

Computer Science Department
Senior Project Final Report – Spring 2025

Title of the project:	AI Interview App with focus on Software Engineering positions
Team Members:	Aimenov Arslan, Askaraliyev Nurzhigit, Askhat Ansar, Sergey Kim
Project Advisor/Co-Advisors	Fatih Demirci
Executive Summary (10%)	
<p>Our project, called “entervue”, provides an AI-powered app for interview training. It covers areas of the IT sphere, such as Software Engineering, Data Science, Database Management, Backend and Frontend Development. It utilizes LLM, Speech-to-Text, and Text-to-Speech tools to simulate mock-interviews with real-time feedback. Additionally, the application has a training mode to help users improve their skills. Users may track their progress and see the results of past interviews. The admin panel allows an administrator to see information about all users and their progress. Project was evaluated by manual testing and user feedback collected with Google Form.</p>	
Introduction (10%)	
<p>Problem Statement. The main problem identified is the underpreparedness of jobseekers when applying for technical job interviews. Lack of realistic practice tools to explore and develop technical and social skills is the main cause. Lastly, existing solutions suffer from lack of interactivity, no ability to simulate a real interview, excessive focus on technical skills or high cost.</p> <p>Motivation and Significance. Even qualified interviewees find themselves unable to articulate their experience or problem-solving approach during interviews. Alternatively, without technical skills, an interviewee will not be able to solve the presented problems. This is our motivation for using a double-pronged approach at addressing gaps in both technical and soft skills. Finally, the growing trend in digital hiring processes makes our solution to the problem relevant and timely.</p> <p>Problem Solution. Entervue is an app that provides a platform to test your skills in both technical tests and simulated interview environments. Through speech-to-text and text-to-speech natural language processing technologies, the app delivers a realistic experience to its users. Powered by AI, entervue offers customizable difficulty levels, job-specific question sets, voice interaction, real-time feedback and performance tracking.</p> <p>Report Organization. This report covers the full scope of the entervue application. This includes problem identification, background, implemented solution, development of the project over two semesters and the evaluation of the final system.</p>	
Background and Related Work (15%)	

Prior Research and Related Work. Several related researches directly informed the design decisions for interview:

Jebreen (2021) analyzed job advertisements in Jordan, in particular, about software engineering positions. Results showed that strong technical skills were insufficient, highlighting the need for communication and adaptability during interviews.

(“Software engineering jobs” or “software engineer” or “software tester” or “Software QA” or “Software development”) or (“Application Development”) or (“programmers” or “Modelling” or "Developers")

Figure 1 Keywords for job advertisements search.

Table 2 Hard skills required by employers.

Rank	Skills	Manifestation
1	Programming	138
2	Requirement Engineering	134
3	Software Testing	122
4	Software Design	98
5	System Analysis	92
6	Project Management	86

Table 3 Soft skills required by employers.

Rank	Skills	Manifestation
1	English language proficiency	169
2	Interpersonal/communication skills	163
3	Problem solving	162
4	Creativity	151
5	Self-learning	146
6	Critical thinking	133
7	Curiosity	121
8	Organizational skills	118
9	Team work	115

Marinovici (2014) explored the hidden requirements during software engineering job interviews. The results showed that problem-solving abilities, teamwork and adaptability were deciding factors beyond the technical ability of a job seeker.

Rusu (2011) proposed a decision-based game simulating software engineering interviews. It was shown that interactive based, game-like scenarios could improve the readiness of interview candidates.

Anton (n.d.) analyzed patterns across behavioral and technical interviews for software engineering. He compiled practical advice, highlighting the usefulness of customizable interview preparation tools.

Together, these papers helped shape entervue, creating the focus on real-time communication and adaptability practice settings. It inspired entervue’s problem-solving questions, technical questions with real time feedback and customizable difficulty with several different job positions.

Existing Solutions and Reasoning for Computing Based Solution. There were three types of existing solutions our research focused on:

Static Q&A Lists. In particular, the Leet Code “Top Interview Questions” section. It offered a large databank of problems, but lacked interaction, adaptability and speaking practice.

The screenshot shows a LeetCode problem page for "Remove Duplicates from Sorted Array". The problem description states: "Given an integer array `nums` sorted in non-decreasing order, remove the duplicates in-place such that each unique element appears only once. The relative order of the elements should be kept the same. Then return the number of unique elements in `nums`." It includes a "Custom Judge" section with a code snippet for testing the solution. On the right side, there is a sidebar with a list of topics: Array, Strings, Linked List, Trees, and Sorting and Searching. The "Array" topic is selected and highlighted in a dark grey box.

Figure 4 & 5. An example problem and a range of topics

The screenshot shows the "Introduction" page for LeetCode's "Top Interview Questions". It features a title "Introduction" with social media icons for Facebook, Twitter, and LinkedIn. The text reads: "This is LeetCode's official curated list of Top classic interview questions to help you land your dream job. Our top interview questions are divided into the following series: 1. Easy Collection, 2. Medium Collection, 3. Hard Collection." It concludes with the statement: "to help you master Data Structure & Algorithms and improve your coding skills."

