

Team 15 Project Specification

Job Hunting & Interview Experience Platform

Project Overview

This project is a job hunting and interview experience platform that allows users to:

1. Share and explore interview experiences.
2. Track their own job application progress.
3. Practice coding problems with AI-driven feedback.
4. Participate in live coding mock interviews.

This platform aims to streamline job seekers' preparation by offering a collaborative and AI-powered experience.

Team Members

- Boning Xing (boningx)
- Jiachun Xu (jiachunx)
- Yuwei Da (yuweid)
- Shenyu Tao (shenyut)

Product Owner for First Sprint

- Boning Xing (boningx)

Product Backlog

A. Posts & Discussion Module

- Users can **create posts** to share interview experiences.
- Posts can be **categorized** by:
 - Company (e.g., Google, Amazon, etc.).
 - Interview round (e.g., OA, Technical Interview, HR Interview).
 - Post type (Interview experience, Resume submission, General discussion).
 -
- **Timeline** Feature: Linked Interview Experiences
 - When creating a post, users can link it to their previous interview rounds for the same company.
 - In the post details page, related interview rounds will be displayed at the bottom as clickable links, allowing users to navigate through the entire interview process.
- Users can **comment, like, share, and quote** posts.
- Posts can be **searched, filtered, and sorted**.

B. User Dashboard & Job Tracking

- Users have a **personal dashboard** to:

- Track **applied job positions**.
 - View a **timeline of interviews**.
 - See **statistics** on job application progress.
 - Manage **interview experience records**.
- Users can filter and visualize their **job search history**.

C. AI-Powered Coding Practice & Mock Interview

In many real technical interviews, candidates are required to solve coding problems while explaining their thought process aloud. Our platform provides an AI-powered mock interview environment that simulates this experience.

- Users will be presented with a coding problem.
- While solving the problem in the coding area, users are encouraged to explain their approach in real-time.
- The built-in AI mock interviewer listens to the explanation and analyzes the code being written.
- AI provides real-time feedback, asks clarifying questions, and suggests improvements, simulating the behavior of a real technical interviewer.
- The system will also validate the user's solution by running test cases and providing feedback on correctness and efficiency.

This feature mimics real technical interviews, helping users practice and improve their coding and problem-solving communication skills.

D. Company & Job Insights

- The system generates **company-specific data analysis** based on posted data.
- When searching for a company, an **overview post** appears at the top, summarizing interview information.

First Sprint Backlog

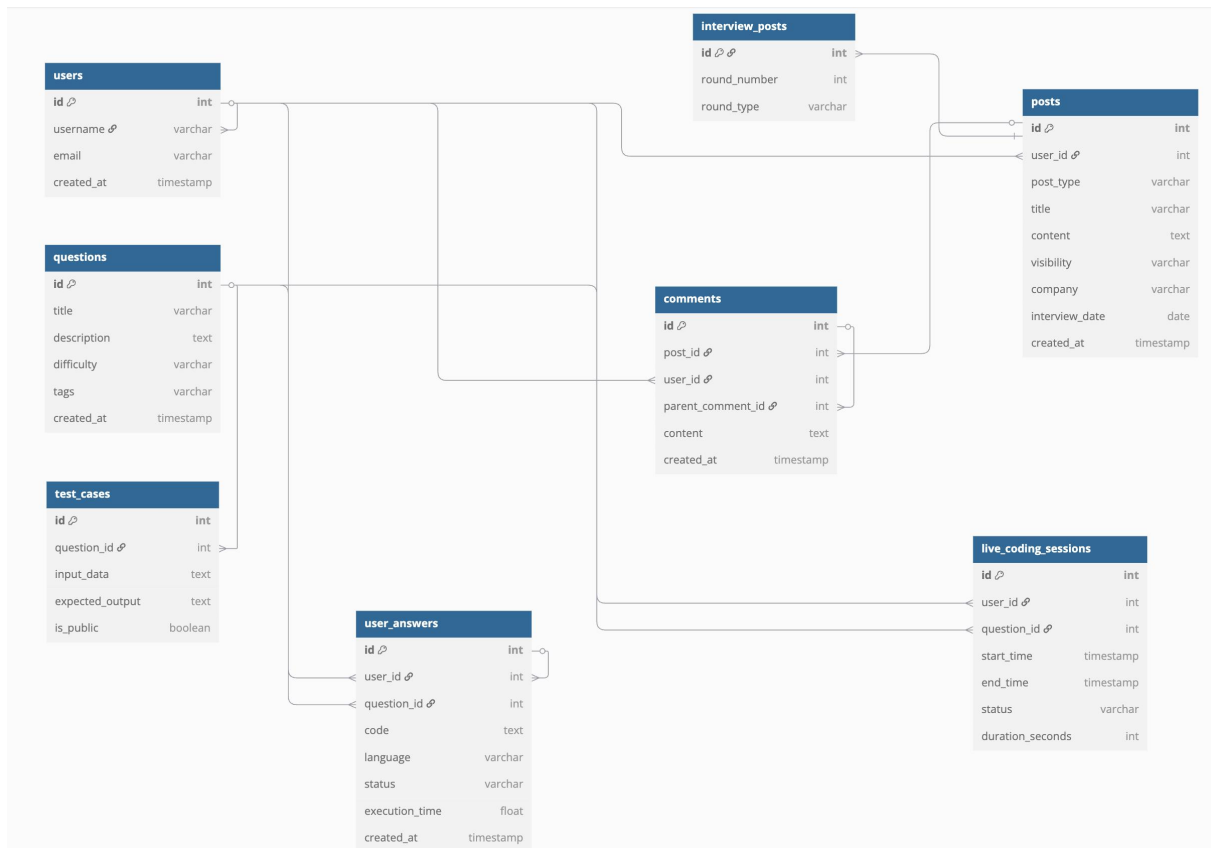
For **Sprint 1**, the focus is on **building a minimal demo** that supports:

- A simple web page where a user can enter a solution to a coding problem.
- Backend **compilation** support for running user-submitted code.
- Displaying execution results on the web page.

Data Models

- **System Requirements**
 1. Users can share interview experiences.
 2. Users can track their own interviews (private records).
 3. Users can comment & discuss.
 4. Users can practice coding questions.
 5. Live coding sessions should be supported.
 6. Posts should be linked by interview rounds into a timeline.

- **ER Diagram**



- **Database Schema**

The database consists of the following tables:

1. Users Table

```
Table users {
  id int [primary key] // Unique user ID
  username varchar [unique] // Display name
  email varchar [unique] // Email for authentication
  created_at timestamp
}
```

2. Posts Table

```
Table posts {
  id int [primary key] // Unique post ID
  user_id int [ref: > users.id] // Post belongs to a user
  post_type varchar // "interview", "general", "company_review",
  "resume_submission", "personal_record"
  title varchar // Title for general posts
  content text // Post content
  visibility varchar // "public" or "private"
  company varchar [null] // Company name (if applicable)
  interview_date date [null] // Date of interview round (if applicable)
}
```

```
    created_at timestamp
}
```

3. Interview Posts Table

```
Table interview_posts {
    id int [primary key, ref: > posts.id] // Links to posts
    round_number int // 1, 2, 3... (which round is this?)
    round_type varchar // Example: "Online Assessment", "Technical Interview"
}
```

4. Comments Table

```
Table comments {
    id int [primary key] // Unique comment ID
    post_id int [ref: > posts.id] // Comment belongs to a post
    user_id int [ref: > users.id] // Comment made by a user
    parent_comment_id int [ref: > comments.id, null] // NULL = top-level comment, else reply
    content text // Comment text
    created_at timestamp
}
```

5. Questions Table

```
Table questions {
    id int [primary key] // Unique question ID
    title varchar // Question title
    description text // Full question text
    difficulty varchar // "Easy", "Medium", "Hard"
    tags varchar // Tags (e.g., "Array, Dynamic Programming")
    created_at timestamp
}
```

6. Test Cases Table

```
Table test_cases {
    id int [primary key] // Unique test case ID
    question_id int [ref: > questions.id] // Linked to a question
    input_data text // Example: "2\n3 4"
    expected_output text // Example: "7"
    is_public boolean // If true, the user sees this test case; else, it's hidden for evaluation
}
```

7. User Answers Table

```
Table user_answers {
    id int [primary key] // Unique submission ID
    user_id int [ref: > users.id] // User who submitted
```

```

question_id int [ref: > questions.id] // Question being solved
code text // User's code submission
language varchar // "Python", "Java", "C++"
status varchar // "Accepted", "Wrong Answer", "Timeout", "Compilation
Error"
execution_time float // Execution time in seconds
created_at timestamp
}

