**HEALTH MONITORING API**

**Flask JWT Device API**

This is a simple Flask-based API for managing devices with user authentication using JWT tokens.

**What it does**

* Lets users register and log in securely (passwords are hashed).
* Allows logged-in users to add, update, and delete devices.
* Anyone can view device information.
* Uses JWT tokens to protect certain routes.
* Stores data in a SQLite database.

**Features**

* User registration and login with password hashing.
* JWT authentication to secure routes.
* CRUD operations for devices (Create, Read, Update, Delete).
* Filter and fetch devices by ID or location.
* Easy-to-use JSON API that works smoothly with tools like Postman.

**Project Structure**

pro\_w1/

│

├── app/

│ ├── \_\_init\_\_.py # Flask app, DB, JWT setup

│ ├── models.py # SQLAlchemy models for User and Device

│ └── routes.py # API route handlers

│

├── instance/

│ └── devices.db # SQLite database file (auto-generated)

│

├── run.py # Entry point to run the Flask app

├── README.md # Project overview and instructions

└── requirements.txt # Python packages needed

**How to set it up**

1. **Clone the repo**
2. git clone <your-repo-url>
3. cd pro\_w1
4. **Create and activate a virtual environment (optional but recommended)**
5. python -m venv venv
6. source venv/bin/activate # For Linux/Mac
7. venv\Scripts\activate # For Windows
8. **Install dependencies**
9. pip install -r requirements.txt
10. **Run the app**
11. python run.py

The API should now be live at http://127.0.0.1:5000/

**Example Endpoints**

* POST /register - Register a new user
* POST /login - Log in and get JWT token
* POST /devices - Add a device (JWT required)
* GET /devices?id=1 - Get device by ID
* GET /devices?location=Office - Get device by location
* PUT /devices/<device\_id> - Update a device (JWT required)
* DELETE /devices/<device\_id> - Delete a device (JWT required)
* GET /devices/all - Get all devices