NAAN MUDHALVAN

PROJECT PHASE - V

COLLEGE CODE : **3108**

COLLEGE NAME : Jeppiaar Engineering College

DEPARTMENT : **Artificial Intelligence and Data Science**

STUDENT NM-ID : **aut23aids029**

REGISTER NO : **310823243029**

Completed the project named as

TECHNOLOGY - Energy Consumption Tracker App using JavaScript, HTML, CSS, and IoT Integration

SUBMITTED BY,

NAME : **MATHAVAN V**

MOBILE NO : **9363561502**

Phase 5: Project Demonstration & Documentation

Title: Energy Consumption Tracker App

Abstract:

The Energy Consumption Tracker App aims to provide users with insights into their energy usage by leveraging JavaScript, HTML, CSS, and IoT (Internet of Things) technologies. In its final phase, the system integrates real-time data collection from IoT devices, user-friendly interfaces for data visualization, and secure data management. This document provides a comprehensive report of the project's completion, covering the system demonstration, technical documentation, performance metrics, source code, and testing reports. The project is designed to help users monitor and optimize their energy consumption effectively. Screenshots, diagrams, and codebase snapshots will be included for a full understanding of the system's architecture and functionality.

Index

1. Project Demonstration ..... Page 2
2. Project Documentation ..... Page 3
3. Feedback and Final Adjustments ..... Page 4
4. Final Project Report Submission ..... Page 5
5. Project Handover and Future Works ..... Page 6

1. Project Demonstration

Overview: The Energy Consumption Tracker App will be demonstrated to stakeholders, showcasing its features, performance improvements, and functionality. This demonstration highlights the system's real-time data collection, user interface, and data visualization capabilities.

Demonstration Details:

- **System Walkthrough:** A live walkthrough of the application, from user interaction to the display of energy consumption metrics.

- **Real-Time Data Integration:** The demonstration will show how the app collects and displays real-time energy data from IoT devices.
- **User Interface:** Showcase the user-friendly interface that allows users to visualize their energy consumption patterns.
- **Performance Metrics:** Highlight response times, data accuracy, and system scalability under multiple users.
- **Security Measures:** Explain the security protocols in place to protect user data and ensure privacy.

Outcome: By the end of the demonstration, the system's ability to provide real-time insights into energy consumption and its user-friendly interface will be showcased to stakeholders.

2. Project Documentation

Overview: Comprehensive documentation for the Energy Consumption Tracker App is provided to detail every aspect of the project. This includes system architecture, code explanations, and usage guidelines for both users and administrators.

Documentation Sections:

- **System Architecture:** Diagrams illustrating the complete system, including IoT device integration, data flow, and user interface components.
- **Code Documentation:** Source code and explanations for all code modules, including JavaScript for data handling, HTML for structure, and CSS for styling.
- **User Guide:** A manual for end users explaining how to interact with the app and interpret energy consumption data.
- **Administrator Guide:** Instructions for system maintenance, monitoring, and performance testing procedures.
- **Testing Reports:** Detailed reports on performance metrics, load testing, and data security evaluations.

Outcome: All critical components of the system will be well-documented, providing a clear guide for future development, deployment, or system scaling.

3. Feedback and Final Adjustments

Overview: Feedback from the project demonstration will be collected from instructors, stakeholders, and a broader group of test users. This feedback will be used to make final refinements before project handover.

Steps:

- **Feedback Collection:** Gather feedback from mentors, stakeholders, and test users via surveys and observation during the demonstration.
- **Refinement:** Address any performance bottlenecks, inaccuracies in data display, or usability issues based on the feedback.
- **Final Testing:** After adjustments, the system will undergo final testing to ensure full functionality, usability, and scalability.

Outcome: Final adjustments will optimize the system for a broader rollout, ensuring that it is fully ready for real-world deployment.

4. Final Project Report Submission

Overview: The final project report provides a comprehensive summary of all phases, key achievements, challenges faced, and outcomes of the Energy Consumption Tracker App project. This report will include testing results, performance improvements, and future recommendations.

Report Sections:

- **Executive Summary:** A concise overview of the project, outlining its objectives and major achievements.
- **Phase Breakdown:** A detailed breakdown of each phase, covering IoT integration, user interface development, and data management.
- **Challenges & Solutions:** Document key challenges encountered, such as data accuracy or user interface issues, and how they were resolved.

