Home assignment

Objective:

Your task is to build a simple URL shortener service and ensure it runs seamlessly within a Dockerized environment (as a container).

Assignment Details

1. **Set Up the URL Shortener Service**
   The service expose two REST APIs :
   a. API accept a long URL as a parameter and return a shortened version
   b. API resolve a shortened URL to the original long URL (from a)
2. **Additional requirements**:
   a. **Time-To-Live (TTL)**: Each URL should expire after a configurable amount of time. Default can be infinite if TTL is not provided.
   b. **Analytics endpoint** (`GET /stats/<short_id>`) to return hit count, creation time, and TTL status
3. **Containerization**
   - Write a **Dockerfile** to containerize the application.
   - Ensure that the application runs within a **Docker container**.
4. **Unit Testing (Bonus Task)**
   - Write **unit tests** to validate the core functionalities:
     - Shortening a URL returns a valid short URL.
     - Resolving a short URL returns the correct original URL.
     - Handling invalid or nonexistent short URLs gracefully..
5. **Instructions for Running**
   - Provide clear **README.md** instructions on how to build and run the containerized service.
   - Include commands for building, running, and stopping the service.
   - Provide instructions on how to run tests

General instructions

- Priorities: start from the application before the infrastructure part of the docker container
- You can use any help from internet as long as you understand what and why you are using it. AI is valid but not any other help from other person.
- You can use any OO language (java, python, c++ etc)
- Refer to REST APIs best practices
- To store your data (if needed) you can use any DB or in memory DS - choose the one that easy to work with for the simplicity and the time you have
- Test your code and try to think about scale and edge cases
- We prefer to see something working even if its not perfect then something that don't

Deliverables

- The full project in a **GitHub repository** or as a ZIP file.
- A **Dockerfile** for containerization.
- A **README.md** with setup and usage instructions.