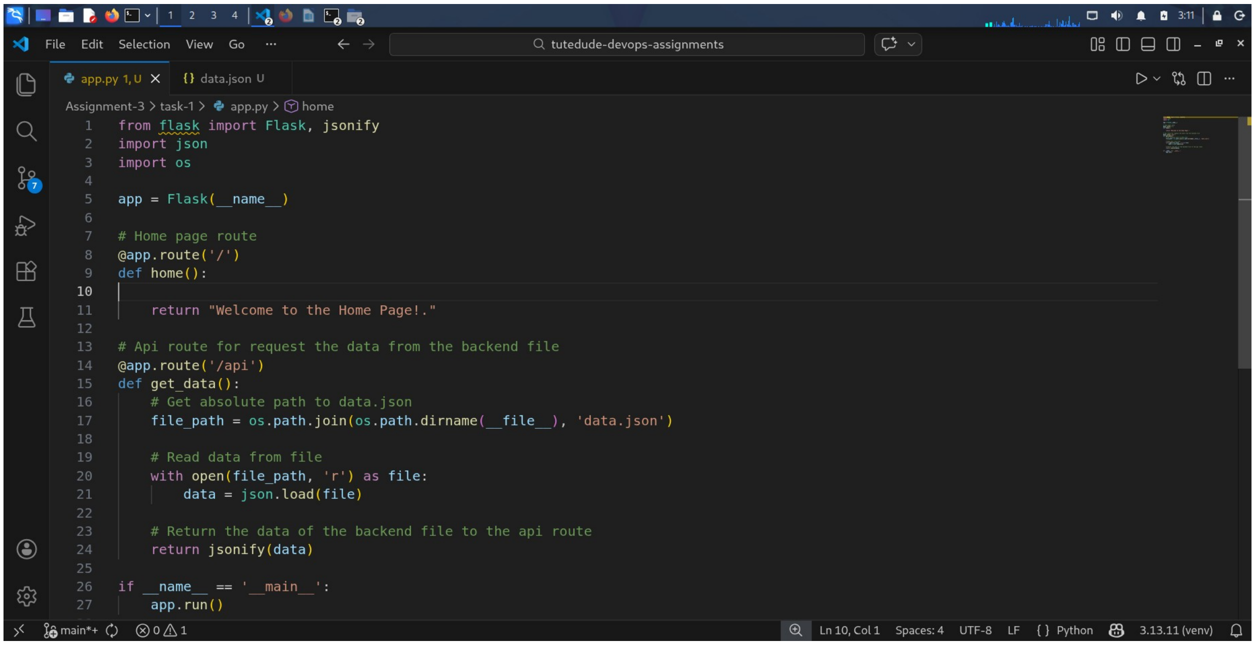
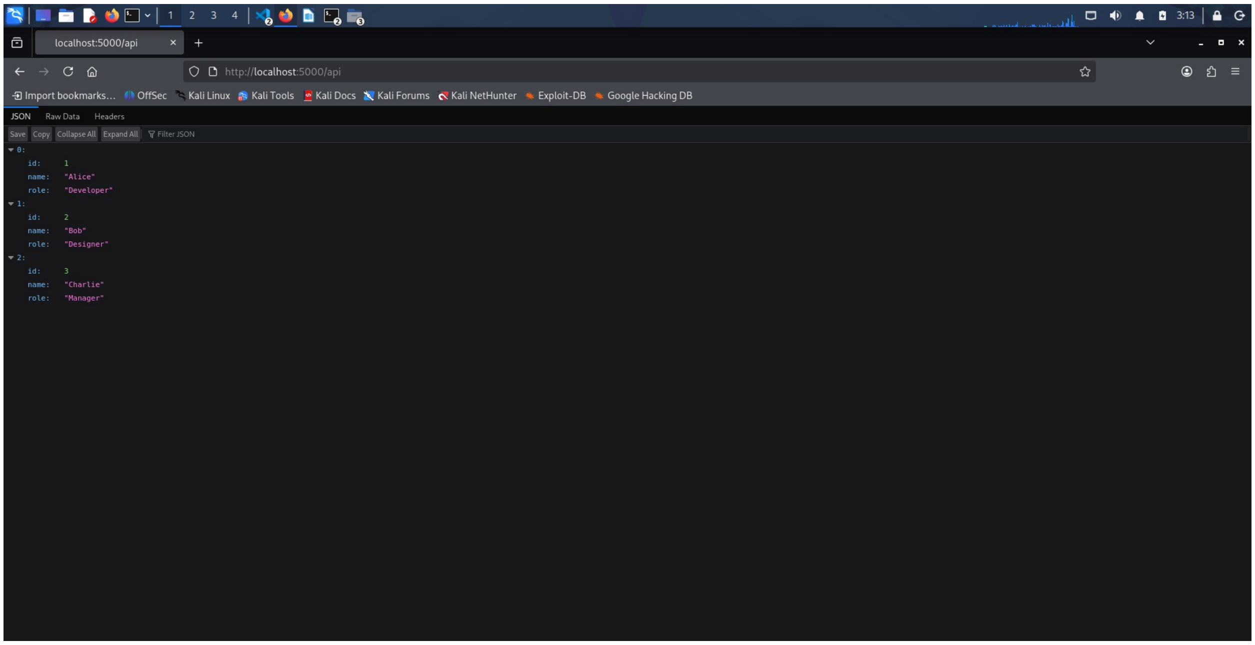
1. Create a Flask application with an `/api` route. When this route is accessed, it should return a JSON list. The data should be stored in a backend file, read from it, and sent as a response.