Figure 6. Ability to choose difficulty

Online Courses. Courses such as UdeMy’s “Ace the Coding Interview,” broadly covered the topic of technical interviews, but lacked the ability to simulate the interview, thus lacking in the practical side of the preparation.

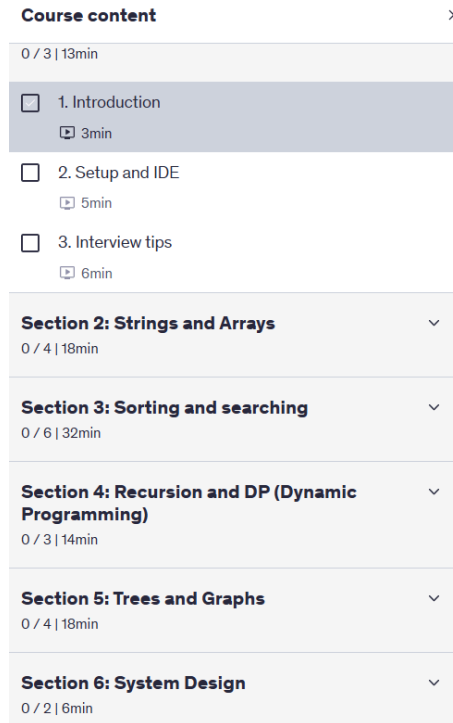


Figure 7. Course content

Flashcard or Quiz Apps. These applications, such as “Quizlet’s Interview Flashcards”, help memorize facts, but the interviewees cannot use them to develop speech clarity, adaptability or dynamic thinking under pressure

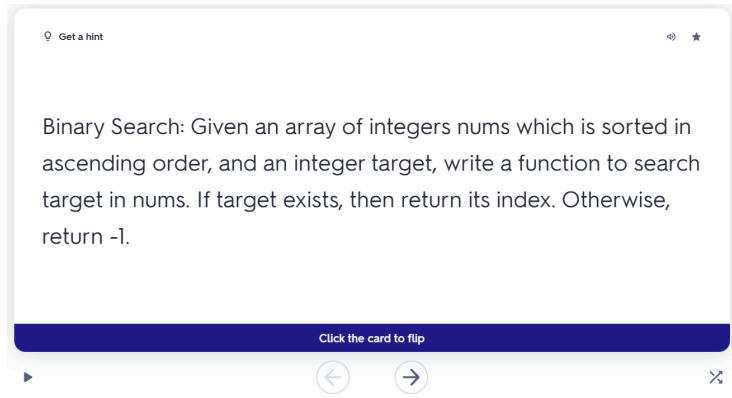


Figure 8. Flashcard problem

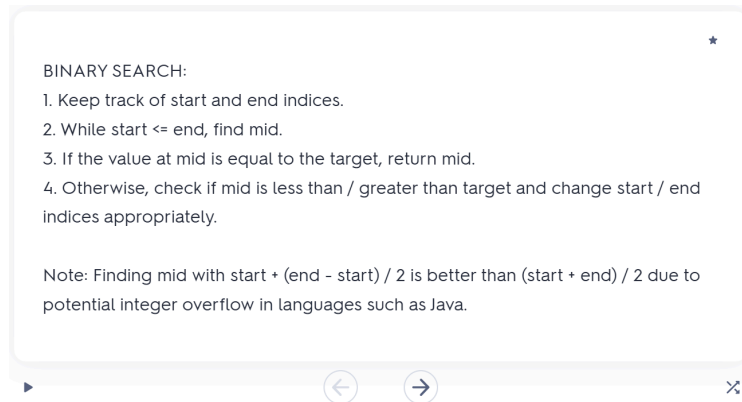


Figure 9. Problem answer

Taking into consideration their strengths and weaknesses, we decided to create a highly interactive, technical preparation focused application. It has all the benefits of Q&A Lists and Flashcards, covers a broad range of job positions and the AI simulation allows the trainees to develop no less important soft skills.

Project Approach (20%)

Detailed Description of Solution. The solution uses client-server architecture with mobile app frontend and backend for data storage, authentication and real-time operations.

Software Architecture:

- Frontend. Developed using FlutterFlow, enabling rapid UI creation with cross-platform capabilities.
- Backend. Built on Firebase, with real-time database functionality, user authentication and data security.

- Data Storage. Interview questions, user answers and profiles are stored in Firebase's SQL Database - Firestore.