- **Outcomes:** A summary of the system's current capabilities and readiness for deployment.

Outcome: A detailed project report will be submitted, outlining the entire journey from concept to completion.

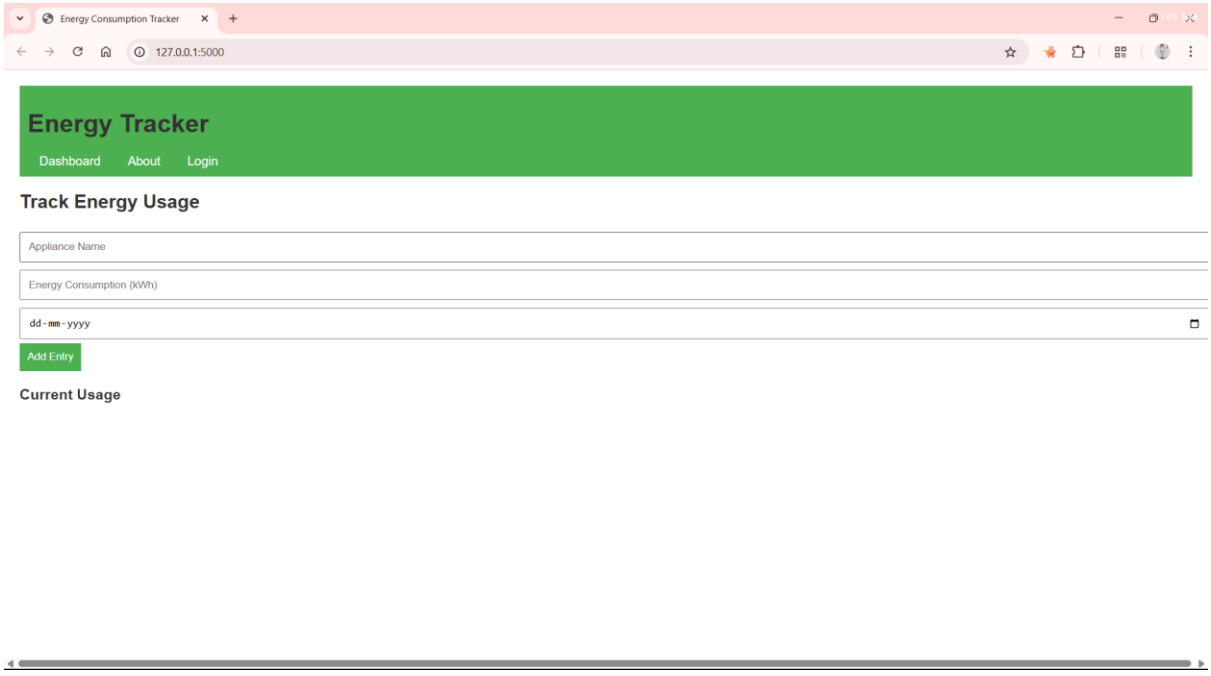
5. Project Handover and Future Works

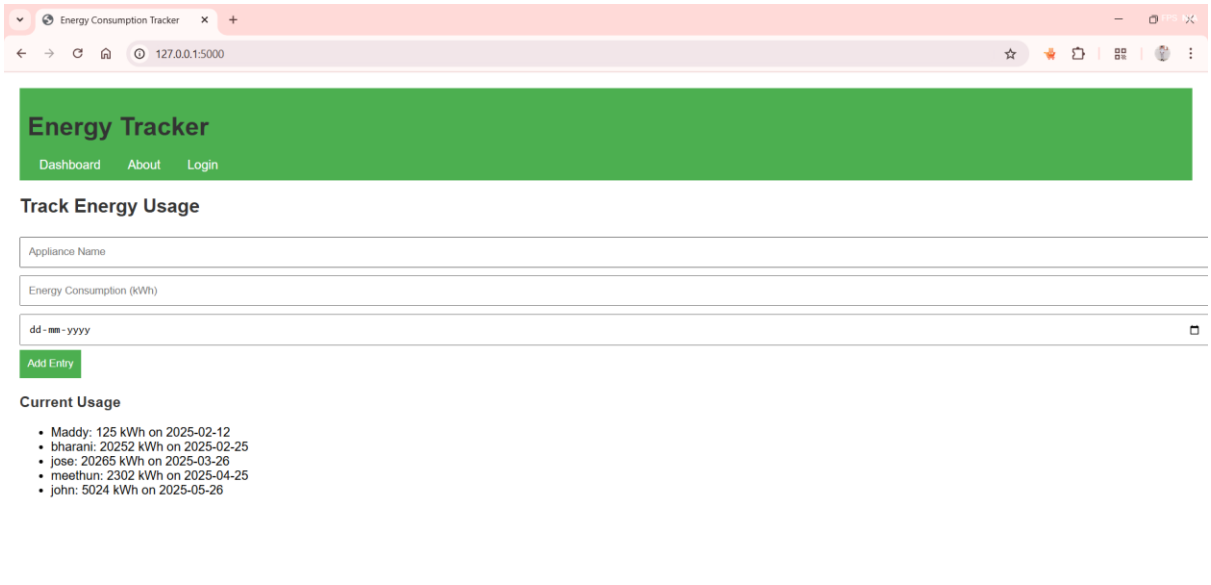
Overview: The project's introduction for future development.

