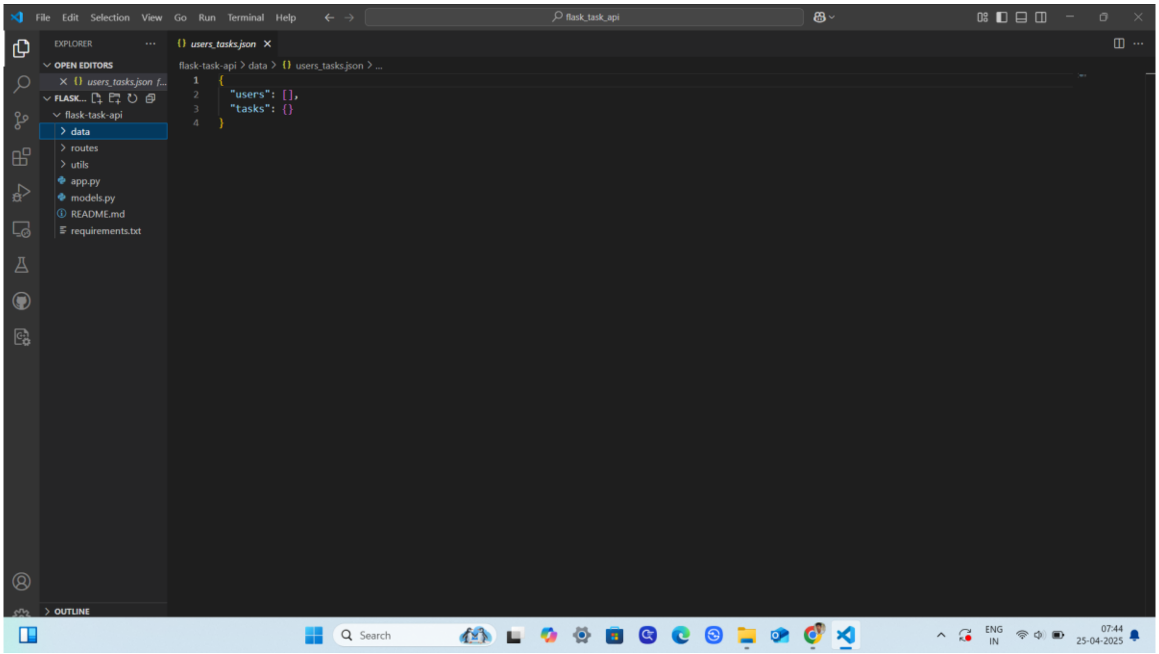
Backend Developer Task Log

This document contains a detailed log of my work on the Backend Developer task for House of MarkTech. Each log entry is listed chronologically with a brief description of the activities completed during that time. Screenshots corresponding to each time period will be added below each entry.

