

Full stack Challenge

Description

This interview challenge is designed to assess a candidate's ability to build a full-stack web application using a combination of front-end and back-end technologies. The challenge will involve creating a simple web application for managing tasks, allowing users to add, edit, delete, and view tasks. The candidate can choose their preferred front-end framework and back-end technology for this challenge.

Assignment

Front-End *(Choose one: Angular, React, Vue, Svelte, etc)*

- Create a user interface for managing tasks. The interface should have the following features:
 - Display a list of tasks with their titles and statuses (completed or not).
 - Allow users to add a new task with a title and status.
 - Allow users to mark a task as completed or incomplete.
 - Allow users to edit the title of a task.
 - Allow users to delete a task.
- Implement a responsive design for the front-end to ensure usability on both desktop and mobile devices.
- Use state management (e.g., Redux, Vuex, or React Context) to manage the application's state.
- Fetch and send data to the back-end using HTTP requests (e.g., RESTful APIs or GraphQL).

Back-End *(Choose one: Node.js, .NET, Go, Python, etc)*

- Create a server that provides RESTful API endpoints for managing tasks. The API should support the following operations:
 - Retrieve a list of tasks.
 - Add a new task.
 - Update the title and status of a task.
 - Delete a task.
- Implement proper error handling for API requests, including validation of incoming data.
- Use a database (e.g., MongoDB, PostgreSQL, MySQL, or SQLite) to store and retrieve task data. You can choose the database system that you are most comfortable with.

Considerations

- Ensure that the front-end and back-end are properly connected and can communicate effectively.
- Handle security considerations such as input validation and protection against common web vulnerabilities (e.g., SQL injection, CSRF, XSS).
- Write clean and maintainable code, following best practices for the chosen stack.
- Provide clear instructions on how to run the application locally.
- Include a README.md file with information about the technologies used, project setup instructions, and any additional notes.
- The developer is expected to apply best practices for security, scalability, and efficiency in the implementation.

Bonus

- Implement user authentication (optional but a plus). Allow users to register, log in, and associate tasks with specific users. Use tokens or sessions for authentication.