(https://github.com/aaditya173/tutedude-devops-assignments/tree/main/Assignment-3/task-1)



File Edit Selection View Go ···
tutedude-devops-assignments
app.py 1, U
data.json U
Assignment-3 > task-1 > app.py > home
main*+
Ln 10, Col 1 Spaces: 4 UTF-8 LF { } Python 3.13.11 (venv)

```python
from flask import Flask, jsonify
import json
import os

app = Flask(__name__)

# Home page route
@app.route('/')
def home():

    return "Welcome to the Home Page!."

# Api route for request the data from the backend file
@app.route('/api')
def get_data():
    # Get absolute path to data.json
    file_path = os.path.join(os.path.dirname(__file__), 'data.json')

    # Read data from file
    with open(file_path, 'r') as file:
        data = json.load(file)

    # Return the data of the backend file to the api route
    return jsonify(data)

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

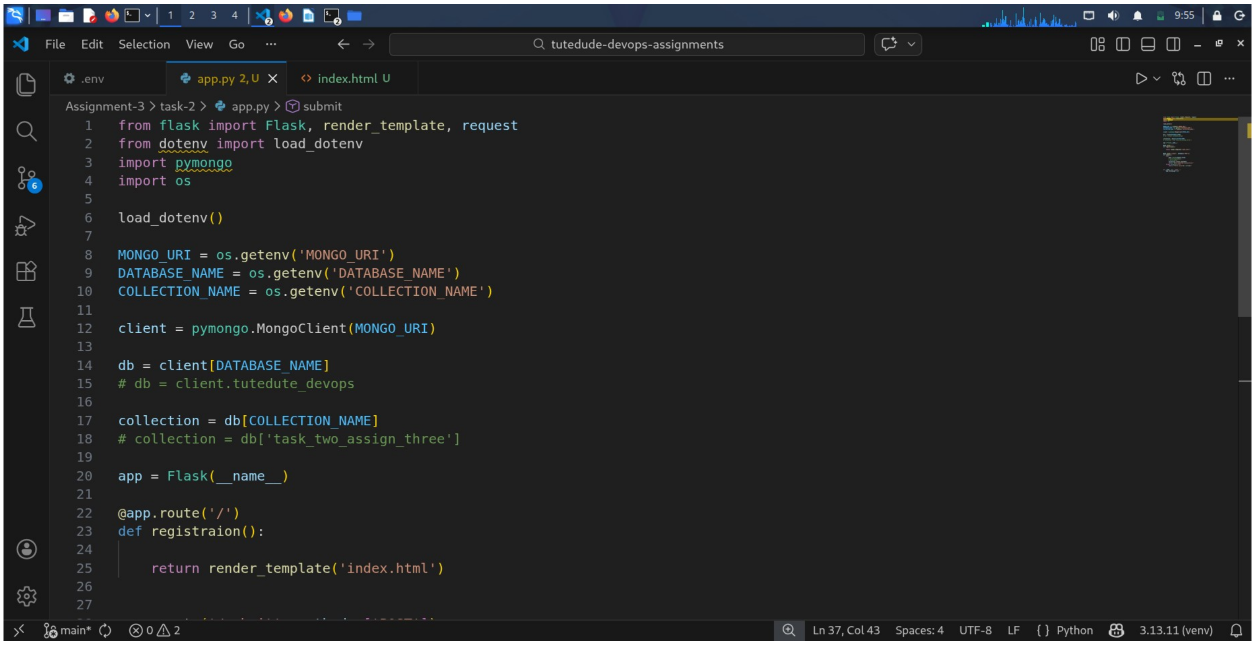


localhost:5000/api