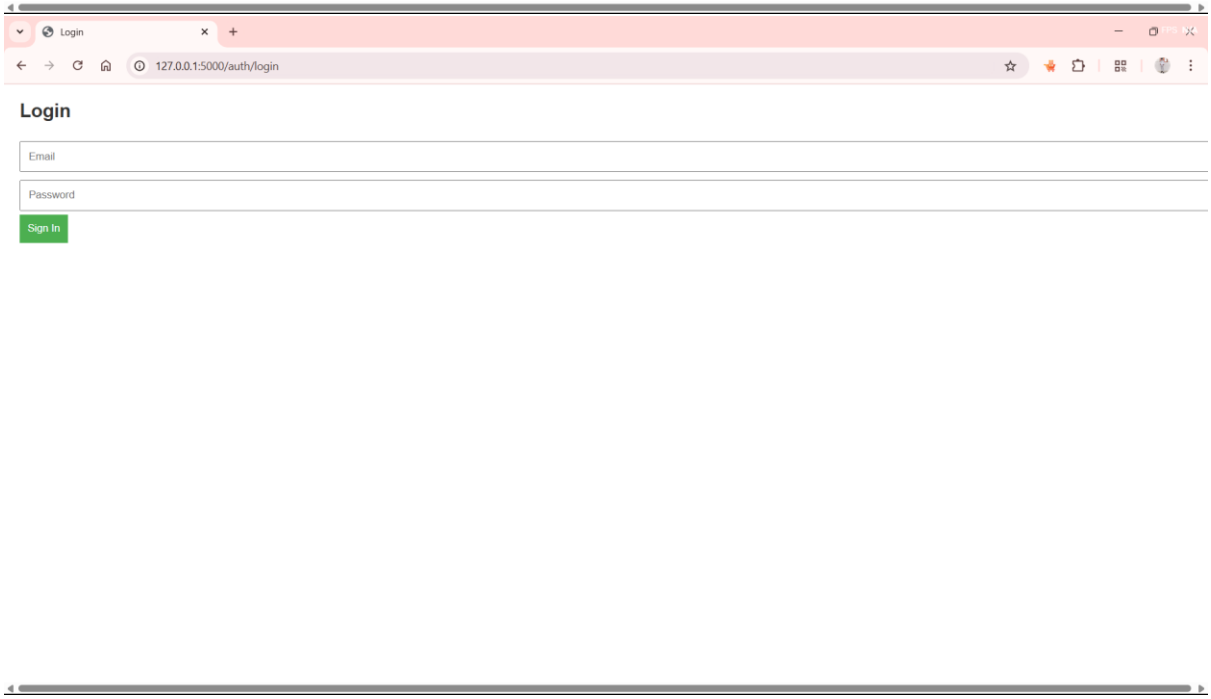
Handover Details:

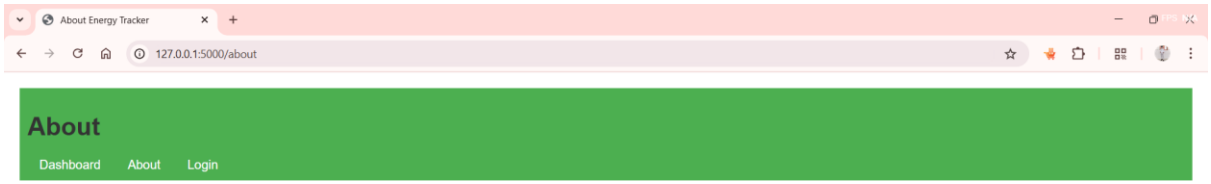
- **Next Steps:** Suggestions for future work, including scaling the system to support more users, enhancing data visualization features, and implementing additional IoT device compatibility.