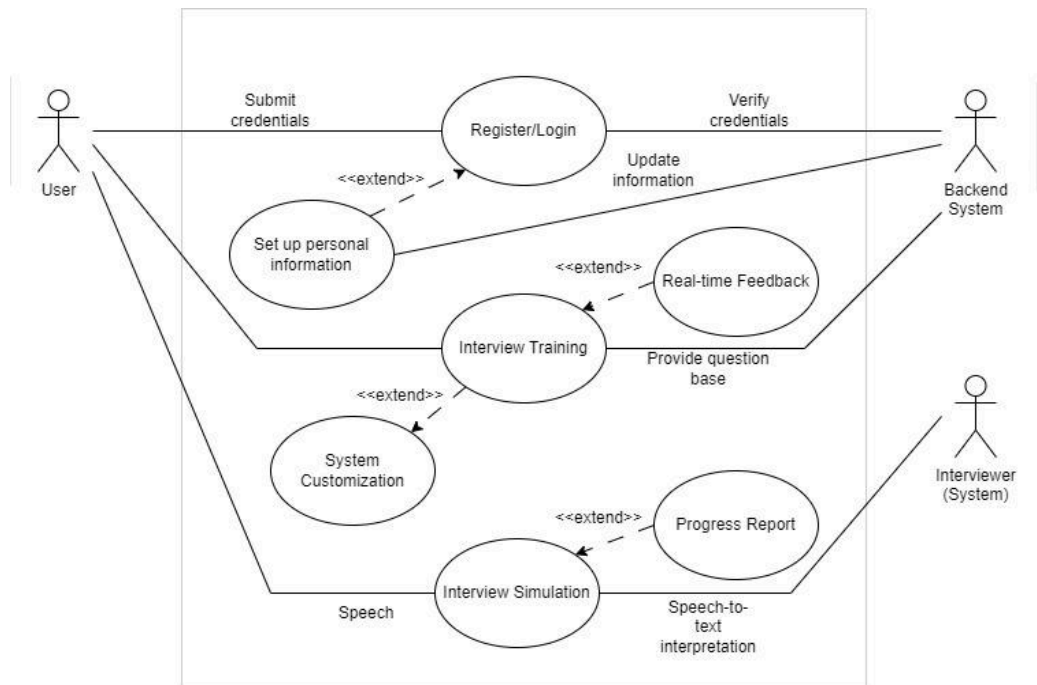


Figure 10. Use Case Diagram

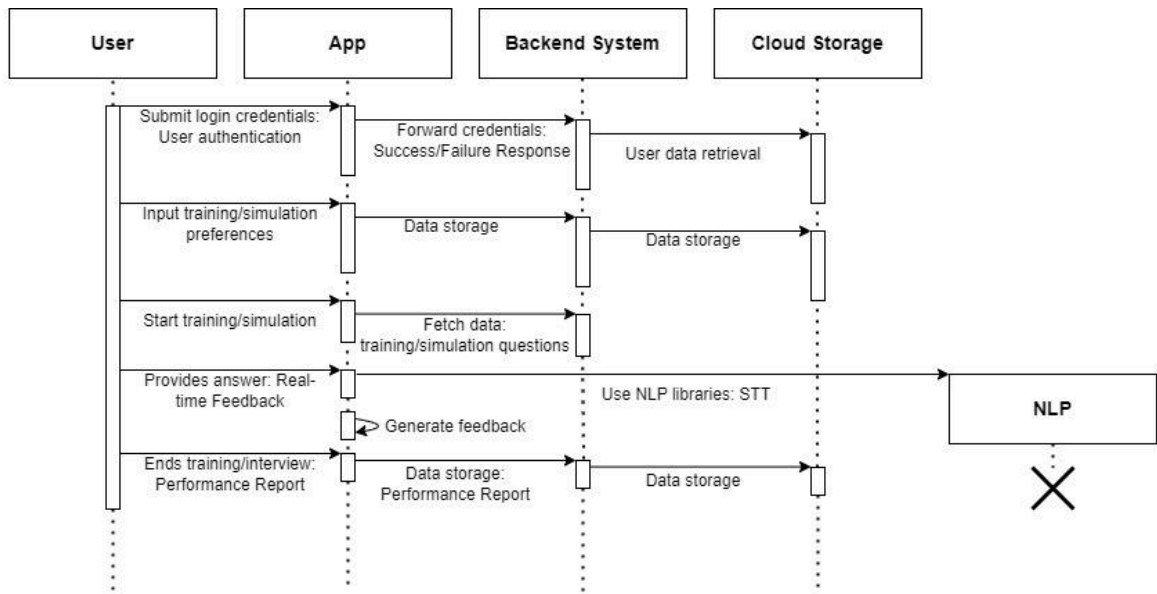


Figure 11. Workflow Diagram

Third-party Tool	Purpose	Integration Summary
FlutterFlow	Frontend app development	Used to design the app's UI/UX from scratch.
Firebase Authentication	User login and registration	Integrated to manage user credentials and authentication securely.
Firebase Firestore	Data storage	Used for storing questions, answers, and user profiles, along with real-time data interaction.
Trello	Project management	Used to track tasks, assign responsibilities, and monitor development progress across the team.
OpenAI API	AI interview simulation	Used to provide real-time interview simulation capabilities and analyze the results with sentiment analysis for feedback
flutter_tts	Translating text into speech	Integrated for translating chatGPT text into speech
speech_to_text	Translating speech into text	Integrated for natural language processing for the interview simulation

