

Technical Test: Fullstack Developer

Candidate Name : Sia Kevin Septian Saputra
Applied Position : Full Stack Developer
Test Date : 10 January 2025

PROGRAMMING

Duration: 24 hours

General Instructions:

Please read the instructions carefully before proceeding. Failure to comply with these instructions will only cause disadvantages.

1. The use of any framework is free as long as the framework version is still actively maintained.
2. For scoring, each item will consist of several aspects, such as: Coding style (Best Practice), logic, database structure design and query.
3. Your code tidiness is considered.
4. Please be mindful of the security aspect of your implemented source code.
5. Adequate security measures will be a plus.
6. Database migration is required.
7. The use of feature tests or unit tests is a plus.
8. For frontend development, use frameworks or UI libraries such as React, Vue, Svelte, etc. (no jQuery).
9. Data communication between Backend and Frontend preferably using API.
10. After finishing this task, please put all your work files into a .zip
 - a. compressed file with file name format as follow:
 - b. [test date]-[your name]-[position applied].zip
11. The zip file must contains:
 - a. Source Code
 - b. Simple user guide on how to configure your implemented system (such as: in which file database configuration is located, what files has to be edited to run the system) (***make the instructions as easy as possible, think of us as beginners**)
 - c. Simple note about system features.
 - d. API Specification Document (Swagger, Postman Collection, or Text Document) (***if backend and front end are separate**)

12. The application must be delivered in 24 hours. Every 30 minutes delay will deduct 10 points of your final score.

Exam Requirement:

Create a mini application for a Todo List. The application should allow users to add, edit, and delete tasks, as well as mark tasks as completed. Use the following guidelines:

1. The application should have a user interface that displays the list of tasks.
2. Users should be able to add a new task by entering a task description in an input field and clicking an "Add" button.
3. Each task in the list should display the task description and a checkbox to mark it as completed.
4. Users should be able to edit a task by clicking on its description and modifying it in an editable field.
5. Users should be able to delete a task by clicking on a delete button next to it.
6. Users should be able to changing the order of the task list
7. Completed tasks should be visually distinguished from incomplete tasks (e.g., strike-through text or different background color).

You are free to choose any front-end framework or library (e.g., React, Angular, Vue.js) and back-end language or framework (e.g., PHP, Node.js with Express, Ruby on Rails, Django, GO-Lang) to implement the application.