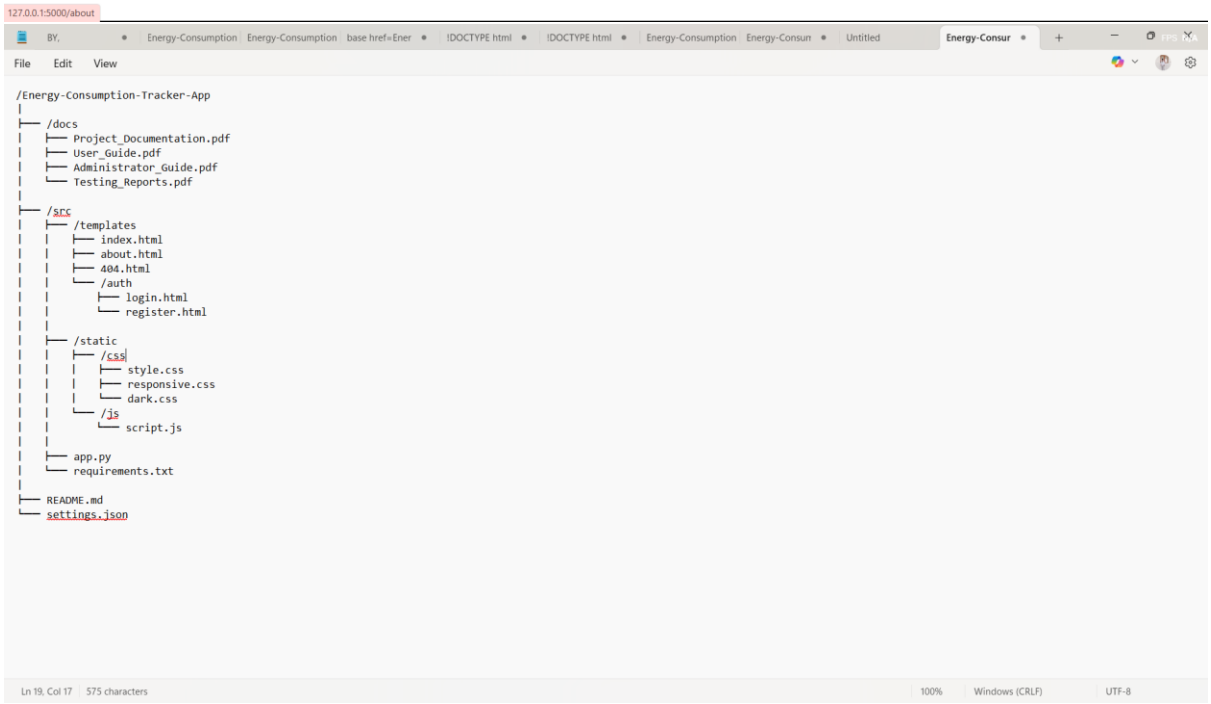
Outcome: The Energy Consumption Tracker App will be officially handed over, along with recommendations for future enhancements and guidelines for system maintenance.