Table 12. Third-Party Components

Task	Responsible Member(s)
Selection of AI libraries and tools along with their integration	Arslan & Ansar
Design features and front-end of the application	Arslan & Sergey
Implement the cloud infrastructure for secure data storage and AI processing	Arslan & Sergey
Mock interview system development	Ansar & Nurzhigit
Implementation of text-to-speech and speech-to-text functions	Sergey & Ansar
Personalized reports and feedback generation for the user	Nurzhigit & Arslan
Voice recognition function development	Sergey & Nurzhigit
Design of user manuals	Arslan & Ansar
Evaluation and Feedback Collection	Arslan & Nurzhigit
Prepare user manuals and system documentation	Sergey & Nurzhigit
Testing and debugging the whole application	All members
Documentation of the project	All members

Table 13. Roles

Other features

Design:

- Developed and implemented progress tracking. There is a graph that displays any user's progress over time that can be viewed (refer to Figures 42 and 43)
- Designed and developed a chat window interface. It features an AI assistant with capabilities to simulate a real interview (refer to Figures 28-30, 31-33, 35-37).
- Created a minimalistic, responsive User Interface (refer to Figures 17-25).

AI Integration:

- Integrated AI libraries for natural language processing. It understands and generates human-like responses during interviews, be it training or real (refer to Figure 30).
- Implemented voice recognition functionalities (refer to Figures 31-33).

- Configured the system to give a sentiment analysis based feedback upon the completion of the interview (refer to Figures 38-39).

Project Execution (15%)

Fall'24 Semester

1. Project Initiation

- a. Define the project goals, deliverables and milestones.
- b. Assign roles to team members.

2. Requirements Gathering and Analysis

- a. Determine detailed requirements for the app.
- b. Analyze the requirements for AI integration.

3. System Design

- a. Design the architecture for the app, including frontend, backend, and AI integration.
- b. Implement the cloud infrastructure for secure data storage and AI processing.

4. Frontend Development

- a. Develop the UI.
- b. Implement features like mock interview sessions, speech-to-text, text-to-speech, and difficulty level selection.
- c. Design progress tracking and feedback display pages.

5. Project Management and Coordination

- a. Use project management tools (Trello, Gantt diagrams) to track progress updates and milestone monitoring.
- b. Ensure effective communication and collaboration (Telegram, Zoom).

6. Backend Development

- a. Set up the backend for user authentication and data management.
- b. Implement database systems.

In the **Fall** semester, we focused on defining the scope and setting clear milestones. This included planning the implementation of functionalities. Our first goal was the agreement on how to divide responsibilities equally and define the ways of collaborative teamwork. The responsibilities inside the group were divided by Arslan, who was our Team Lead. He made sure every member contributed equally to the total work done and based on each member's personal desires and capabilities. We created a chat in Telegram messenger to keep each other updated about new milestones and assignments, make notes and plan the team meetings ahead. We had been meeting the team's Adviser every week or two on the communication platform Zoom to keep track of the progress on the project and discuss our next steps. Gantt diagrams and Trello application were used to keep track of milestones and indication of responsibilities. Examples are provided below.

