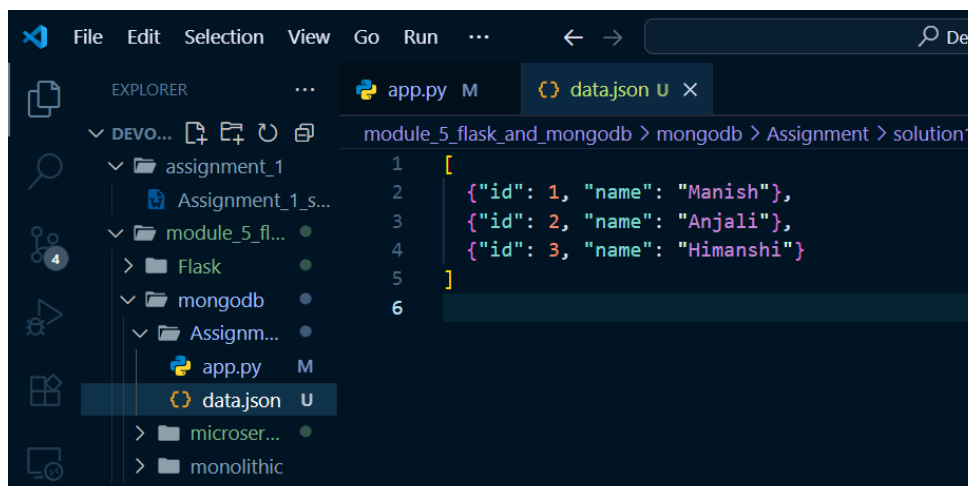


# Assignment 3

1. Create a Flask application with an `/api` route. When this route is accessed, it should return a JSON list. The data should be stored in a backend file, read from it, and sent as a response.

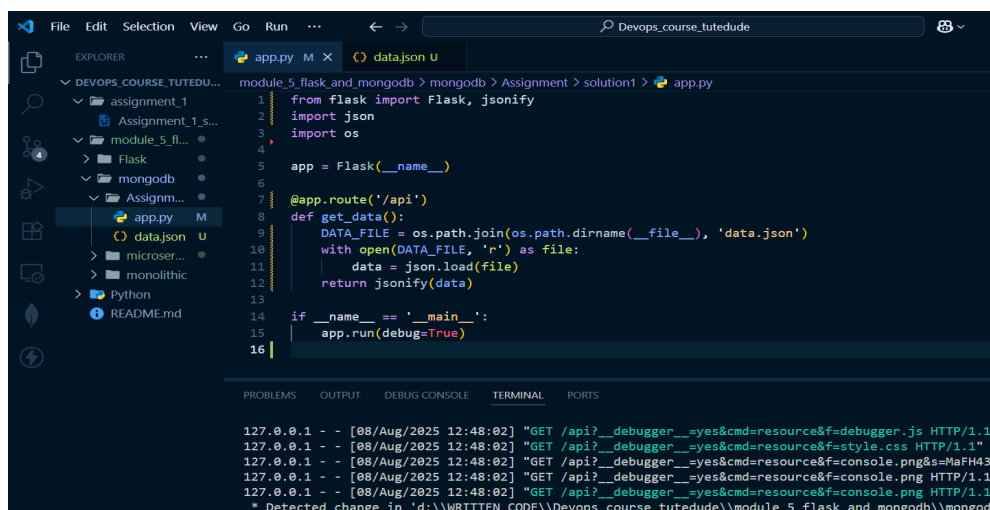
Solution1: **data.json**



The screenshot shows the VS Code editor with the `data.json` file open. The Explorer sidebar on the left shows the project structure: `assignment_1` (containing `Assignment_1_s...`), `module_5_fl...` (containing `Flask`, `mongodb`, `Assignm...`, `app.py`, and `data.json`), `microser...`, and `monolithic`. The `data.json` file contains the following JSON list:

```
1  [
2    {"id": 1, "name": "Manish"},
3    {"id": 2, "name": "Anjali"},
4    {"id": 3, "name": "Himanshi"}
5  ]
6
```