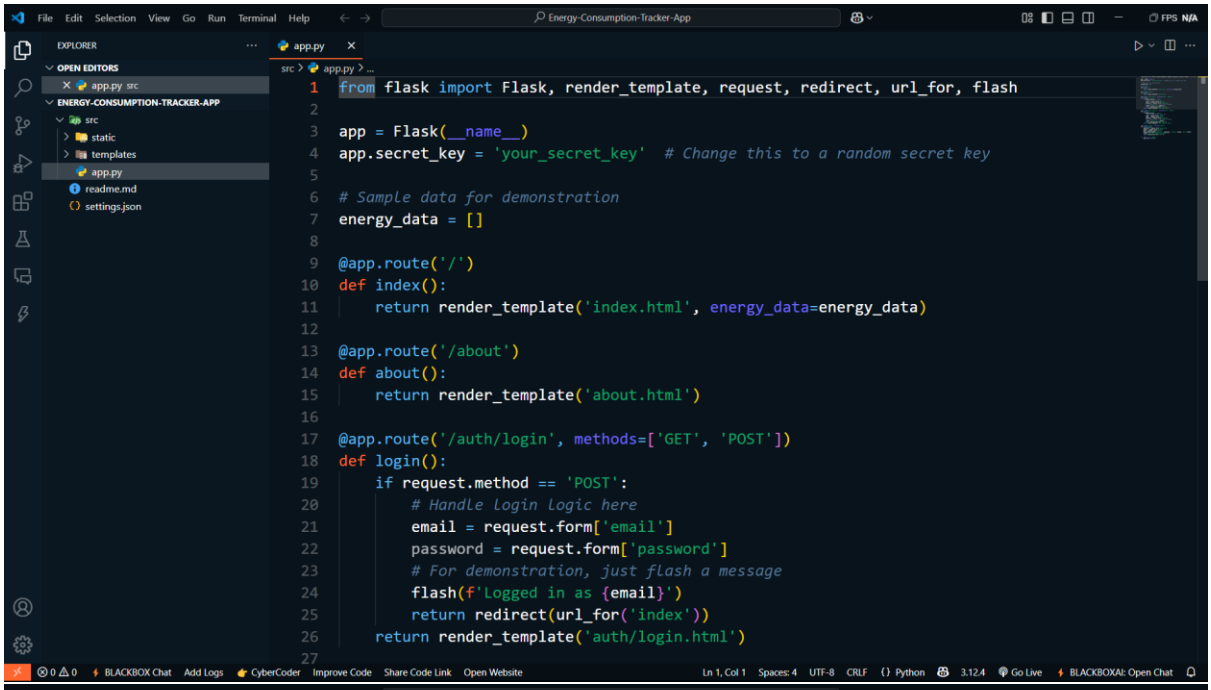


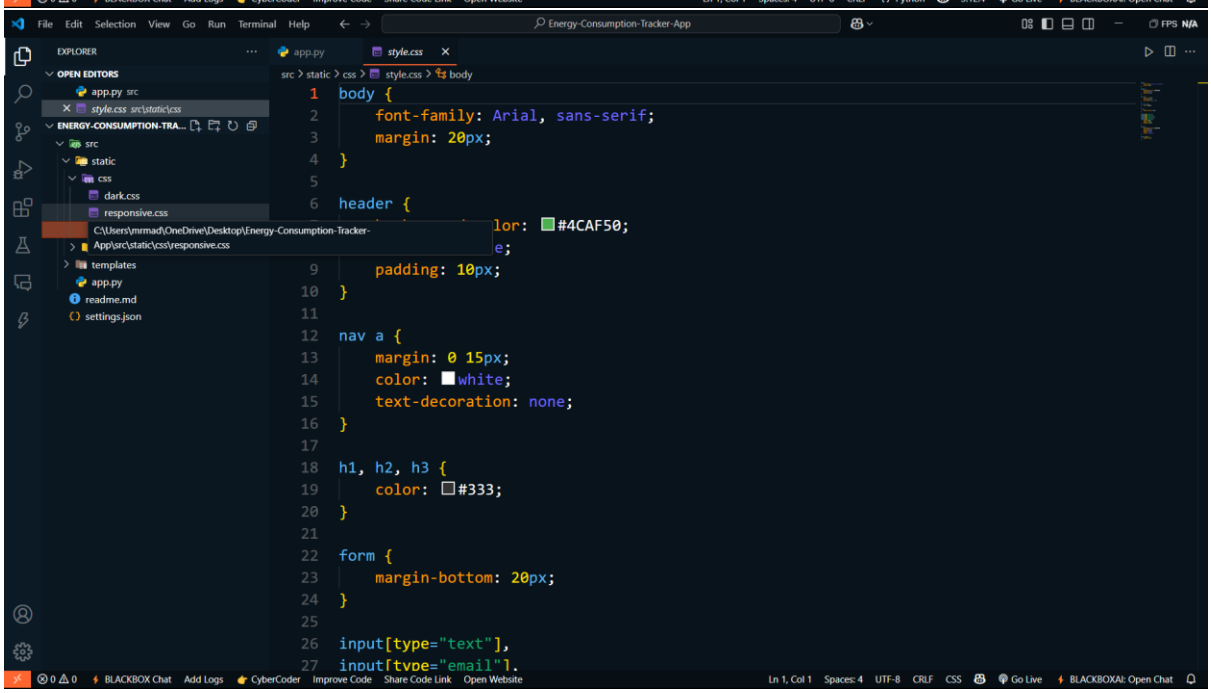

Energy Consumption Tracker
127.0.0.1:5000
Energy Tracker
Dashboard
About
Login
Track Energy Usage
Appliance Name
Energy Consumption (kWh)
dd-mm-yyyy
Add Entry
Current Usage
Maddy: 125 kWh on 2025-02-12
bharani: 20252 kWh on 2025-02-25
jose: 20265 kWh on 2025-03-26