http://localhost:5000/api
Import bookmarks... OffSec Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB
JSON Raw Data Headers
Save Copy Collapse All Expand All Filter JSON

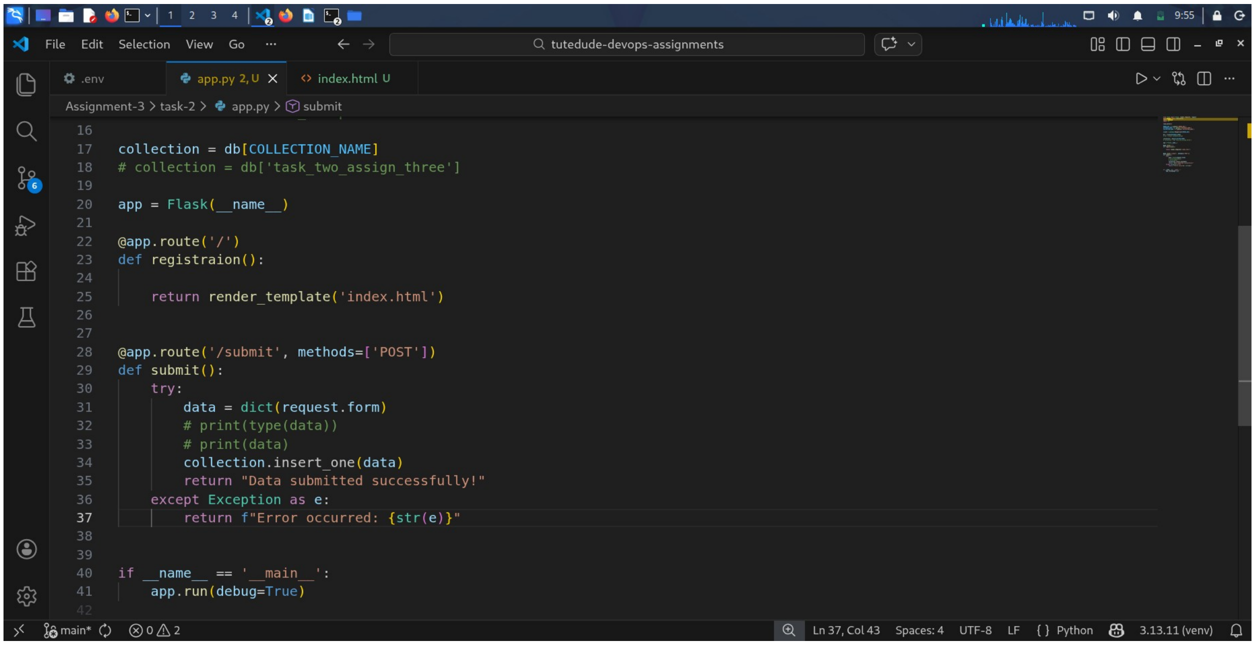
```
▼ 0:
    id:      1
    name:    "Alice"
    role:    "Developer"
▼ 1:
    id:      2
    name:    "Bob"
    role:    "Designer"
▼ 2:
    id:      3
    name:    "Charlie"
    role:    "Manager"
```

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.

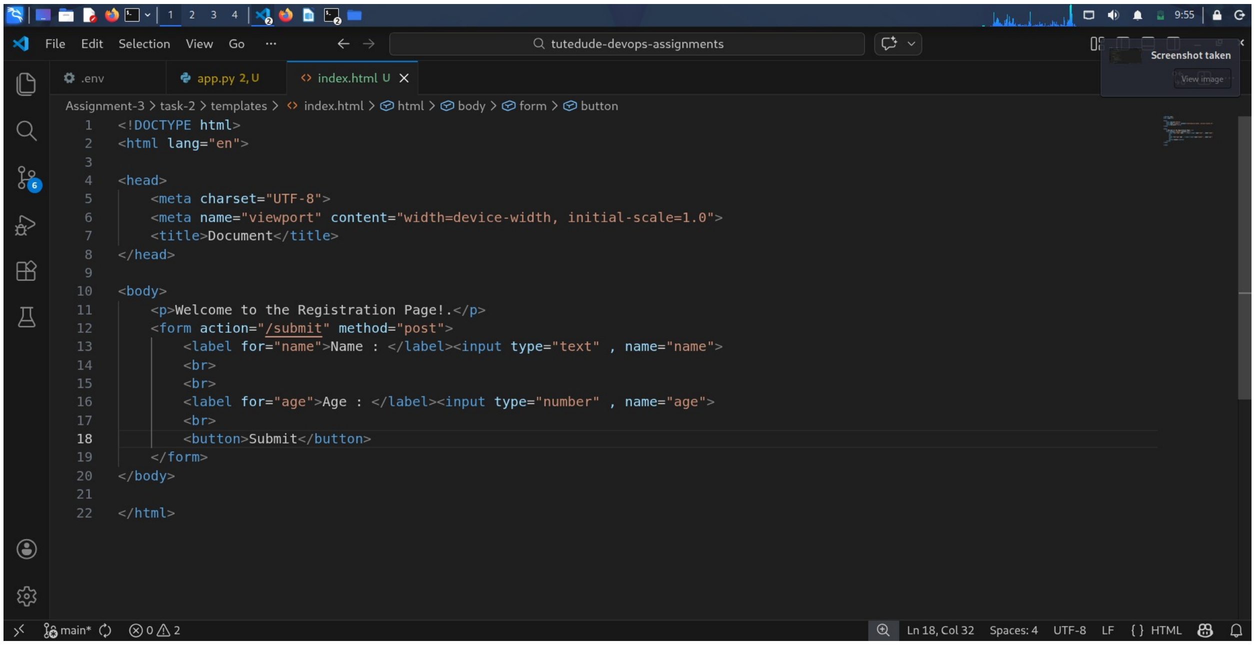
(https://github.com/aaditya173/tutedude-devops-assignments/tree/main/Assignment-3/task-2)


File Edit Selection View Go ···
tutedude-devops-assignments
.env
app.py 2, U
index.html U
Assignment-3 > task-2 > app.py > submit
main*
0 2
Ln 37, Col 43 Spaces: 4 UTF-8 LF { } Python 3.13.11 (venv)

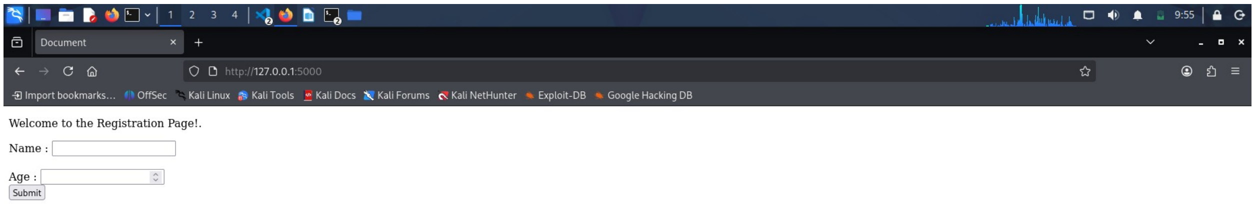
```python
from flask import Flask, render_template, request
from dotenv import load_dotenv
import pymongo
import os