**app.py**

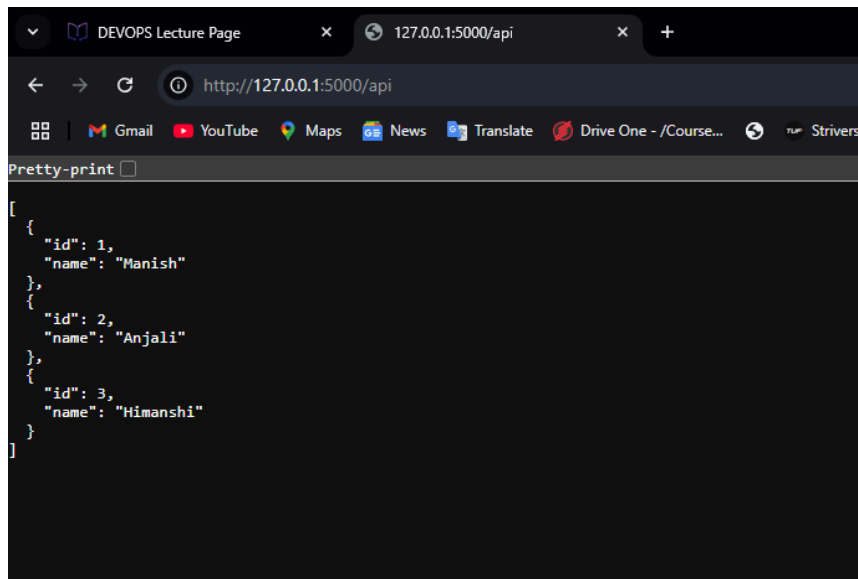


The screenshot shows the VS Code editor with the `app.py` file open. The Explorer sidebar on the left shows the project structure: `DEVOPS_COURSE_TUTEDU...` (containing `assignment_1`, `module_5_fl...`, `microser...`, and `monolithic`), `Python`, and `README.md`. The `app.py` file contains the following Python code:

```
1  from flask import Flask, jsonify
2  import json
3  import os
4
5  app = Flask(__name__)
6
7  @app.route('/api')
8  def get_data():
9      DATA_FILE = os.path.join(os.path.dirname(__file__), 'data.json')
10     with open(DATA_FILE, 'r') as file:
11         data = json.load(file)
12     return jsonify(data)
13
14 if __name__ == '__main__':
15     app.run(debug=True)
16
```

The terminal at the bottom shows the output of the application running on `127.0.0.1` at port `5000`. It displays several GET requests to the `/api` endpoint, each returning a JSON response. A message at the bottom indicates a detected change in the file path.

## Output :

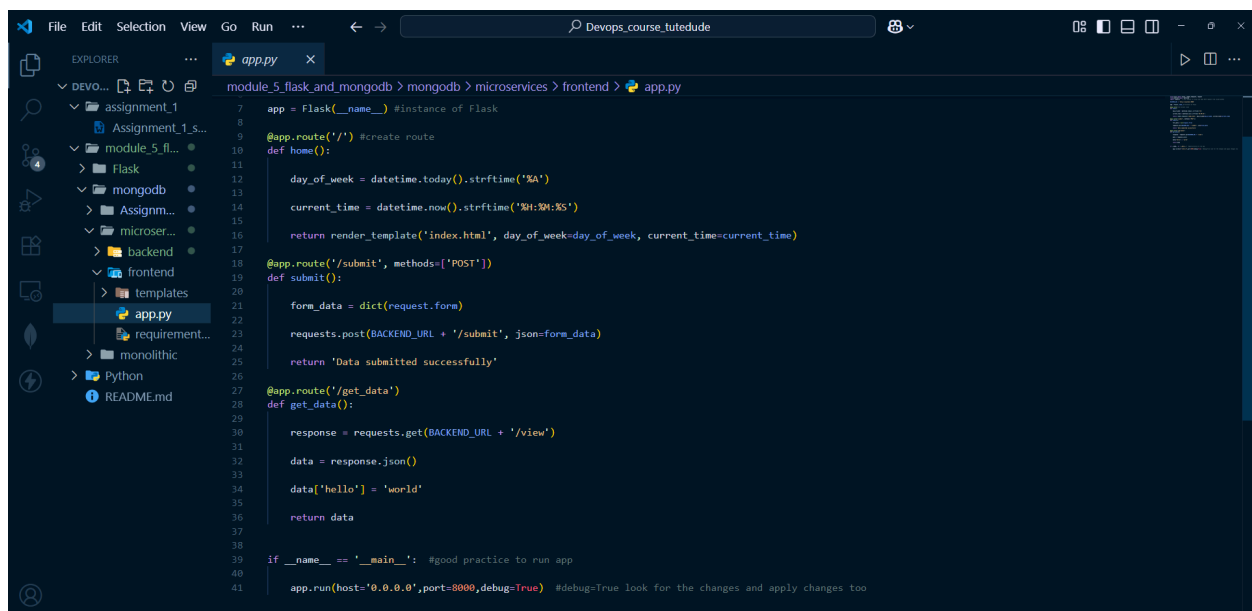


```
[
  {
    "id": 1,
    "name": "Manish"
  },
  {
    "id": 2,
    "name": "Anjali"
  },
  {
    "id": 3,
    "name": "Himanshi"
  }
]
```

