URL Shortener Service Assignment

Overview

Design and implement a **RESTful URL shortening service** similar to bit.ly or tinyurl.com. This service should allow users to submit long URLs and receive shortened versions that redirect to the original URLs when accessed.

Requirements

Core Functionality

- Create an API endpoint to accept a long URL and return a shortened URL.
- Create a redirection endpoint that redirects shortened URLs to their original destinations.
- Implement proper error handling for invalid or expired URLs.
- Ensure generated short URLs are unique and reasonably short.

Technical Requirements

- Implement the service using any backend language and framework you're comfortable with (e.g. Ruby/Rails, Node.js, Python, Java, Go).
- Use a database of your choice for persistence (SQL or NoSQL)
- Include unit and integration tests with reasonable coverage.
- Document your API endpoints.

Bonus Features (Optional)

- Implement user authentication to allow users to manage their shortened URLs.
- Add analytics to track the number of times a shortened URL is accessed.
- Create a simple frontend interface for shortening URLs.
- Add custom URL support (allowing users to choose their shortened path).
- Implement URL expiration functionality.

Submission Guidelines

- Create a private GitHub repository and add our team member as a collaborator.
- Make **r**egular, meaningful commits to show your development process.
- Include a README with:
    - Setup and running instructions
    - API documentation

    - Architecture overview
    - Design decisions and tradeoffs
    - Future improvements (if you had more time)
- Ensure your solution can be easily set up and run locally.

Evaluation Criteria

We'll evaluate your submission based on:

- Code quality and organization
- System design and architecture
- Test coverage and quality
- Documentation clarity
- Problem-solving approach
- Handling of edge cases

Time Expectation

This assignment is designed to be completed within **6-8 hours**. Focus on delivering a working solution with clean code rather than implementing all possible features.

Next Steps

After submission, we'll review your code and schedule a pairing session to:

- Discuss your implementation.
- Walk through the code together.
- Potentially extend the functionality.