```

8.Live Coding Sessions Table

```

Table live_coding_sessions {
  id int [primary key] // Unique session ID
  user_id int [ref: > users.id] // User currently coding
  question_id int [ref: > questions.id] // Question being solved
  start_time timestamp
  end_time timestamp [null] // NULL if still in progress
  status varchar // "In Progress", "Completed"
  duration_seconds int // How long the user took (calculated at the end)
}

```

- **How the Design Meets Requirements**

The following table shows how the design meets the system requirements:

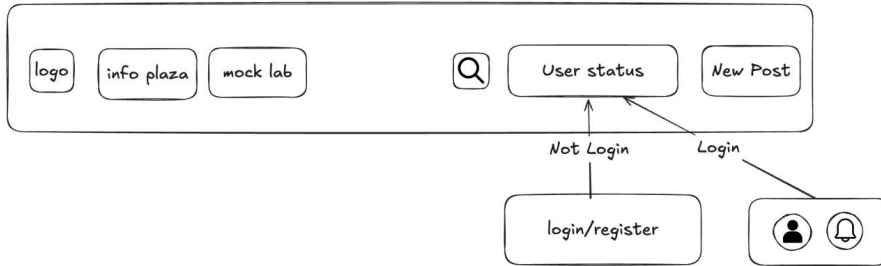
Requirement	Solution
Users can share interview experiences	posts table stores interview rounds & related content.
Users can track their own interviews (private records)	posts.visibility = 'private' enables private tracking.
Users can comment & discuss	comments table enables discussion on posts.
Users can practice coding questions	questions table stores problems, test_cases stores evaluations.
Live coding sessions should be supported	live_coding_sessions table tracks user progress.
Posts should be linked by interview rounds	Query by company and interview_date instead of a separate relationship table.

Wireframes & User Flow

The **UI wireframes** demonstrate the **navigation flow** and **functional layout** of the system:

- **Headers**

Header



Company Interview statistics



- **Login/Registration Page**

login




Header

Login

User name:

Password:

login

Login with:   

registration

Header

Registration




User name:

Password:

Password again:

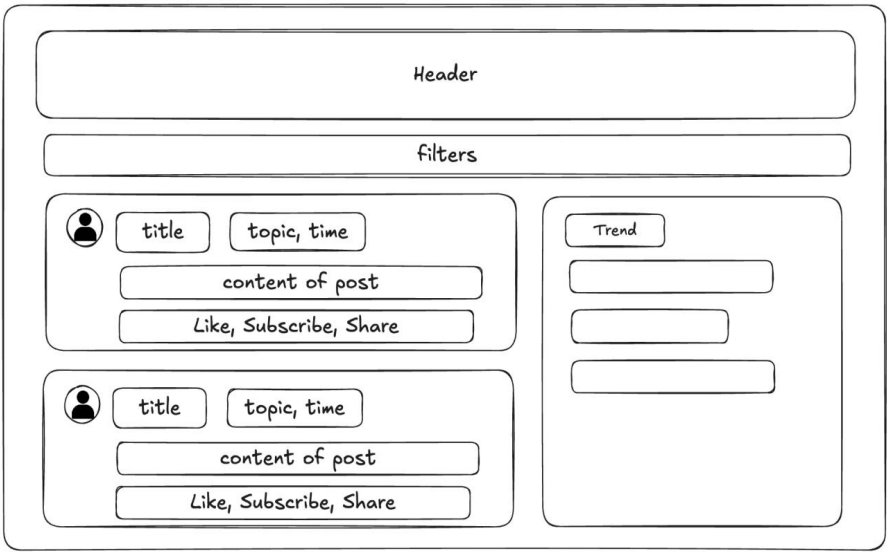
Email:

register

Register with:   