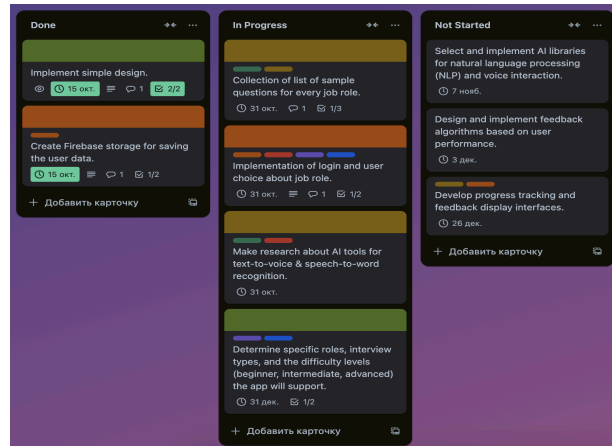
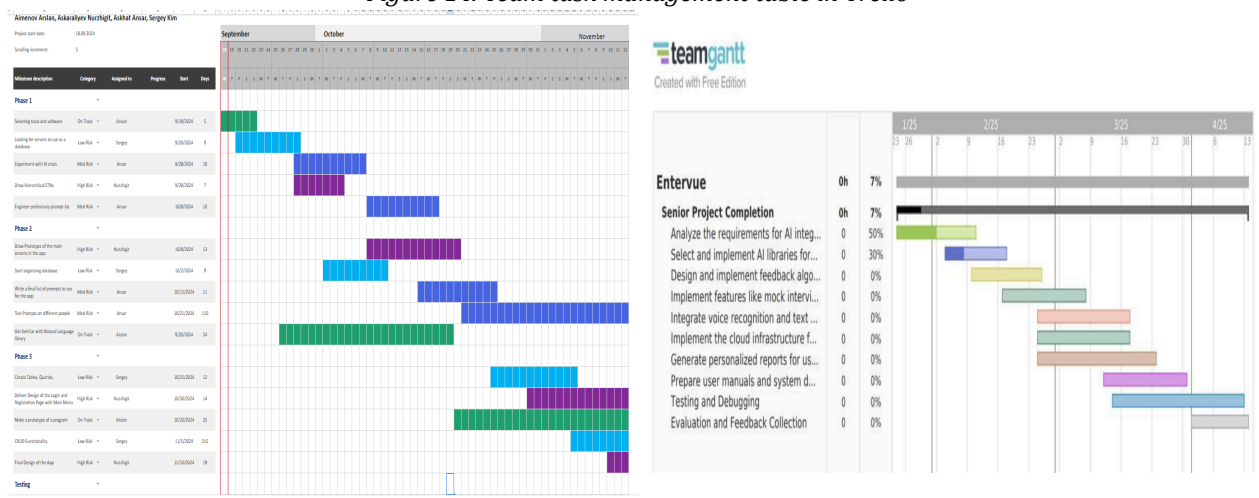
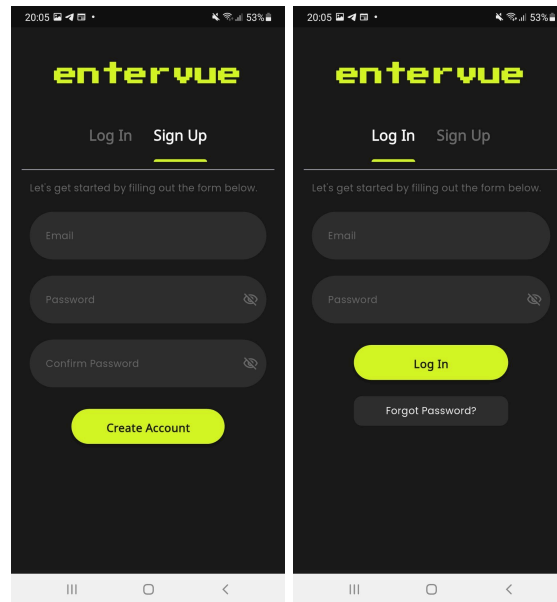


Figure 14. Team task management table in Trello



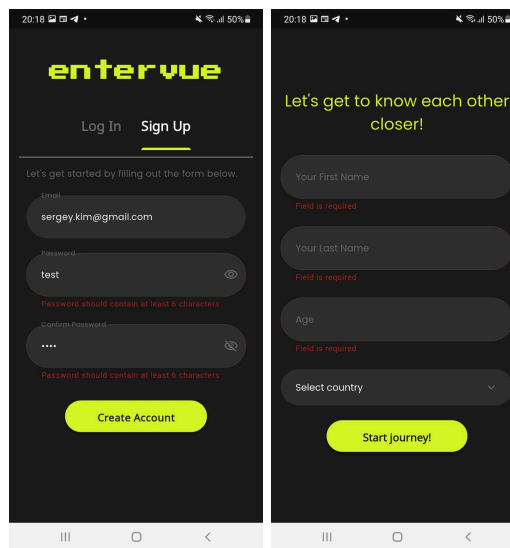
Figures 15 & 16. Project timelines and deadlines using Gantt diagrams

After reviewing several design alternatives, we opted for a modular approach that allowed us to increase User Experience. For instance, we made a choice of font and colors (Accessibility Design to support users with visual impairments or other disabilities), how many pages the application should consist of and what function will every page be responsible for. This helped our team to design the front-end of an application - user-friendly intuitively understood interface for easy navigation, access to app resources and logical flow between pages. After making design choices, the authentication was implemented so the user would be able to register and login with an email entering their name, age and country.



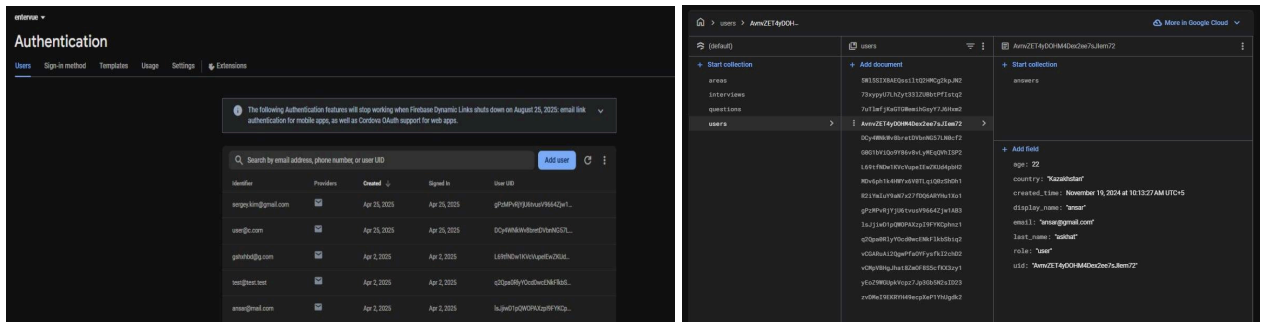
Figures 17 & 18. Sign Up and Login pages.

