

Incident Log API - SPARKLEHOOD Assignment

GitHub-<https://github.com/Surajgupta63/HumanChain-AI-Safety-Incident-Log-API>

This project is a RESTful API service built using Python, Flask, and SQLAlchemy(SQLlite). It is developed to log and manage AI Safety Incidents with basic CRUD functionality. Includes user authentication, incident reporting, updating, deleting, and dashboard viewing.

1. Tech Stack:

- **Language:** Python
- **Framework:** Flask
- **Database:** SQLite (with SQLAlchemy ORM)
- **Frontend:** HTML (Jinja2 templating)

2. Features:

- User Signup, Login, Logout
- Password hashing for secure authentication
- Create, Update, Delete, and Search Incidents
- View all Incidents in a Dashboard
- Incident severity levels: LOW, MEDIUM, HIGH

3. Prerequisites:

Step	What You Need	Command
1	Python installed (version 3.7+ recommended)	Check: python --version
2	Create a virtual environment	python -m venv venv
3	Activate virtual environment (depends on OS)	Windows: venv\Scripts\activate Mac/Linux: source venv/bin/activate
4	Install Flask	pip install Flask
5	Install other dependencies (like Bootstrap, or SQLAlchemy, etc.)	pip install -r requirements.txt (if you have this file)
6	Make sure your main file (app.py or run.py) is present	Run: python app.py

4. Project Structure:

```
humanchain-incident-log/
```

```
    ├── app.py
    ├── db.py
    ├── models/
    |   └── incidents.py
    ├── templates/
    |   ├── login.html
    |   ├── signup.html
    |   ├── dashboard.html
    |   ├── incident.html
    |   └── update.html
    └── 404.html
    └── requirements.txt
    └── README.md
```

5. Error Handling

The app includes error handling for common scenarios. Below are the error types you may encounter and their solutions:

1. 500 Internal Server Error

- This error typically occurs when something goes wrong on the server side.
- **Solution:** Check your app.py for any issues with the logic, such as missing imports or incorrect routes.

2. 404 Not Found

- This error occurs if the URL is incorrect or the route doesn't exist.
- **Solution:** Double-check that you are visiting the correct URL, and make sure all routes are defined in app.py.

3. 400 Bad Request

- If the form is submitted without necessary fields (e.g., missing incident title or description), you may encounter this error.
- **Solution:** Ensure all required fields are filled before submitting the form.

Custom Error Pages

You can add custom error pages by defining error handlers in your Flask app. For example, to handle 404 errors:

```
@app.errorhandler(404)  
def page_not_found(error):  
    return render_template('404.html'), 404
```

6. Getting Started

1. Clone the Repository

```
1: git clone https://github.com/your-username/humanchain-incident-log.git  
cd humanchain-incident-log
```

OR:

- Unzip the provided code folder.
- Open VS Code and select File → Open Folder.
- Choose the extracted project folder.

2. (Optional) Create a Virtual Environment

```
python -m venv venv  
source venv/bin/activate # On Mac/Linux  
venv\Scripts\activate # On Windows
```

3. Install Dependencies

```
pip install -r requirements.txt
```

4. Database Setup

No manual setup needed.

The SQLite database `humanchain_incidents.db` will be automatically created when you first run the app.

Running the Application:

`python app.py`

Then open your browser at:

`http://localhost:8000`

Available Routes

Route	Methods	Description
/	GET	Home/Login page
/login	GET, POST	User login
/signup	GET, POST	User registration
/logout	GET	Logout the current user
/dashboard	GET	View all incidents dashboard
/incident	GET, POST	Create a new incident report
/update/<id>	GET, POST	Update an existing incident
/delete/<id>	GET	Delete an incident
/search	GET, POST	Search for an incident by ID
/page_not_found	GET	Search for wrong url path

Important Notes

- Incident severity must be either LOW, MEDIUM, or HIGH.
- Passwords are securely handled and stored.
- Basic validation and error handling included.
- Designed for local development and demonstration purposes.

Future Enhancements (Optional)

- Admin roles and permissions
- Frontend improvements (Bootstrap/React)
- Docker containerization
- Deployment to Heroku/AWS

Example requirements.txt

Flask
Flask-SQLAlchemy
bcrypt (for securing the user password)



Thank you for using HumanChain AI Safety Incident Log API Project!

Suraj Gupta.