Golang test task

Develop a REST API Server in Go for URL Health Check

The primary objective of this task is to create a RESTful API server using Go (Golang). The server will accept a list of URL addresses via a REST API endpoint, and then ping each URL to determine if they are active (i.e., responding with an HTTP 200 status code). The server should then return the status of each URL to the client.

Key Requirements

- API Endpoint Creation:
    - Develop a REST API endpoint (e.g., /ping-urls) to accept a JSON payload containing a list of URLs.
    - Validate the incoming JSON for proper URL format.

- URL Pinging Functionality:
    - Implement a function to ping each URL in the list.
    - Check if each URL responds with an HTTP 200 status code, indicating that the URL is active.

- Response Handling:
    - The server should respond with a JSON object containing each URL and its status (e.g., {"http://example.com": "active", "http://example2.com": "inactive"}).
    - Handle potential errors such as timeouts or non-200 responses.

- Logging:
    - Implement logging for key events and errors.

Bonus

- unit tests
- implement feature for stop ping process if one of url respond with fail
- add Readme file with instructions

Python (Django) test task

Develop a Django REST-Api for Event Management

The primary goal of this task is to create a Django-based REST-Api that manages events (like conferences, meetups, etc.). The application will allow users to create, view, update, and delete events. It should also handle user registrations for these events.

Key Requirements

- Design an Event model with fields such as title, description, date, location, and organizer.
- Implement CRUD (Create, Read, Update, Delete) operations for the Event model.
- Basic User Registration and Authentication.
- Event Registration
- API documentation
- Docker
- Readme file

Bonus Points

- Implement an advanced feature like event search or filtering.
- Add a feature for sending email notifications to users upon event registration.