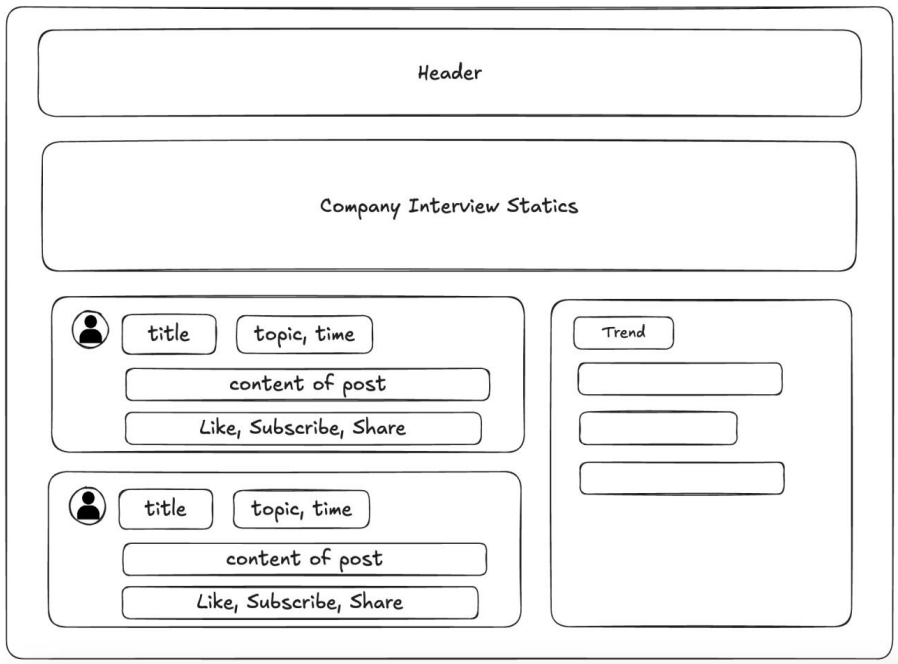
- **Home Page** – Displays interview experience posts.

Global Post List



- **Home Page after searching for certain company** – Displays the company interview statics post in the first place.

Company Post List



- **Post Creation Page** – Allows users to share experiences.

Create a post

Common postInterview ExperienceCompanyPhaseTime

Please type in post title here.0/40

Tool Box(Bold, italic, underline, @, upload text)

Main body(plain text or markdown format)

Add tags (optional)draft boxSave as draftSubmit

- **Post Details Page** – Shows post content, discussion, and related interviews.

