

# SOFTWARE ENGINEER

## CASE

# Software Engineer Position at Playable Factory

Thank you for your interest in the Software Engineer position at Playable Factory.

We're excited about your application and look forward to assessing your skills through this case study.



# Task: Develop a To-Do App with ChatGPT Integration

## Requirements:

- **Develop a To-Do App with the following specifications:**
  - Use React.js for the frontend and Node.js for the backend. ✓
  - Implement two separate applications that communicate via a REST API. ✓
  - Use either Express.js or Nest.js for the backend. ✓
  - Use React.js or any React framework for the frontend. ✓
- **The app should include the following pages:**
  - Login Page: Implement user authentication using either JWT or Firebase. ✓
  - To-Do Management Page: Provide functionality to add, edit, delete, and list to-dos. ✓
- **Use validation libraries such as joi, express-validator, or zod for the backend.** ✓
- **Use a real database (no fake database is allowed).** ✓
- **Design the UI using libraries like Ant Design or Mantine.** ✓
- **Provide a README.md file with clear instructions on how to:** ✗
  - Run the application.
  - Set up the database and manually create a user.
- **Restrict access so that each user can only view and manage their own to-dos.** ✓
- **Implement the following features for to-dos:**
  - Image Upload: Use images as thumbnails for to-dos, displayed as small images in the to-do list. Ensure that only valid images are uploaded. ✗
  - File Upload: Allow files to be attached to to-dos, which can be downloaded by the user. ✗
  - Text Search: Provide backend support for searching to-dos by text. ✗





## ChatGPT Integration for To-Do Recommendations:

In addition to the core to-do management features, you will integrate ChatGPT to provide automatic recommendations for each to-do item. The objective is to generate AI-based suggestions that help users manage their tasks better and display these recommendations alongside the to-dos.

### 1. Add To-Do with Recommendation:

- When a new to-do is created, the content of the to-do will be sent as a prompt to ChatGPT through an API request.
- ChatGPT will generate recommendations based on the to-do content, such as suggested priorities, potential categorization, or helpful tips.
- The generated recommendation will be stored in the database alongside the to-do item.

### 2. Update To-Do and Update Recommendations:

- Whenever a to-do item is updated (e.g., the task description, priority, or tags are changed), the content of the updated to-do will be sent again to ChatGPT as a new prompt.
- ChatGPT will generate new recommendations for the updated to-do.
- The new recommendations will replace the old ones in the database, ensuring that the recommendations reflect the current state of the to-do item.

### 3. Database Schema:

- The to-do schema will include a field for storing recommendation (e.g. recommendation).
- Both the to-do content and the recommendation will be stored together in the database.
- Whenever the to-do is updated, the associated recommendation will also be updated.

### 4. Displaying Recommendations:

- On the front-end, each to-do item will display its corresponding recommendation next to it, giving users useful suggestions or insights based on their tasks.
- The recommendations should update in real-time as the to-do item is modified.



### Bonus Features:

- Use TypeScript for the application. ✗
- Use MongoDB as the database. ✓
- Provide Dockerfiles and a Docker Compose file for containerization. ✗
- Deploy the application using a cloud service provider. ✗
- Allow tagging of to-dos and filtering the to-do list by tags. ✗
- Implement pagination, with logic handled on the backend. ✓

### Evaluation Criteria:

- Code organization and readability. ✓
- Proper and modern usage of packages and tools. ✓
- Absence of performance issues. ✓
- Avoidance of anti-patterns or practices that could lead to bugs. ✓

### Version Control:

- The project should be managed using Git version control ✓
- Make use of git branches for feature development, bug fixes, and code reviews ✓
- Ensure that the commits history is clean and logically organized. ✓





You have 3 days!  
Good luck!