meethun: 2302 kWh on 2025-04-25
john: 5024 kWh on 2025-05-26

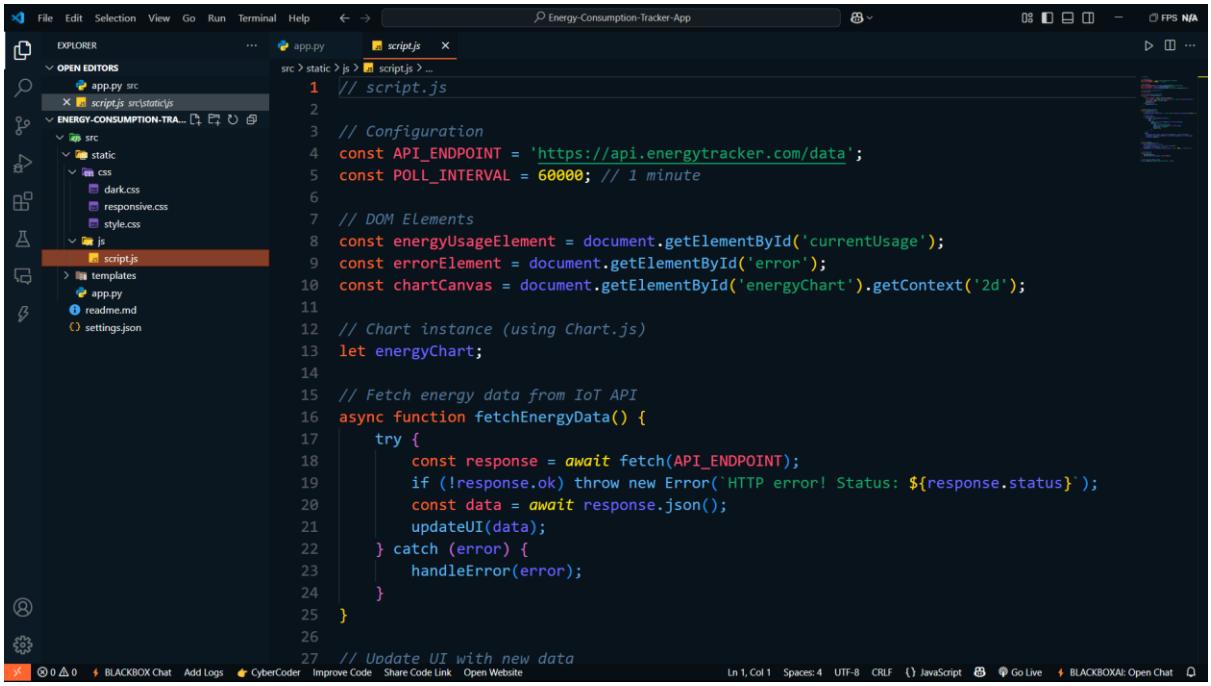

Login
127.0.0.1:5000/auth/login
Login
Email
Password
Sign In


About Energy Tracker
127.0.0.1:5000/about
About
Dashboard
About
Login
About This Project
Track and optimize your energy consumption with our intuitive monitoring system.


