**Unity & API Integration using Flask**

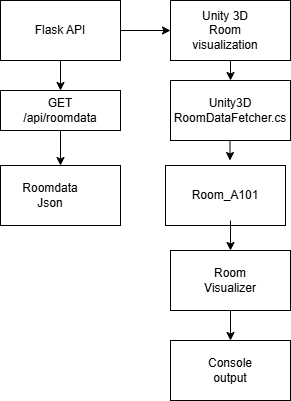
**Objective**

The objective of this project is to build a real-time digital twin simulation using Unity 3D for 3D room visualization and Flask (Python) to serve simulated sensor data through a REST API. The goal is to enable interactive feedback and status monitoring of campus rooms based on real or simulated input such as temperature, electricity usage, occupancy, and HVAC system state.

**Tools & Technologies**

|  |  |
| --- | --- |
| **Tool** | **Purpose** |
| Unity 3D | 3D visualization, scripting (C#) |
| Python 3 | Flask backend simulation |
| Flask | REST API to serve sensor data |
| GitHub | Version control and collaboration |

**Architecture Overview:**

****

**Summary:**

I implemented a complete data flow between a Python-based Flask API and Unity3D to simulate real-time room monitoring. I began by creating a Flask backend that generates mock sensor data for two rooms, including temperature and electricity status. This data is automatically updated every 10 seconds using a background thread. On the Unity side, I created simple cube GameObjects representing rooms and attached a custom RoomVisualizer script to each. I also developed a RoomDataFetcher script using UnityWebRequest to send HTTP GET requests to the API every 5 seconds. Once the JSON data is received and parsed, it is printed in Unity’s Console, showing the current temperature and electricity state for each room. This setup successfully demonstrates backend-to-frontend integration and provides a foundation for adding real-time visual feedback and interactivity in upcoming sprints.

**Implementation Summary**

Flask Backend (Python)

* Created a REST API using Flask (app.py)
* Simulates data: temperature, occupancy, electricity, hvacStatus
* Runs at: <http://127.0.0.1:5000/api/roomdata>

Unity Frontend

* GameObjects (Cubes) named after rooms (e.g., Room\_A101)
* RoomDataFetcher.cs fetches JSON data every 5 seconds
* RoomVisualizer.cs logs data to Console or updates room colors
* Materials (green/yellow/red) represent sensor thresholds

**Outcomes**

* Created reusable Flask backend for simulated campus data
* Integrated REST API with Unity using UnityWebRequest
* Developed Unity script to parse and process JSON sensor data
* Displayed dynamic room info through Console

**Sample outputs:**

Terminal output

A screenshot of a computer

AI-generated content may be incorrect.

Unity 3D Console output

**A screenshot of a computer

AI-generated content may be incorrect.**