load_dotenv()

MONGO_URI = os.getenv('MONGO_URI')
DATABASE_NAME = os.getenv('DATABASE_NAME')
COLLECTION_NAME = os.getenv('COLLECTION_NAME')

client = pymongo.MongoClient(MONGO_URI)

db = client[DATABASE_NAME]
# db = client.tutedute_devops

collection = db[COLLECTION_NAME]
# collection = db['task_two_assign_three']

app = Flask(__name__)

@app.route('/')
def registraion():

    return render_template('index.html')

```


File Edit Selection View Go ···
tutedude-devops-assignments
.env
app.py 2, U
index.html U
Assignment-3 > task-2 > app.py > submit
main*
0 2
Ln 37, Col 43 Spaces: 4 UTF-8 LF { } Python 3.13.11 (venv)

```python

collection = db[COLLECTION_NAME]
# collection = db['task_two_assign_three']

app = Flask(__name__)

@app.route('/')
def registraion():

    return render_template('index.html')


@app.route('/submit', methods=['POST'])
def submit():
    try:
        data = dict(request.form)
        # print(type(data))
        # print(data)
        collection.insert_one(data)
        return "Data submitted successfully!"
    except Exception as e:
        return f"Error occurred: {str(e)}"


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


File Edit Selection View Go ···
tutedude-devops-assignments
.env
app.py 2, U
index.html U
Screenshot taken
View image
Assignment-3 > task-2 > templates > index.html > html > body > form > button
main*
0 2
Ln 18, Col 32 Spaces: 4 UTF-8 LF { } HTML
9:55

```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>

<body>
    <p>Welcome to the Registration Page!.</p>
    <form action="/submit" method="post">
        <label for="name">Name : </label><input type="text" , name="name">
        <br>
        <br>
        <label for="age">Age : </label><input type="number" , name="age">
        <br>
        <button>Submit</button>
    </form>
</body>

</html>
```

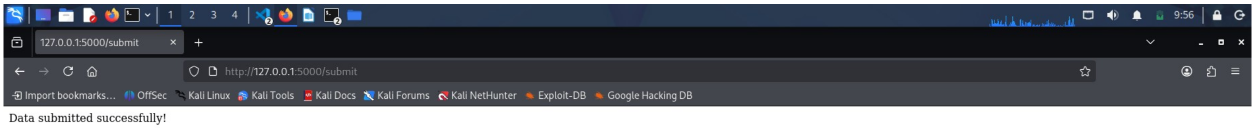

Document
http://127.0.0.1:5000
Import bookmarks... OffSec Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB
9:55

Welcome to the Registration Page!.

Name :

Age :

Submit



Data submitted successfully!