2. Create a form on the frontend that, when submitted, inserts data into MongoDB Atlas. Upon successful submission, the user should be redirected to another page displaying the message **"Data submitted successfully"**. If there's an error during submission, display the error on the same page without redirection.

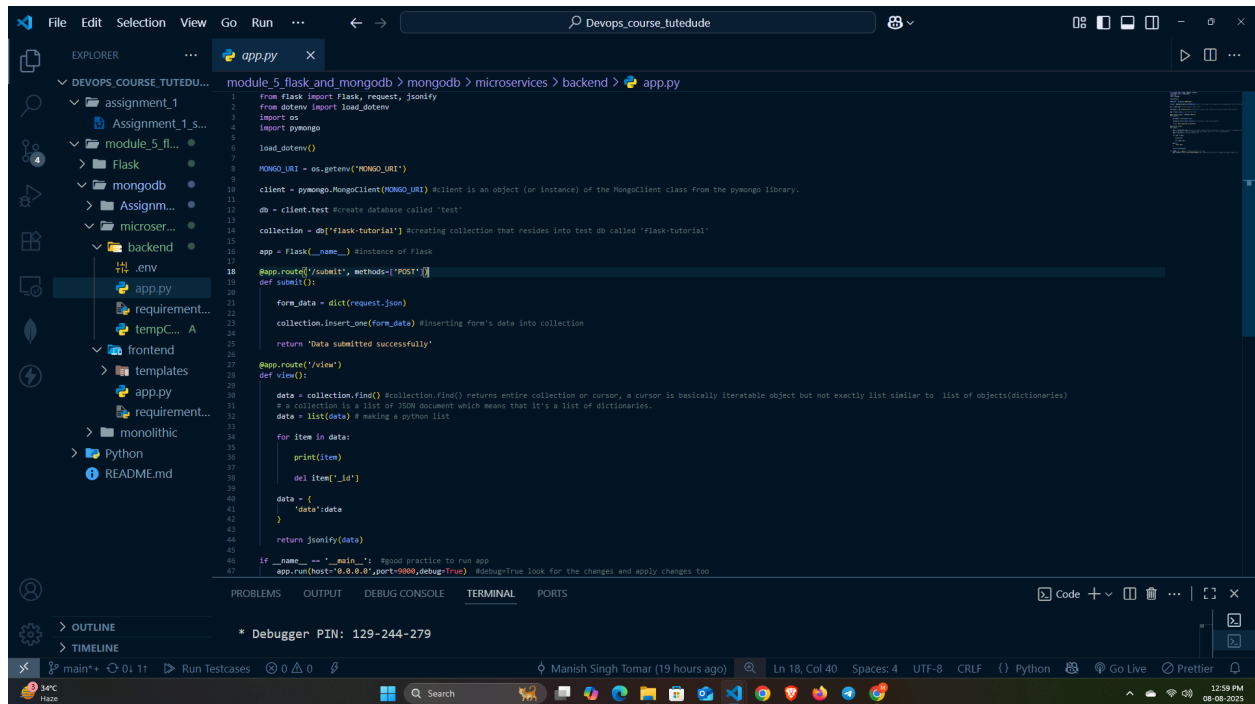
Solution 2:

