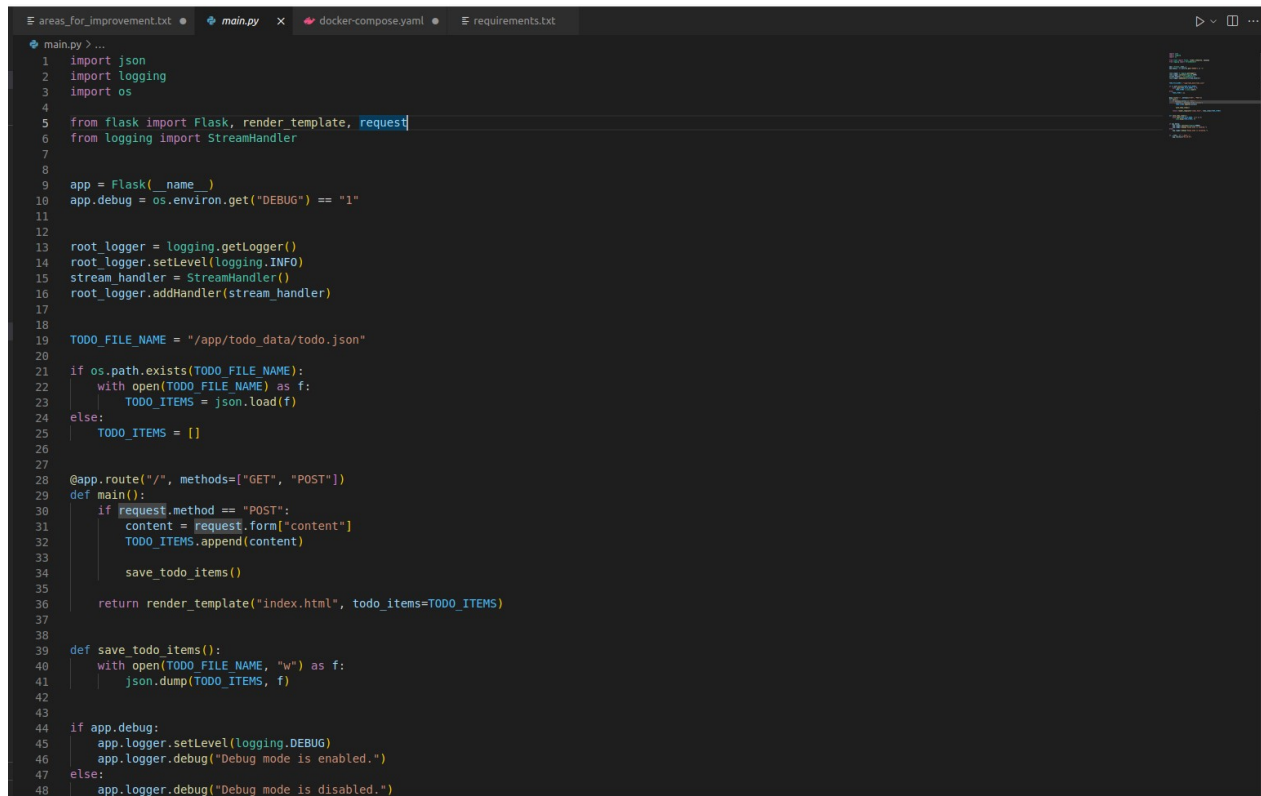


Talha Khan (2303.009.KHI.DEG)

Muhammad Moiz Khan (2303.022.KHI.DEG)

ASSIGNMENT 4.4



```
1 import json
2 import logging
3 import os
4
5 from flask import Flask, render_template, request
6 from logging import StreamHandler
7
8
9 app = Flask(__name__)
10 app.debug = os.environ.get("DEBUG") == "1"
11
12
13 root_logger = logging.getLogger()
14 root_logger.setLevel(logging.INFO)
15 stream_handler = StreamHandler()
16 root_logger.addHandler(stream_handler)
17
18
19 TODO_FILE_NAME = "/app/todo_data/todo.json"
20
21 if os.path.exists(TODO_FILE_NAME):
22     with open(TODO_FILE_NAME) as f:
23         TODO_ITEMS = json.load(f)
24 else:
25     TODO_ITEMS = []
26
27
28 @app.route("/", methods=["GET", "POST"])
29 def main():
30     if request.method == "POST":
31         content = request.form["content"]
32         TODO_ITEMS.append(content)
33         save_todo_items()
34
35     return render_template("index.html", todo_items=TODO_ITEMS)
36
37
38
39 def save_todo_items():
40     with open(TODO_FILE_NAME, "w") as f:
41         json.dump(TODO_ITEMS, f)
42
43
44 if app.debug:
45     app.logger.setLevel(logging.DEBUG)
46     app.logger.debug("Debug mode is enabled.")
47 else:
48     app.logger.debug("Debug mode is disabled.")
```

1) First we import the necessary modules and libraries required for the application. These include `json`, `logging` and `os` for handling JSON data, logging events, and interacting with the operating system.

2) Then we create the Flask application using `Flask(__name__)`. The `__name__` variable represents the name of the current module. Additionally, the `app.debug` flag is set to `True` or `False` based on the value of the environment variable `DEBUG`.

3) Then We setup the logging configuration to write log messages. The root logger is accessed using `logging.getLogger()` and set to log messages at the `INFO` level. A `StreamHandler` is created to handle the log messages, and it is added to the root logger.

4) Then in the next step, the code checks if a JSON file named "todo.json" exists at the specified path (`TODO_FILE_NAME`). If the file exists, its content is loaded into the `TODO_ITEMS` list using `json.load()`. If the file doesn't exist, an empty list is assigned to `TODO_ITEMS`.