Post Details

Header

Post title

Post content

Like, quote, subscribe, share, message

Comment content

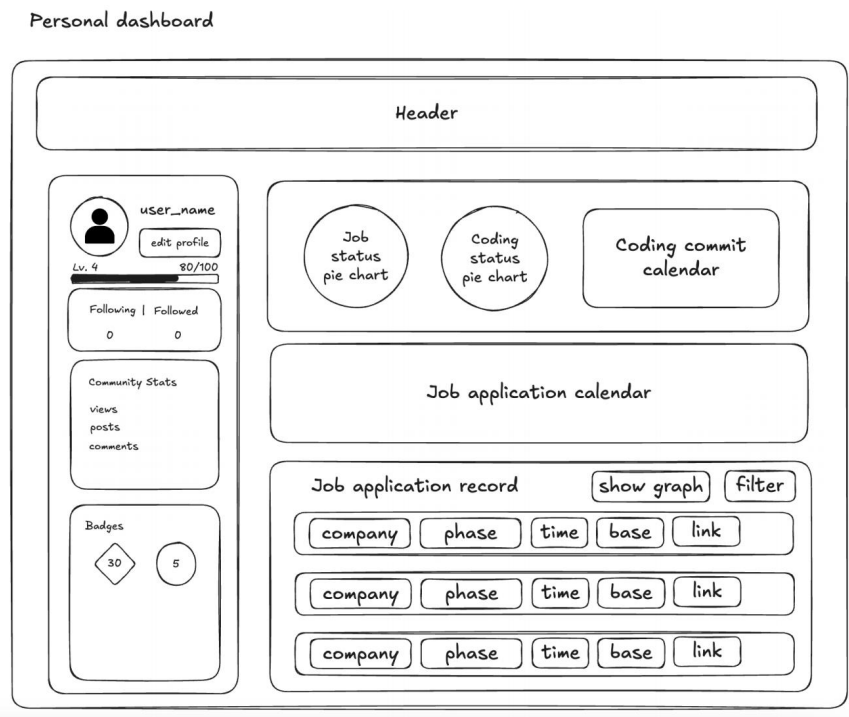
Like, quote, message

Comment content

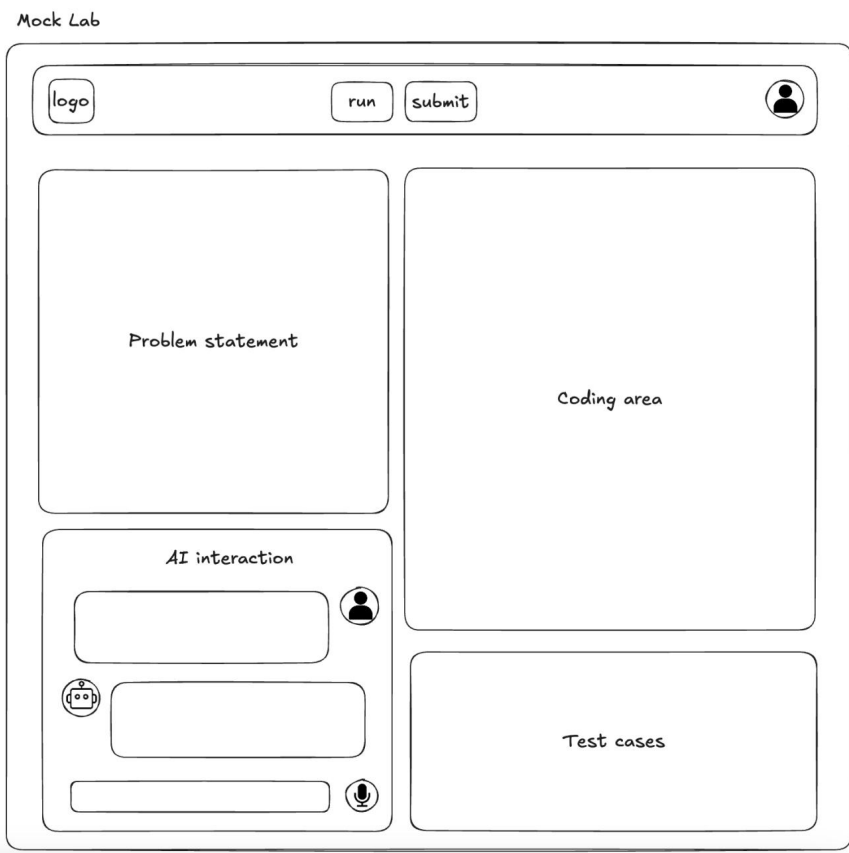
Like, quote, message

Trend

- **Dashboard – Tracks user interview progress and statistics.**

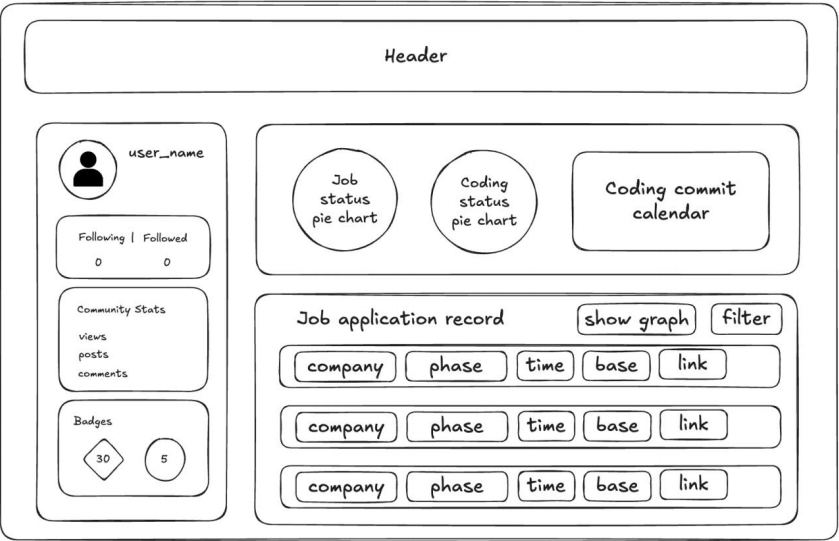


- **Mock Lab – AI-powered live coding interview experience.**



- **User Profile – Displays job tracking & application records.**

User profile



These wireframes guide the **frontend development** and ensure a **consistent user experience**.