## Frontend:

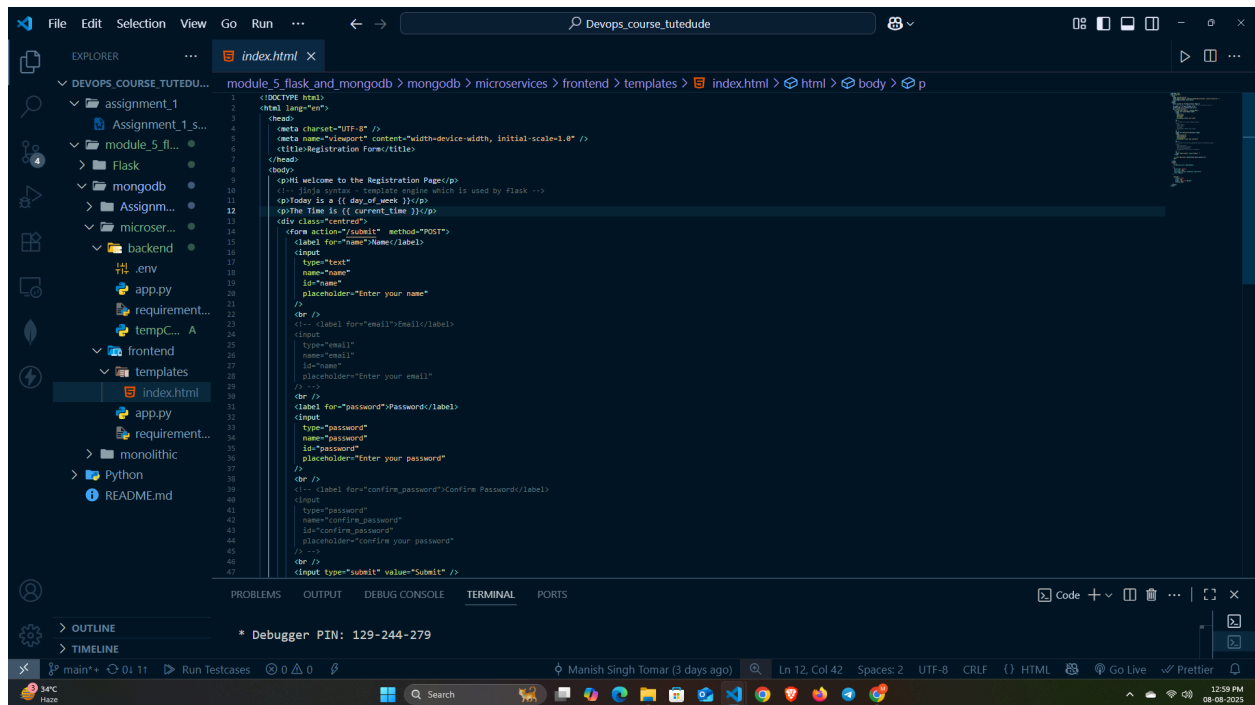


```
1  app = Flask(__name__) #instance of Flask
2
3  @app.route('/') #create route
4  def home():
5
6      day_of_week = datetime.today().strftime('%A')
7      current_time = datetime.now().strftime('%H:%M:%S')
8
9      return render_template("index.html", day_of_week=day_of_week, current_time=current_time)
10
11 @app.route('/submit', methods=['POST'])
12 def submit():
13
14     form_data = dict(request.form)
15
16     requests.post(BACKEND_URL + '/submit', json=form_data)
17
18     return 'Data submitted successfully'
19
20 @app.route('/get_data')
21 def get_data():
22
23     response = requests.get(BACKEND_URL + '/view')
24
25     data = response.json()
26
27     data["hello"] = 'world'
28
29     return data
30
31 if __name__ == '__main__': #good practice to run app
32
33     app.run(host='0.0.0.0', port=8000, debug=True) #debugs=True look for the changes and apply changes too
```