During signing up into our application, the user needs to pass the validation process. This means the email should be valid, password should contain at least 6 characters and should be entered twice to confirm, as well as the labels where user enters name, age and country cannot be empty. Otherwise, the user simply can't register.



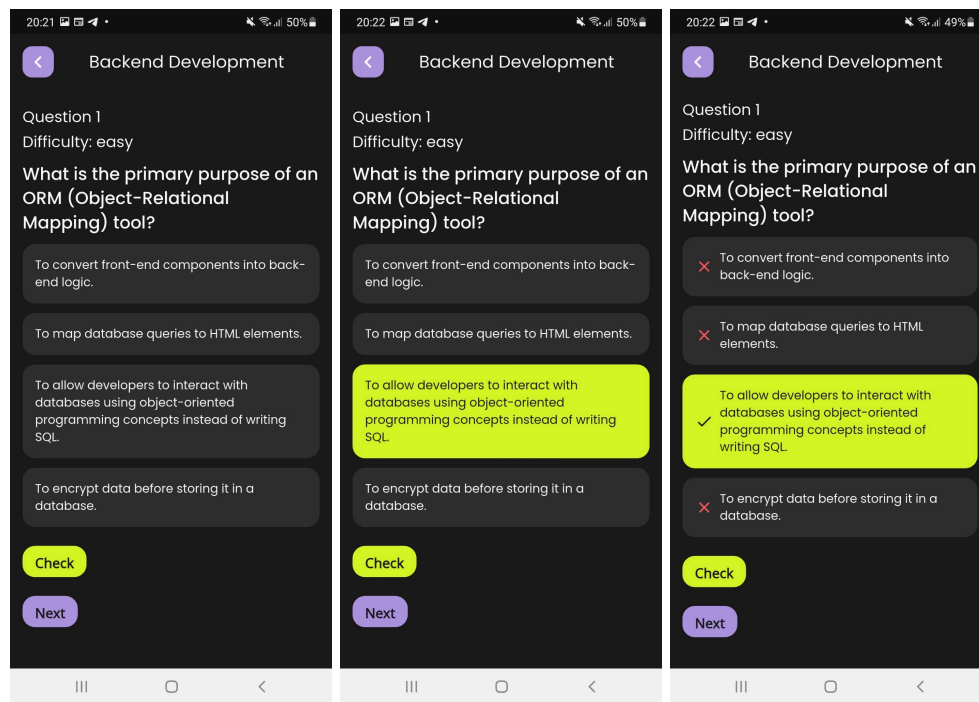
Figures 19 & 20. Sign Up pages.

This led us to set the database management system to efficiently and securely store User Data. We collectively decided to use Firebase for its easier integration to the tools we've been using compared to the other database management systems. Moreover, unlike traditional SQL databases, Firebase's NoSQL structure allowed us to store unstructured or semi-structured data without rigid schema definitions.



Figures 21 & 22. User Data storage in Firebase.

The team also created a database of multiple choice questions with correct answers that cover a wide variety of questions without being repetitive for every job role in our application. These diverse questions were to provide users a chance to see his/her own weaknesses and areas to improve. We made sure the questions were not repetitive, meaning for every session there will be new questions for the specific user and specific job role, and the user will not get the questions that he/she already answered in the previous sessions. Moreover, we provided the user with the opportunity to check the correct answer of the question instantly.



Figures 23, 24 & 25. Example of multiple choice quiz question.

By the end of the Fall semester, the application had a clear design with logical flow between pages, user login and sign up, quiz with multiple choice questions and saved results on the profile page of the user.

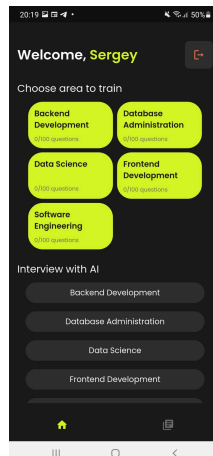


Figure 26. The main page of the application.