7:44 AM

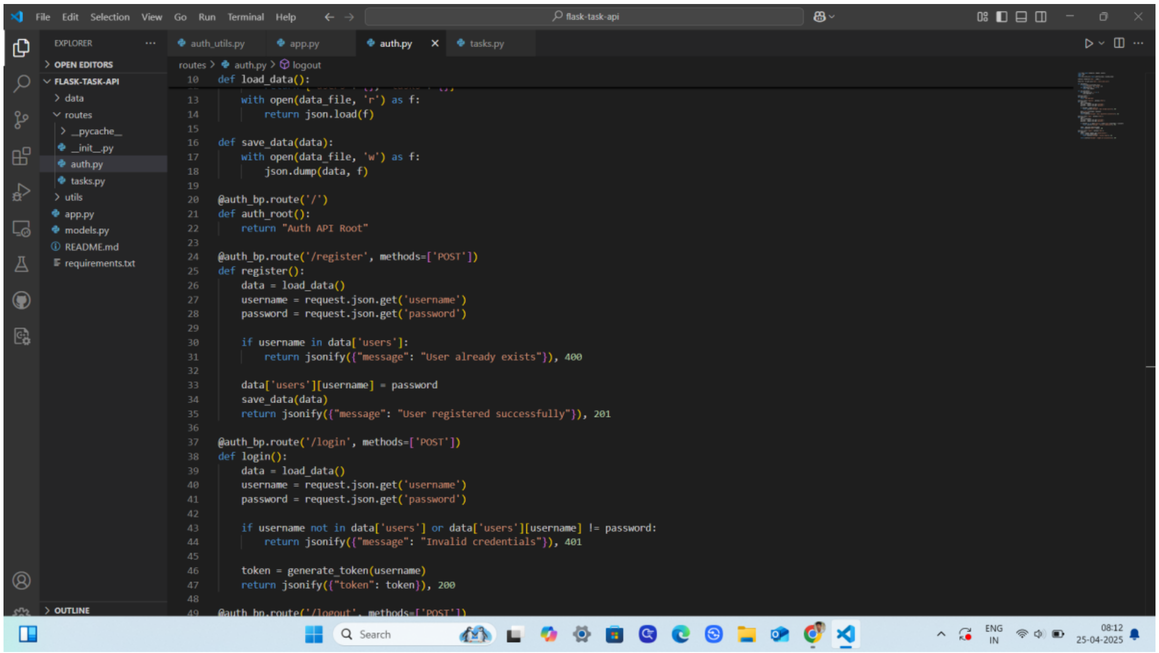
Reviewed the task requirements from the HIRING DOC and set up the development environment. Verified access to necessary repositories and installed dependencies.



```
{
  "users": [],
  "tasks": {}
}
```

8:12 AM

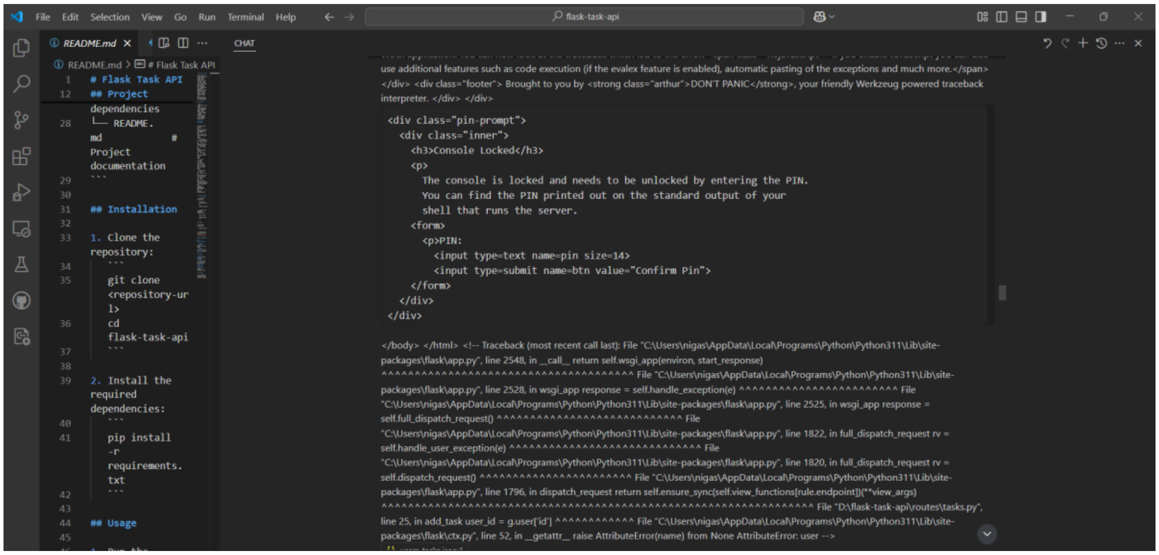
Coded the `auth.py` file, implementing user registration and login endpoints for authentication.



```python
def load_data():
    with open(data_file, 'r') as f:
        return json.load(f)

def save_data(data):
    with open(data_file, 'w') as f:
        json.dump(data, f)

@auth_bp.route('/')
def auth_root():
    return "Auth API Root"

@auth_bp.route('/register', methods=['POST'])
def register():
    data = load_data()
    username = request.json.get('username')
    password = request.json.get('password')

    if username in data['users']:
        return jsonify({"message": "User already exists"}), 400

    data['users'][username] = password
    save_data(data)
    return jsonify({"message": "User registered successfully"}), 201

@auth_bp.route('/login', methods=['POST'])
def login():
    data = load_data()
    username = request.json.get('username')
    password = request.json.get('password')

    if username not in data['users'] or data['users'][username] != password:
        return jsonify({"message": "Invalid credentials"}), 401

    token = generate_token(username)
    return jsonify({"token": token}), 200

@auth_bp.route('/logout', methods=['POST'])
```