## Backend:



```
module_5 flask_and_mongodb > mongodb > microservices > backend > app.py
1 from flask import Flask, request, jsonify
2 from dotenv import load_dotenv
3 import os
4 import pymongo
5 load_dotenv()
6
7
8 MONGO_URI = os.getenv("MONGO_URI")
9
10 client = pymongo.MongoClient(MONGO_URI) #client is an object (or instance) of the MongoClient class from the pymongo library.
11
12 db = client.test #create database called 'test'
13
14 collection = db['flask-tutorial'] #creating collection that resides into test db called 'flask-tutorial'
15
16 app = Flask(__name__) #instance of Flask
17
18 @app.route('/submit', methods=['POST'])
19 def submit():
20     form_data = dict(request.json)
21     collection.insert_one(form_data) #inserting form's data into collection
22     return 'Data submitted successfully'
23
24 @app.route('/view')
25 def view():
26     data = collection.find() #collection.find() returns entire collection or cursor, a cursor is basically iterable object but not exactly list similar to list of objects(dictionaries)
27     # a collection is a list of JSON document which means that it's a list of dictionaries.
28     data = list(data) #making a python list
29
30     for item in data:
31         print(item)
32         del item['_id']
33
34     data = {
35         'data':data
36     }
37     return jsonify(data)
38
39 if __name__ == '__main__': #good practice to run app
40     app.run(host='0.0.0.0', port=9000, debug=True) #debug=True look for the changes and apply changes too
```



```
module_5 flask_and_mongodb > mongodb > microservices > frontend > templates > index.html > html > body > p
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6     <title>Registration Form</title>
7   </head>
8   <body>
9     <p>Hi welcome to the Registration Page</p>
10    <!-- single syntax -> template engine which is used by flask -->
11    <p>Today is <= {{ day_of_week }} /></p>
12    <p>The time is <= {{ current_time }} /></p>
13    <div class="contreg">
14      <form action="/submit" method="POST">
15        <label for="name">Name</label>
16        <input
17          type="text"
18          name="name"
19          id="name"
20          placeholder="Enter your name"
21        />
22        <br />
23        <label for="email">Email</label>
24        <input
25          type="email"
26          name="email"
27          id="email"
28          placeholder="Enter your email"
29        />
30        <br />
31        <label for="password">Password</label>
32        <input
33          type="password"
34          name="password"
35          id="password"
36          placeholder="Enter your password"
37        />
38        <br />
39        <label for="confirm_password">Confirm Password</label>
40        <input
41          type="password"
42          name="confirm_password"
43          id="confirm_password"
44          placeholder="confirm your password"
45        />
46        <br />
47        <input type="submit" value="Submit" />
48      </form>
49    </div>
50  </body>
51 </html>
```

```
1 module_5_flask_and_mongodb > mongodb > microservices > frontend > app.py
2
3 app = Flask(__name__) #instance of flask
4
5 @app.route("/") #create route
6 def home():
7
8     day_of_week = datetime.today().strftime('%A')
9
10    current_time = datetime.now().strftime('%H:%M:%S')
11
12    return render_template("index.html", day_of_week=day_of_week, current_time=current_time)
13
14
15 @app.route("/submit", methods=['POST'])
16 def submit():
17
18     form_data = dict(request.form)
19
20     requests.post(BACKEND_URL + '/submit', json=form_data)
21
22     return "Data submitted successfully"
23
24
25 @app.route("/get_data")
26 def get_data():
27
28     response = requests.get(BACKEND_URL + '/view')
29
30     data = response.json()
31
32     data["hello"] = "world"
33
34     return data
35
36
37 if __name__ == '__main__': #good practice to run app
38
39     app.run(host='0.0.0.0', port=8080, debug=True) #debug=True look for the changes and apply changes too
40
41
```

\* Debugger PIN: 129-244-279