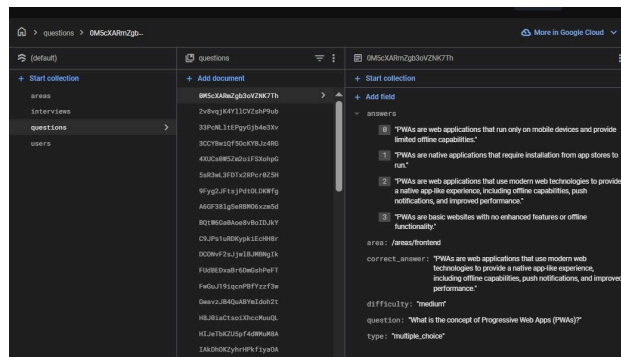


Figure 27. Document of question in Firebase.

Spring'25 Semester:

1. AI Integration

- Select and implement AI libraries/frameworks for natural language processing (NLP) and voice interaction.
- Integrate voice recognition and text generation using third-party services.

2. Feedback Generation Functionality

- A system to monitor user performance, providing feedback on strengths and areas for improvement.

3. Progress Tracking Functionality

- A simple progress tracking system to monitor user performance and improvements over time.

4. Testing and Debugging

- Test individual components.
- Implement functional testing.

5. Evaluation and Feedback Collection

- Collect feedback from experienced interviewers and job candidates.
- Iterate on the design based on user testing and expert evaluations.

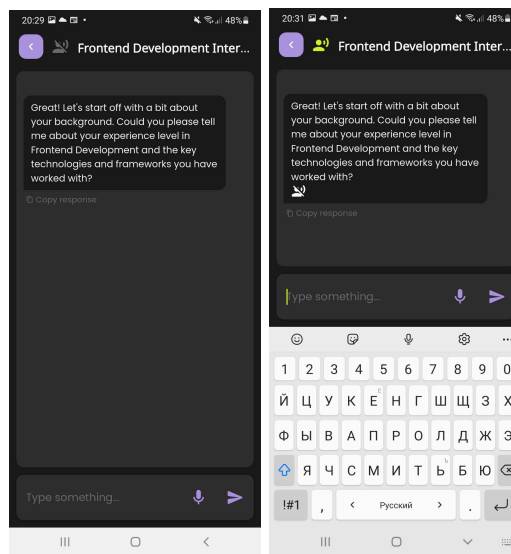
6. Documentation and Training

- a. Prepare user manuals and system documentation for users to understand how to use the app.
- b. Identify ideas for future improvements.

In the **Spring** semester, we continued the development of the application by implementing real-time interview chat with AI-assistant including score and feedback section after each interview and faced challenges. First of all, there was a choice to make which Large Language Model to use for the AI assistance(Google's Gemini or Openai's GPT). We tested which of them provided clearer, more understandable questions including the difficulty level of them, as well as the analysis of the quality of answers users give. Considering these factors, the team's decision was to use GPT because of its mature ecosystem and developer support, API stability and flexibility and integration compatibility. GPT APIs integrated more smoothly with the platforms and frameworks we were using, reducing development overhead and compatibility issues. Moreover, our team had previous experience working with GPT in past academic projects or internships. This existing familiarity accelerated the development process.

After that, one of the problems we encountered was prompt generation - we improved the prompt generation because we needed to describe and explain missing details each time, otherwise the interview sessions with the AI assistant did not behave correctly.

Next milestone of our project was to enable voice messages during the chat, which led us to implement speech-to-text and text-to-speech functionalities(speech_to_text and flutter_tts packages were used for each functionality, respectively). Speech-to-text enabled the user to send voice messages to the chat and an AI-assistant converted it to the text to understand and give its response. Speech-to-text was easier to implement compared to the text-to-speech. The problem with the text-to-speech implementation was the fact that it did not stop until it voices the whole text, which led to another challenge with real-time feedback functionality. After the user finished the interview, the AI-assistant gave feedback and voiced the whole feedback text without the opportunity for the user to stop it in case he/she does not want to listen to the generated feedback. This problem was solved with changing the version of the flutter_tts package, where more settings were available and we wrote the logic so users could stop or even disable voicing the feedback.



Figures 28 & 29. Chat with an AI assistant.

As shown in Figures 14 & 15 above, the difference between two screenshots is in the icon on the top left corner. The yellow color of an icon indicates that the AI assistant will voice its every message it sends, while the grey color means the user will get only text messages.

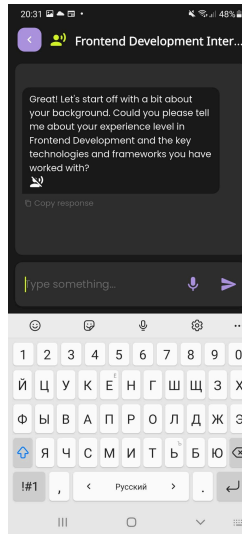


Figure 30. AI is voicing the message.

We added an icon after the text message which pops up only when AI is speaking the text. Pressing this button disables this functionality and AI stops speaking, sending only text messages.