127.0.0.1:5000/about
File
Edit
View
/Energy-Consumption-Tracker-App
├── /docs
│ ├── Project_Documentation.pdf
│ ├── User_Guide.pdf
│ ├── Administrator_Guide.pdf
│ └── Testing_Reports.pdf
├── /src
│ ├── /templates
│ │ ├── index.html
│ │ ├── about.html
│ │ ├── 404.html
│ │ └── /auth
│ │ ├── login.html
│ │ └── register.html
│ ├── /static
│ │ ├── /css
│ │ │ ├── style.css
│ │ │ ├── responsive.css
│ │ │ └── dark.css
│ │ └── /js
│ │ └── script.js
│ ├── app.py
│ └── requirements.txt
├── README.md
└── settings.json
Ln 19, Col 17
575 characters
100%
Windows (CRLF)
UTF-8


File Edit Selection View Go Run Terminal Help
Energy-Consumption-Tracker-App
EXPLORER
OPEN EDITORS
app.py src
ENERGY-CONSUMPTION-TRACKER-APP
src
static
templates
app.py
readme.md
settings.json
app.py
src > app.py > ...
from flask import Flask, render_template, request, redirect, url_for, flash
app = Flask(__name__)
app.secret_key = 'your_secret_key' # Change this to a random secret key
# Sample data for demonstration
energy_data = []
@app.route('/')
def index():
return render_template('index.html', energy_data=energy_data)
@app.route('/about')
def about():
return render_template('about.html')
@app.route('/auth/login', methods=['GET', 'POST'])
def login():
if request.method == 'POST':
# Handle login logic here
email = request.form['email']
password = request.form['password']
# For demonstration, just flash a message
flash(f'Logged in as {email}')
return redirect(url_for('index'))
return render_template('auth/login.html')
BLACKBOX Chat
Add Logs
CyberCoder
Improve Code
Share Code Link
Open Website
Ln 1, Col 1
Spaces: 4
UTF-8
CRLF
Python
3.12.4
Go Live
BLACKBOXAI: Open Chat


File Edit Selection View Go Run Terminal Help
Energy-Consumption-Tracker-App
EXPLORER
OPEN EDITORS
app.py src
style.css src\static\css
ENERGY-CONSUMPTION-TRA...
src
static
css
dark.css
responsive.css
C:\Users\mrmad\OneDrive\Desktop\Energy-Consumption-Tracker-App\src\static\css\responsive.css
templates
app.py
readme.md
settings.json
app.py
style.css
src > static > css > style.css > body
body {
font-family: Arial, sans-serif;
margin: 20px;
}
header {
lor: #4CAF50;
e;
padding: 10px;
}
nav a {
margin: 0 15px;
color: white;
text-decoration: none;
}
h1, h2, h3 {
color: #333;
}
form {
margin-bottom: 20px;
}
input[type="text"],
input[type="email"],
BLACKBOX Chat
Add Logs
CyberCoder
Improve Code
Share Code Link
Open Website
Ln 1, Col 1
Spaces: 4
UTF-8
CRLF
CSS
Go Live
BLACKBOXAI: Open Chat


File Edit Selection View Go Run Terminal Help
Energy-Consumption-Tracker-App
FPS N/A
EXPLORER
OPEN EDITORS
app.py src
script.js src\static\js
ENERGY-CONSUMPTION-TRA...
src
static
css
dark.css
responsive.css
style.css
js
script.js
templates
app.py
readme.md
settings.json
app.py
script.js
src > static > js > script.js > ...
1 // script.js
2
3 // Configuration
4 const API_ENDPOINT = 'https://api.energytracker.com/data';
5 const POLL_INTERVAL = 60000; // 1 minute
6
7 // DOM Elements
8 const energyUsageElement = document.getElementById('currentUsage');
9 const errorElement = document.getElementById('error');
10 const chartCanvas = document.getElementById('energyChart').getContext('2d');
11
12 // Chart instance (using Chart.js)
13 let energyChart;
14
15 // Fetch energy data from IoT API
16 async function fetchEnergyData() {
17 try {
18 const response = await fetch(API_ENDPOINT);
19 if (!response.ok) throw new Error(`HTTP error! Status: ${response.status}`);
20 const data = await response.json();
21 updateUI(data);
22 } catch (error) {
23 handleError(error);
24 }
25 }
26
27 // Update UI with new data
0 0 BLACKBOX Chat Add Logs CyberCoder Improve Code Share Code Link Open Website
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF {} JavaScript Go Live BLACKBOXAI: Open Chat