Web Servers Monitoring System

Bylith Backend Home Assignment

- Confidential, Private Use Only -

Introduction

The purpose of this assignment is to assess your **way of thinking**, **adapting to new technologies** and **quick learning**. You will be measured by your **level of code** and its **efficiency, stability, reusability, and modularity** throughout the solutions you've come up with. You are expected to **handle/avoid errors**, either in your code or from external resources.
It is **highly recommended** to use Object Oriented Programming (OOP) for your implementation: design patterns, logical capsulations, reusable methods, and OOP Best Practices.

Assignment Introduction

In this assignment you are requested to develop a system that will enable health monitoring of webservers in the cloud.
A Webserver data structure will include by minimum, the following fields:

- Name
- HTTP URL

(You can add any other field you find useful for the assignment)

Core Functionality

1. Ability to add / edit / delete / list webservers
2. Development of automated worker that will monitor the webservers status
    a. Each webserver should be sampled at least 1 time per minute
    b. Webserver success status is determined by two factors: (AND)
        i. Getting HTTP Response **Code 2xx**
        ii. HTTP Response Latency < 60 seconds
    c. Every monitor request should be saved in a dedicated requests table for later use (History)
    d. Server is defined as "Healthy" in case 5 consecutive requests **are** considered "Success" as defined above
    e. Server is defined as "Unhealthy" in case 3 consecutive requests **aren't** considered "Success" as defined above
3. Development of a REST API including the following endpoints:
    a. **C**reate Webserver – Endpoint that will allow creating a new Web Server
    b. **R**ead (Get) Webserver – Should include all basic webserver details, current health status and last 10 requests objects
    c. **U**pdate Webserver – Endpoint that will allow updating Web Server
    d. **D**elete Webserver – Endpoint that will allow deleting Web Server
    e. Get list of all Web Servers and their current health-status
    f. Get list of a specific webserver requests history

Assignment Technologies & Environment

1. Languages: Node.js, PHP, Python, Java
2. Databases: MySQL, MariaDB or PostgreSQL (SQL Based DB)

Bonus

1. Learn and implement how the system should process non-success (!2xx) HTTP Response Codes, according to the recommended HTTP Protocol standards (i.e., 3xx, Retry-After, etc.)
2. Send a notification e-mail to list of pre-defined admins once a server becomes unhealthy
3. Design the system that it'll be extendable to support any potential protocol (i.e., FTP, SSH, etc.)
    a. You can use the following free FTP server for tests: https://dlptest.com/ftp-test/

Assignment Submission

Your assignment submission must include:

1. **A Readme** file with instructions on how to **boot & use** the project
2. SQL DB Dump
3. Operational Postman File that simulates the API Methods created (URLs, Parameters, etc.)

The project should be submitted to a **private** repository in GitHub, and given Read-Only access to:
Asaf:
GitHub asaf@bylith.com (asafrokach)
Aviram:
GitHub aviram@bylith.com (aviramaz)

Note: *This project is confidential, should be reviewed by Bylith representatives only and not forwarded / used by / presented to anyone else.*

For any question/problem, please contact the HR Team.

Once the assignment is shared, please also notify the HR Team by sending an email to the following mailbox: assignments@bylith.com

Good luck!