8:22 AM

Encountered an error: `g.user` not set in `/tasks/tasks` endpoint. Added a `token_required` decorator to extract the user from the token and set `g.user`.


File Edit Selection View Go Run Terminal Help
flask-task-api
README.md
CHAT
README.md > # Flask Task API

```markdown
# Flask Task API
## Project
dependencies
└── README.md                # Project documentation
```

Installation

1. Clone the repository:
   ```
   git clone <repository-url>
   cd flask-task-api
   ```

2. Install the required dependencies:
   ```
   pip install -r requirements.txt
   ```

Usage
```

```
use additional features such as code execution (if the evalex feature is enabled), automatic pasting of the exceptions and much more.</span> </div> <div class="footer"> Brought to you by <strong class="arthur">DON'T PANIC</strong>, your friendly Werkzeug powered traceback interpreter. </div> </div>

```html
<div class="pin-prompt">
  <div class="inner">
    <h3>Console Locked</h3>
    <p>
      The console is locked and needs to be unlocked by entering the PIN.
      You can find the PIN printed out on the standard output of your
      shell that runs the server.
    <form>
      <p>PIN:
        <input type=text name=pin size=14>
        <input type=submit name=btn value="Confirm Pin">
    </form>
  </div>
</div>
```

```
</body> </html> <!-- Traceback (most recent call last): File "C:\Users\nigas\AppData\Local\Programs\Python\Python311\Lib\site-packages\flask\app.py", line 2548, in __call__ return self.wsgi_app(environ, start_response) ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^ File "C:\Users\nigas\AppData\Local\Programs\Python\Python311\Lib\site-packages\flask\app.py", line 2528, in wsgi_app response = self.handle_exception(e) ^^^^^^^^^^^^^^^^^^^^^^^^^ File "C:\Users\nigas\AppData\Local\Programs\Python\Python311\Lib\site-packages\flask\app.py", line 2525, in wsgi_app response = self.full_dispatch_request() ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^ File "C:\Users\nigas\AppData\Local\Programs\Python\Python311\Lib\site-packages\flask\app.py", line 1822, in full_dispatch_request rv = self.handle_user_exception(e) ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^ File "C:\Users\nigas\AppData\Local\Programs\Python\Python311\Lib\site-packages\flask\app.py", line 1820, in full_dispatch_request rv = self.dispatch_request() ^^^^^^^^^^^^^^^^^^^^^^^ File "C:\Users\nigas\AppData\Local\Programs\Python\Python311\Lib\site-packages\flask\app.py", line 1796, in dispatch_request return self.ensure_sync(self.view_functions[rule.endpoint])(**view_args) ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^ File "D:\flask-task-api\routes\tasks.py", line 25, in add_task user_id = g.user['id'] ^^^^^^^^^^^^ File "C:\Users\nigas\AppData\Local\Programs\Python\Python311\Lib\site-packages\flask\ctx.py", line 52, in __getattr__ raise AttributeError(name) from None AttributeError: user -->
```

