Better.

Associate Software Engineer (Python /React)

Assessment

Role Overview

We are a software development agency working with enterprises and startups to create high-quality, scalable software from the ground up. As a Software Engineer, you'll play a key role in building robust applications that prioritize maintainability and simplicity. This role involves designing and developing software with strong engineering foundations, using TypeScript, React, and Python. You'll focus on delivering efficient, clean code and will be involved in every step, from database modeling to implementing automated tests and reviewing code.

Background

Better Software is an AI powered software development agency and build web, mobile and custom applications with solid engineering foundations. We are preferred agency by top notch entrepreneurs, enterprises and ventured backed startups.

Unlike 99% of agency-built software that struggles with poor foundations, leading to costly rebuilds and bug-ridden iterations, we prioritise creating a rock-solid technical base from the start. This ensures smoother scaling and easier future development. Over the last 7 years, we've had the privilege of working with bootstrapped companies that have gone on to generate over $100M in revenue, as well as unicorns and startups backed by Techstars, top VCs like Andreessen Horowitz, and executives from industry giants like Apple and Google.

Task

Thank you for expressing the interest. For us to better understand your skills, please attempt the following assessment:

1. Problem Statement

Using our open-source template in Flask + React

Task#1: Build backend APIs to add, edit, delete comments for a given task using proper CRUD principles. Also include automated tests for these APIs.
Task#2 (bonus): Build the frontend interface to add, edit, delete tasks, using the existing CRUD APIs.

2. Instructions for Submissions:

1. You may have questions — feel free to make reasonable assumptions and clearly mention them in your PR/video.
2. Fork the repo into your personal GitHub and raise PRs against your fork. Do not raise PRs against the original repo.
3. Follow the existing code structure of the flask-react-template
4. When raising a PR, please follow this etiquette guide
5. Avoid using AI tools to generate your solution. We're interested in understanding your thinking and approach.

3. What to Submit:

1. Links to separate PRs for each task (Task 1 and Task 2).
2. A short video walkthrough of the PRs explaining: (a) Your approach (b) Key decisions you made (c) Any trade-offs or technical debt you took on

PS: Due to the high number of applicants, we will not be able to provide individual feedback if you are not selected.