Software Requirement Specification

Problem Statement: A few sentences to describe the problem you are trying to solve, i.e., justify why this software is needed.

The problem we are addressing is creating a platform where individuals can share and collaborate on online assessment questions, interview questions, and related processes. The software is necessary because it serves two primary needs: facilitating a space for users to share and exchange knowledge about assessments, and enabling users to connect with others who are preparing for similar assessments. By fostering communication and collaboration, the platform helps individuals study and prepare more effectively. This collaboration tool is essential for users who want to improve their chances of success through shared learning experiences.

Potential Clients: Who are influenced by this problem and would benefit from the proposed solution? (i.e. the potential users)

The primary users impacted by this problem and who would benefit from the solution are job seekers, students, and professionals preparing for interviews or assessments. They can access shared resources and connect with others facing similar challenges. Recruiters, HR professionals, and career coaches would also benefit by gaining insights into common assessment questions and providing better guidance. Educational institutions and mentors can use the platform to foster collaboration and improve learning outcomes, making the preparation process more effective for all.

Proposed Solution: Write a few sentences that describe how a software solution will solve the problem described above.

The software solution will provide a centralized platform where users can share and access online assessment questions, interview questions, and related resources. It will facilitate collaboration by allowing users to connect with others preparing for the same assessments, enabling group discussions and study sessions. By streamlining access to shared knowledge and fostering communication, the platform will help users prepare more effectively and improve their chances of success.

Functional Requirements: List the (functional) requirements that software needs to have in order to solve the problem stated above. List these in role-goal-benefit format. It is useful to try to group the requirements into those that are essential (must have), and those which are non-essential (but nice to have).

Must have

1. User Login System: As a registered user, I want to securely log in to my account so that I can access personalized content and manage my interview preparations.

2. Main Page (Navigation for Companies, Job Titles, Seasons): As a user preparing for interviews, I want to browse and add companies, job titles, and hiring seasons so that I can track and organize my application process effectively.

3. Submission Review System: As a platform administrator, I want to review all user submissions so that I can ensure the accuracy and quality of information shared on the platform.

4. Forum Functionality (Posts, Comments, Private Chat): As a user preparing for interviews, I want to post in the forum and comment on other posts so that I can gain insights from others and share my own experiences. As a user, I want to have a private chat option so that I can discuss interview strategies confidentially with other users.

5. Find Partners Nearby (Map): As a user preparing for interviews, I want to find nearby students also preparing for interviews so that I can form study groups and prepare together efficiently.

Nice to have

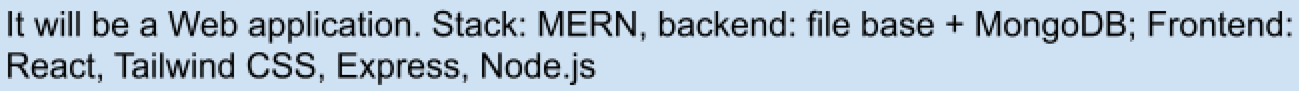
1. Company Leaderboard Page:
   As a student seeking internships or job opportunities, I want to share my positive/negative experiences with companies and view a ranking board based on these shared experiences, so that I can make more informed decisions about potential employers and help other students do the same.

Continues on the next page

Non-functional Requirements:

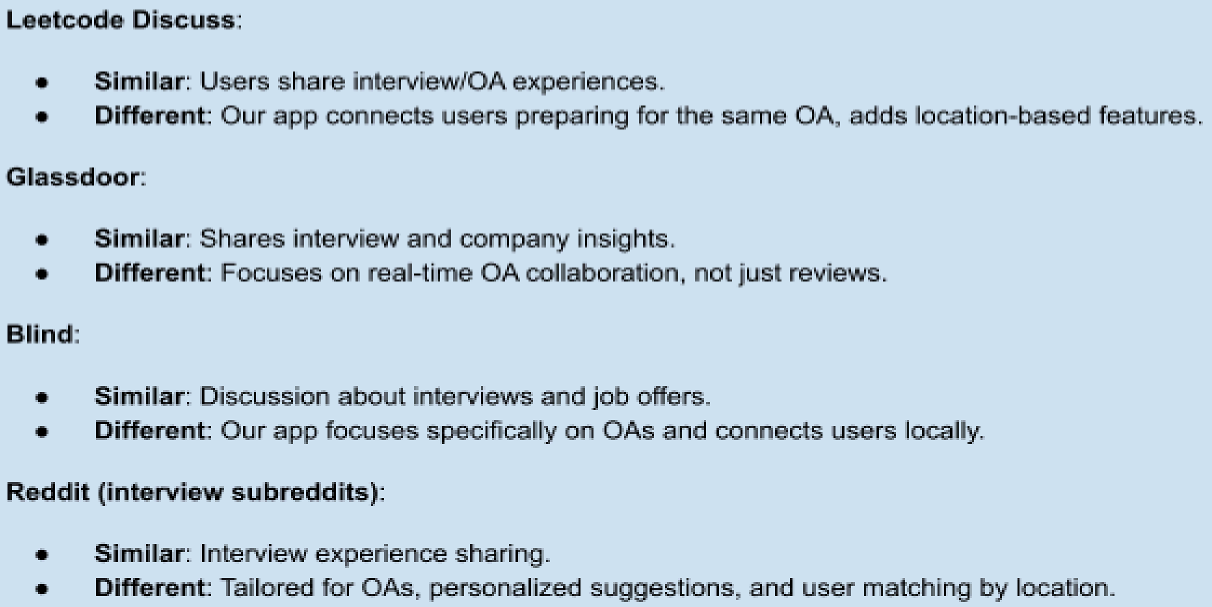
User-Friendly: Simple, intuitive interface.
Data Security: Encrypt personal and location data.
Performance: Fast, smooth, and responsive.
Scalability: Handles growth without lag.
Reliability: 99.9% uptime.
Privacy: Compliant with data protection laws.
Maintainability: Easy to update and fix.
Cross-Platform: Works on all devices.

Software Architecture & Technology Stack: Will this be a Web/desktop/mobile (all, or some other kind of) application? Would it conform to specific software architecture? What programming languages, frameworks, databases, …, will be used to develop and deploy the software?



It will be a Web application. Stack: MERN, backend: file base + MongoDB; Frontend: React, Tailwind CSS, Express, Node.js

Similar Apps: List a few similar applications to the one you are developing. Don't be eager to conclude no similar app exists! There is always something similar to what you are building! Finding those will help you to better specify your project. ***You must be prepared to explain how your app is different from the existing ones.***



Leetcode Discuss:

- **Similar**: Users share interview/OA experiences.
- **Different**: Our app connects users preparing for the same OA, adds location-based features.

Glassdoor:

- **Similar**: Shares interview and company insights.
- **Different**: Focuses on real-time OA collaboration, not just reviews.

Blind:

- **Similar**: Discussion about interviews and job offers.
- **Different**: Our app focuses specifically on OAs and connects users locally.

Reddit (interview subreddits):

- **Similar**: Interview experience sharing.
- **Different**: Tailored for OAs, personalized suggestions, and user matching by location.