8:42 AM

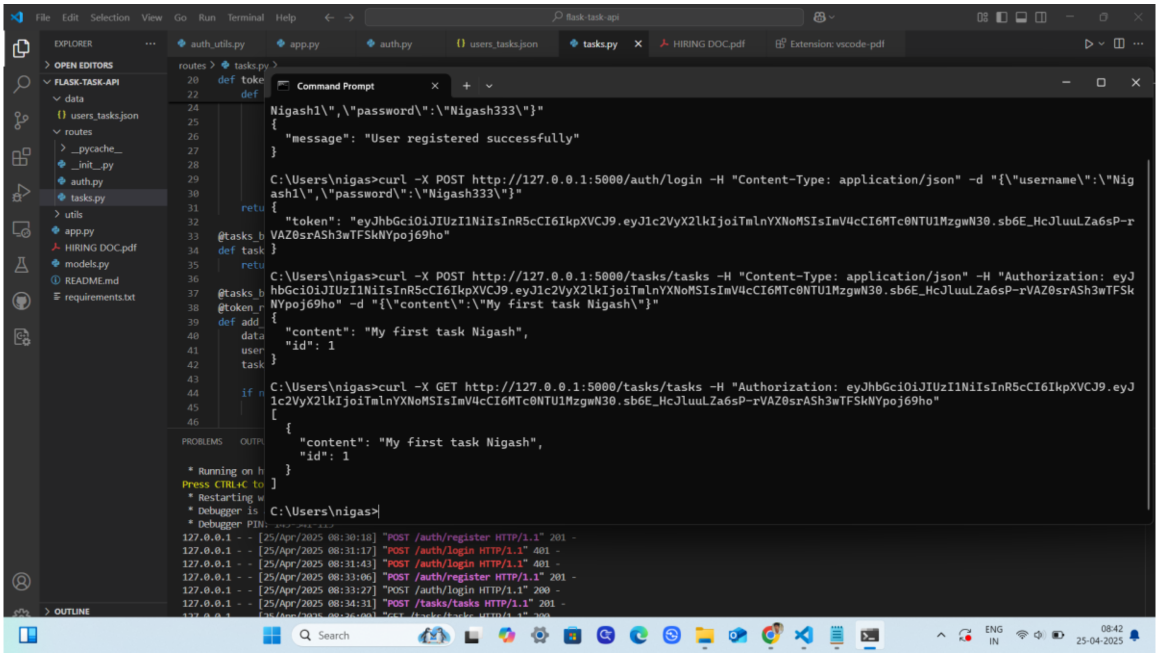
Registered a user and used curl in the command line to log in and add the first task via the API.

Login Command:

```
curl -X POST http://127.0.0.1:5000/login -H "Content-Type: application/json" -d "{\"username\": \"Nigash\", \"password\": \"Nigash333\"}"
```

Add Task Command:

```
curl -X POST http://127.0.0.1:5000/task -H "Authorization: your_token_here" -H "Content-Type: application/json" -d "{\"task\": \"First Task from curl\"}"
```

