***Backend api***

This is the first part of backend, it provides secure RESTful endpoints and handles user authentication and authorization. This section of backend acts as a bridge between frontend, database and AI modes, it’s like a communication hub. To understand how it how everything is implemented visit this websit(<https://www.geeksforgeeks.org/python/python-introduction-to-web-development-using-flask/>).

# These values are used to handle the error accordingly (source: greeksforgreeks)

* 400-For bad request
* 401-For unauthenticated
* 403-For forbidden request
* 404-For not found
* 406-For not acceptable
* 425-For unsupported media
* 429-Too many requests

## Dependencies

* Flask
* Flask-JWT-Extended (for authentication)
* SQLAlchemy (for database)

### How it connects to other sections

Frontend (Cidra)

* Uses http requests to call REST API endpoints
* Frontend must include jwt token headers for all protected routes
* API returns Json responses that frontend uses to update the UI

User clicks Login → Frontend sends POST /auth/login → Backend verifies user and generates JWT for the user→ Frontend stores token and uses it for future requests.

Database(Tadi)

* My section uses SQLAlchemy models to communicate with the database
* I will need the database schema and core logic function to store and get data for api endpoints
* Need api routes to be exposed to frontend?

AI(Bokamoso)

* I expose api endpoints for sending ML models and handle authentication and permission checks before forwarding data to AI modules
* I need outputs(predictions) from AI models that I can send back to frontend

Networking(Calem)

* Ensure that requests can reach my API endpoints
* I have too ensure that jwt tokens always work on https

Server(Obakeng)

* Deploy backend on cloud

#### Files

* app.py- Just creates the Flask app, initializes database, registers blueprints, sets up JWT authentication
* seed\_data.py- Populates database with sample data (audit logs, report logs) for testing
* test\_connection.py-Verifioes connection to the database
* models.py-Defind SQLALchemy models for users, logs and reports
* routes/- contain the api routes (auth,users,admin)