5) In the next step of code, we defined the `/` route using the `@app.route()` decorator. This route accepts both GET and POST requests. If a POST request is received, the content from the submitted form is retrieved using `request.form['content']` and appended to the `TODO_ITEMS` list. The `save_todo_items()` function is called to save the updated list to the JSON file. Finally, the `render_template()` function is used to render the "index.html" template and pass the `todo_items` data to it.

6) The `save_todo_items()` function in the code writes the contents of the `TODO_ITEMS` list to the JSON file specified by `TODO_FILE_NAME`. It opens the file in write mode (`"w"`) and uses `json.dump()` to write the list to the file in JSON format.

7) If the application is running in debug mode (`app.debug == True`), the logging level for the Flask application logger is set to `DEBUG`, and a debug message is logged. Otherwise, a debug message is logged to indicate that debug mode is disabled.

```
49
50
51 if __name__ == "__main__":
52     app.run(host="0.0.0.0")
53
```

8) Finally, atlast the application is run by calling `app.run()` with the host set to `"0.0.0.0"` to allow incoming connections from any IP address.

```
docker-compose.yml
1 version: '3'
2 services:
3   app:
4     build:
5       context: .
6       dockerfile: Dockerfile
7     ports:
8       - '5004-5006:5000'
9     volumes:
10      - todo_data:/app/todo_data
11     deploy:
12       replicas: 3
13
14 volumes:
15   todo_data:
16
17
```

```
Dockerfile
1 FROM python:3.8-slim-buster
2 WORKDIR /home/app/
3 COPY ./ /home/app/
4 ENV PYTHONPATH=${PYTHONPATH}:/home/app/
5
6 VOLUME /app/todo_data
7
8 COPY requirements.txt .
9
10 RUN pip install --no-cache-dir -r requirements.txt
11 CMD ["gunicorn", "main:app", "-b", "0.0.0.0:5000"]
12
13
14
```

```

• (base) muhmmadmoizkhan@all-MS-7D35:~/Downloads/day_4_best_practices_tools_common_pitfalls/app_that_doesnt_follow_best_practices$ docker build --tag app_that_doesnt_follow_b
est_practices .
[+] Building 6.6s (10/10) FINISHED

=> [internal] load build definition from Dockerfile
0.0s
=> => transferring dockerfile: 300B
0.0s
=> [internal] load .dockerignore
0.0s
=> => transferring context: 2B
0.0s
=> [internal] load metadata for docker.io/library/python:3.8-slim-buster
1.8s
=> [1/5] FROM docker.io/library/python:3.8-slim-buster@sha256:89ad1c2cd09bda5bc85ada7eb93b5db57d32dc0105b7c942d272d68f376f67c3
0.0s
=> [internal] load build context
0.0s
=> => transferring context: 6.62kB
0.0s
=> CACHED [2/5] WORKDIR /home/app/
0.0s
=> [3/5] COPY ./ /home/app/
0.0s
=> [4/5] COPY requirements.txt .
0.0s
=> [5/5] RUN pip install --no-cache-dir -r requirements.txt
4.3s
=> exporting to image
0.2s
=> => exporting layers
0.2s
=> writing image sha256:63e39b62970e1682e9e8d36eb00b95f54155f8eedb00a066a4a4de7990a246ef
0.0s
=> naming to docker.io/library/app_that_doesnt_follow_best_practices
0.0s

```

```

(base) muhmmadmoizkhan@all-MS-7D35:~/Downloads/day_4_best_practices_tools_common_pitfalls/app_that_doesnt_follow_best_practices$ docker run -p 5005:5004 app_that_doesnt_fol
low_best_practices

[2023-05-16 09:49:49 +0000] [1] [INFO] Starting gunicorn 20.1.0
[2023-05-16 09:49:49 +0000] [1] [INFO] Listening at: http://0.0.0.0:5000 (1)
[2023-05-16 09:49:49 +0000] [1] [INFO] Using worker: sync
[2023-05-16 09:49:49 +0000] [8] [INFO] Booting worker with pid: 8
[2023-05-16 09:49:56 +0000] [1] [INFO] Handling signal: winch
[2023-05-16 09:49:56 +0000] [1] [INFO] Handling signal: winch
[2023-05-16 10:54:17 +0000] [1] [INFO] Handling signal: winch
[2023-05-16 10:54:17 +0000] [1] [INFO] Handling signal: winch
[2023-05-16 10:54:30 +0000] [1] [INFO] Handling signal: winch
[2023-05-16 10:54:30 +0000] [1] [INFO] Handling signal: winch
[2023-05-16 10:54:37 +0000] [1] [INFO] Handling signal: winch
[2023-05-16 10:54:37 +0000] [1] [INFO] Handling signal: winch
[2023-05-16 11:30:28 +0000] [1] [INFO] Handling signal: winch
[2023-05-16 11:30:28 +0000] [1] [INFO] Handling signal: winch
[2023-05-16 11:30:29 +0000] [1] [INFO] Handling signal: winch
[2023-05-16 11:30:30 +0000] [1] [INFO] Handling signal: winch
[2023-05-16 11:30:30 +0000] [1] [INFO] Handling signal: winch
[2023-05-16 11:30:30 +0000] [1] [INFO] Handling signal: winch
[2023-05-16 11:30:31 +0000] [1] [INFO] Handling signal: winch
[2023-05-16 11:30:31 +0000] [1] [INFO] Handling signal: winch

```

OUTPUT:

Add TODO item

Please provide the TODO item content

Submit

TODO items

ali

ali

Talha

Talha

Talha

Moiz

Moiz