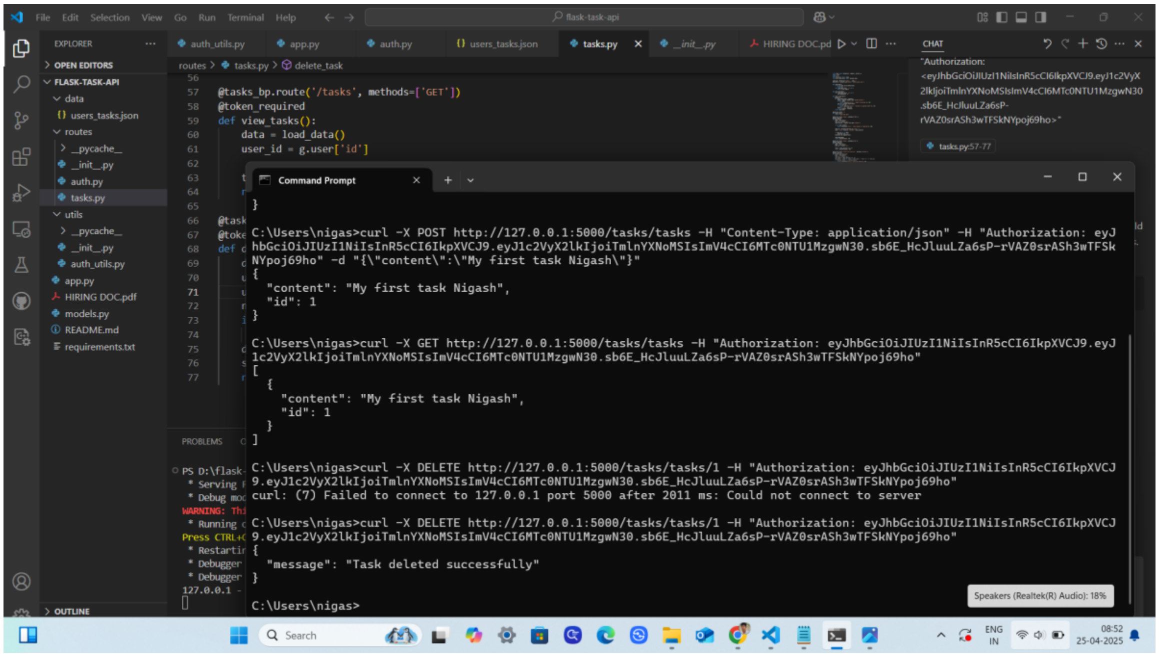


```
Nigash1\",\"password\":\"Nigash333\"}"
{
  "message": "User registered successfully"
}

C:\Users\nigas>curl -X POST http://127.0.0.1:5000/auth/login -H "Content-Type: application/json" -d "{\"username\":\"Nig
ash1\",\"password\":\"Nigash333\"}"
{
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIjoiTmlnYXNoMSIsImV4cCI6MTc0NTU1MzgwN30.sb6E_HcJluuLZa6sP-r
VAZ0srASh3wTFSkNYpoj69ho"
}

C:\Users\nigas>curl -X POST http://127.0.0.1:5000/tasks/tasks -H "Content-Type: application/json" -H "Authorization: eyJ
hbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIjoiTmlnYXNoMSIsImV4cCI6MTc0NTU1MzgwN30.sb6E_HcJluuLZa6sP-rVAZ0srASh3wTFSk
NYpoj69ho" -d "{\"content\":\"My first task Nigash\"}"
{
  "content": "My first task Nigash",
  "id": 1
}

C:\Users\nigas>curl -X GET http://127.0.0.1:5000/tasks/tasks -H "Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ
1c2VyX2lkIjoiTmlnYXNoMSIsImV4cCI6MTc0NTU1MzgwN30.sb6E_HcJluuLZa6sP-rVAZ0srASh3wTFSkNYpoj69ho"
[
  {
    "content": "My first task Nigash",
    "id": 1
  }
]

C:\Users\nigas>
```

```
127.0.0.1 - - [25/Apr/2025 08:30:18] "POST /auth/register HTTP/1.1" 201 -
127.0.0.1 - - [25/Apr/2025 08:31:17] "POST /auth/login HTTP/1.1" 401 -
127.0.0.1 - - [25/Apr/2025 08:31:43] "POST /auth/login HTTP/1.1" 401 -
127.0.0.1 - - [25/Apr/2025 08:33:06] "POST /auth/register HTTP/1.1" 201 -
127.0.0.1 - - [25/Apr/2025 08:33:27] "POST /auth/login HTTP/1.1" 200 -
127.0.0.1 - - [25/Apr/2025 08:34:31] "POST /tasks/tasks HTTP/1.1" 201 -
```

8:52 AM

Attempted to delete a previously added task using the DELETE endpoint with `curl`.



```python
@tasks_bp.route('/tasks', methods=['GET'])
@token_required
def view_tasks():
    data = load_data()
    user_id = g.user['id']
```

