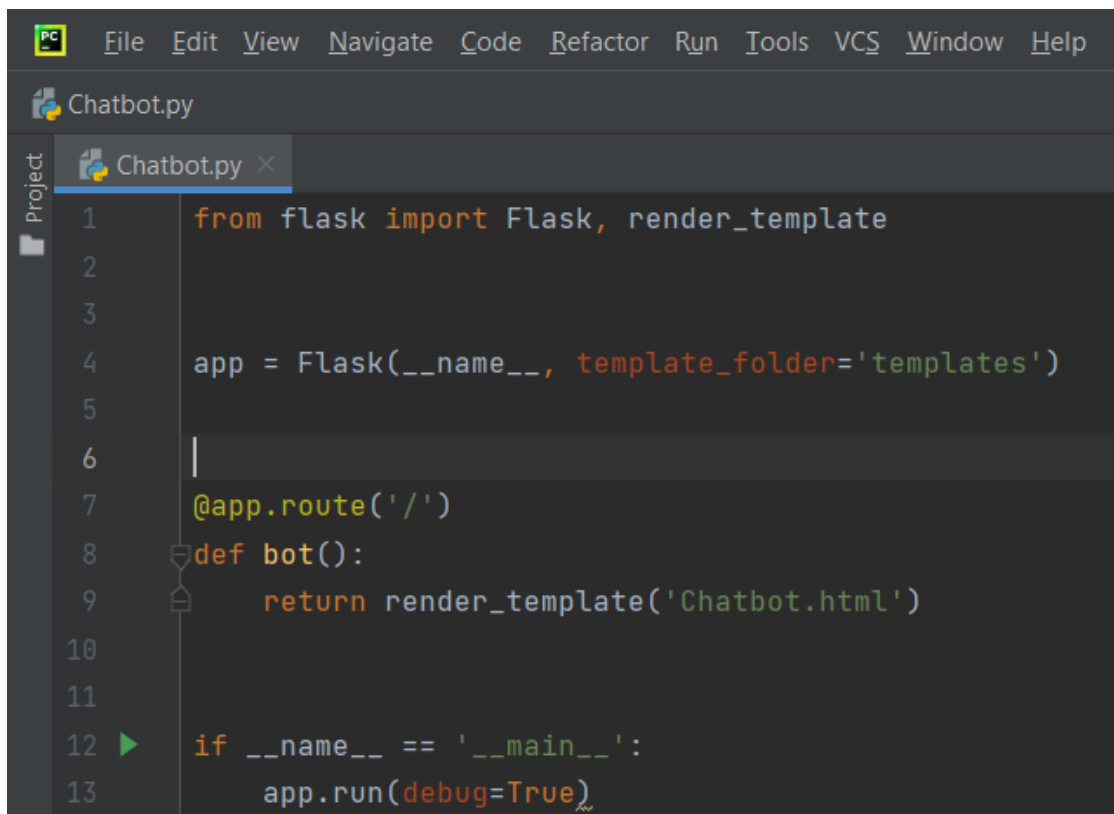
The '/' route is bound with the bot function. Hence, when the home page of a web server is opened in the browser, the HTML page will be rendered.

A screenshot of an IDE window titled 'Chatbot.py'. The code is as follows:

```
1 from flask import Flask, render_template
2
3
4 app = Flask(__name__, template_folder='templates')
5
6 @app.route('/')
7 def bot():
8     return render_template('Chatbot.html')
```

Progress 4: Main Function

This is used to run the application in localhost.

A screenshot of an IDE window titled 'Chatbot.py'. The code is as follows:

```
1 from flask import Flask, render_template
2
3
4 app = Flask(__name__, template_folder='templates')
5
6
7 @app.route('/')
8 def bot():
9     return render_template('Chatbot.html')
10
11
12 if __name__ == '__main__':
13     app.run(debug=True)
```