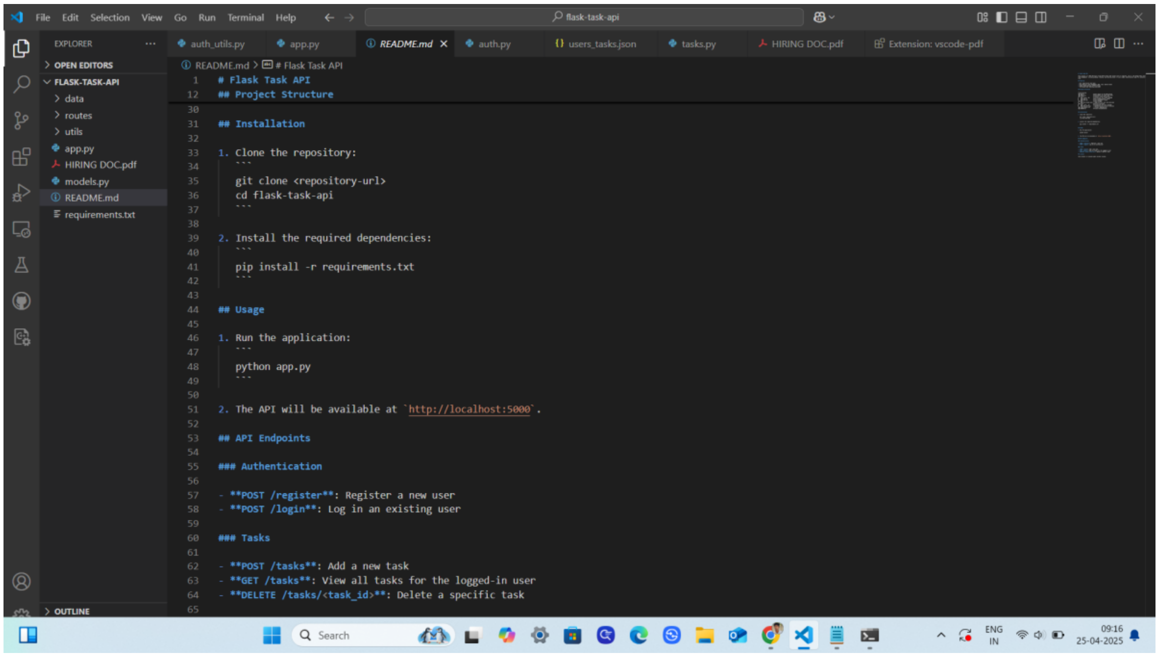
"Authorization:
<eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIjoiTmlnYXNoMSIsImV4cCI6MTc0NTU1MzgwN30.sb6E_HcJluuLZa6sP-rVAZ0srASh3wTFSkNYpoj69ho>"

tasks.py:57-77

```
}

C:\Users\nigas>curl -X POST http://127.0.0.1:5000/tasks/tasks -H "Content-Type: application/json" -H "Authorization: eyJ
hbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIjoiTmlnYXNoMSIsImV4cCI6MTc0NTU1MzgwN30.sb6E_HcJluuLZa6sP-rVAZ0srASh3wTFSk
NYpoj69ho" -d "{\"content\":\"My first task Nigash\"}"
{
  "content": "My first task Nigash",
  "id": 1
}

C:\Users\nigas>curl -X GET http://127.0.0.1:5000/tasks/tasks -H "Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ
1c2VyX2lkIjoiTmlnYXNoMSIsImV4cCI6MTc0NTU1MzgwN30.sb6E_HcJluuLZa6sP-rVAZ0srASh3wTFSkNYpoj69ho"
[
  {
    "content": "My first task Nigash",
    "id": 1
  }
]

C:\Users\nigas>curl -X DELETE http://127.0.0.1:5000/tasks/tasks/1 -H "Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ
9.eyJ1c2VyX2lkIjoiTmlnYXNoMSIsImV4cCI6MTc0NTU1MzgwN30.sb6E_HcJluuLZa6sP-rVAZ0srASh3wTFSkNYpoj69ho"
curl: (7) Failed to connect to 127.0.0.1 port 5000 after 2011 ms: Could not connect to server

C:\Users\nigas>curl -X DELETE http://127.0.0.1:5000/tasks/tasks/1 -H "Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ
9.eyJ1c2VyX2lkIjoiTmlnYXNoMSIsImV4cCI6MTc0NTU1MzgwN30.sb6E_HcJluuLZa6sP-rVAZ0srASh3wTFSkNYpoj69ho"
{
  "message": "Task deleted successfully"
}

C:\Users\nigas>
```

9:16 AM

Wrote a comprehensive `README.md` documenting project setup, usage, and API endpoints.



```markdown
# Flask Task API
## Project Structure

## Installation

1. Clone the repository:
   ```

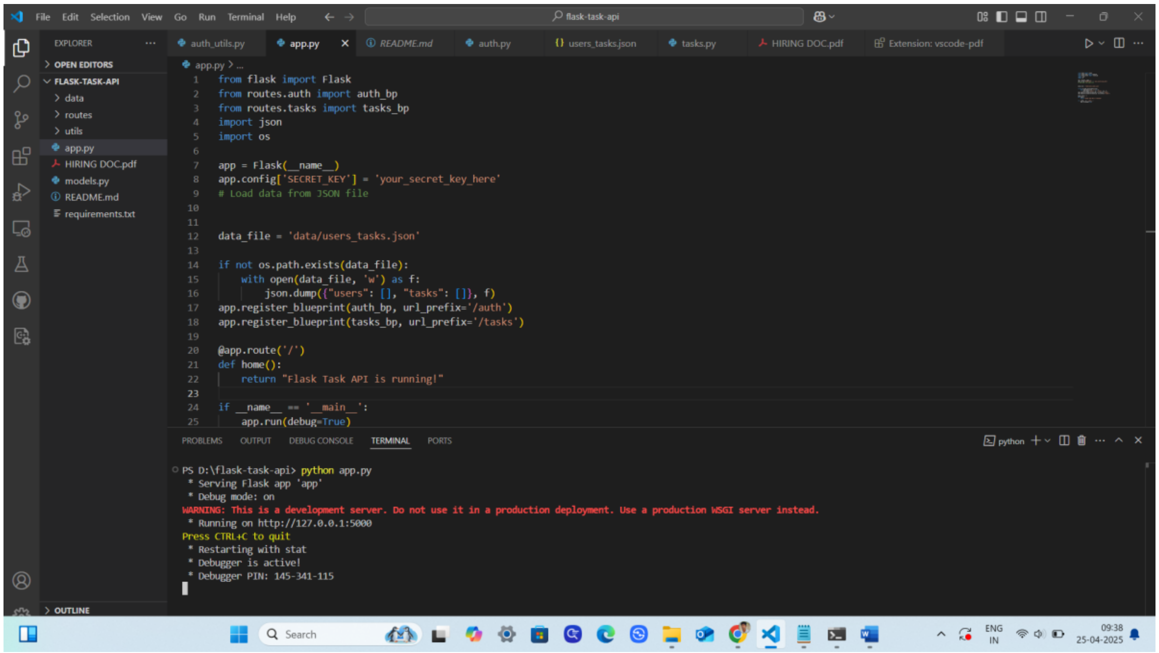
   git clone <repository-url>
   cd flask-task-api
   ```


2. Install the required dependencies:
   ```

   pip install -r requirements.txt
   ```


## Usage

1. Run the application:
   ```

   python app.py
   ```


2. The API will be available at `http://localhost:5000`.

## API Endpoints

### Authentication

- **POST /register**: Register a new user
- **POST /login**: Log in an existing user

### Tasks

- **POST /tasks**: Add a new task
- **GET /tasks**: View all tasks for the logged-in user
- **DELETE /tasks/<task_id>**: Delete a specific task

```

9:38 AM

Completed the backend project and verified all core functionalities were working as expected.



```python
from flask import Flask
from routes.auth import auth_bp
from routes.tasks import tasks_bp
import json
import os

app = Flask(__name__)
app.config['SECRET_KEY'] = 'your_secret_key_here'
# Load data from JSON file


data_file = 'data/users_tasks.json'

if not os.path.exists(data_file):
    with open(data_file, 'w') as f:
        json.dump({"users": [], "tasks": []}, f)
app.register_blueprint(auth_bp, url_prefix='/auth')
app.register_blueprint(tasks_bp, url_prefix='/tasks')

@app.route('/')
def home():
    return "Flask Task API is running!"

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

```
PS D:\flask-task-api> python app.py
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 145-341-115
```