Speech-to-text functionality was used to implement voice messages from the user, to increase the accessibility of an application. The user can press the 'microphone' icon and start speaking, and press the "Stop" button to stop the microphone. After that, the application converts the voice to the text and it will appear on the placeholder of the label where the user enters the text.

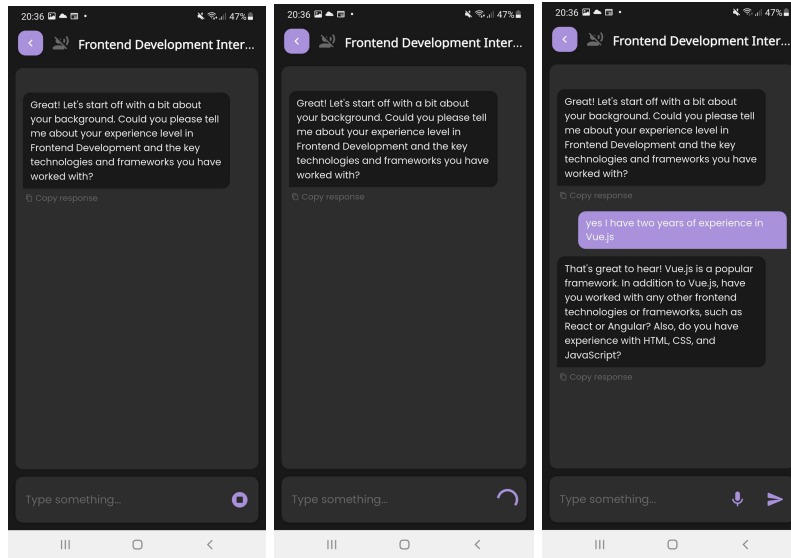


Figure 31, 32 & 33. Voice message of the user.

Another feature of our application is that there are two modes of interviews: Training and Real. The difference is that during Training interview AI-assistant gives instant feedback after each question and indicates whether the answer of the user was correct or not. In contrast, during Real interview AI-assistant does not give feedback after each question, behaving like a real interviewer, adapting to the answers of the user.

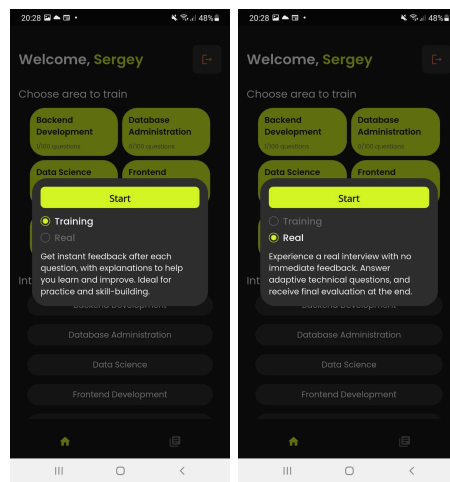


Figure 34. Types of interviews

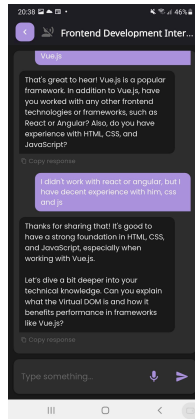
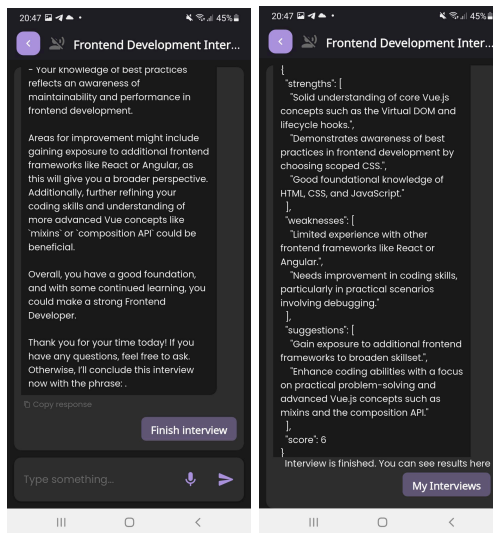


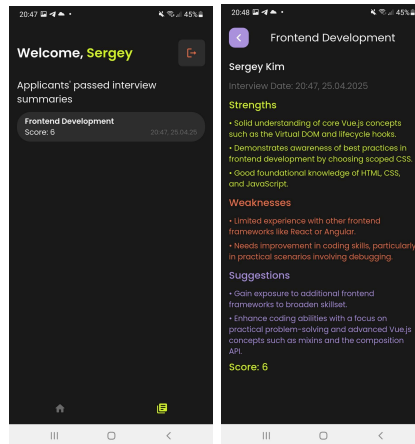
Figure 35. Real-time interview session.

If the user wanted to finish the interview himself/herself, he/she should type “Finish Interview” in the chat. Once an AI-assistant receives such a message, it will stop the interview and give the real-time feedback to the user, including score. And a new button pops up in the chat with text “My Interviews”, pressing which leads to the profile page of the user.



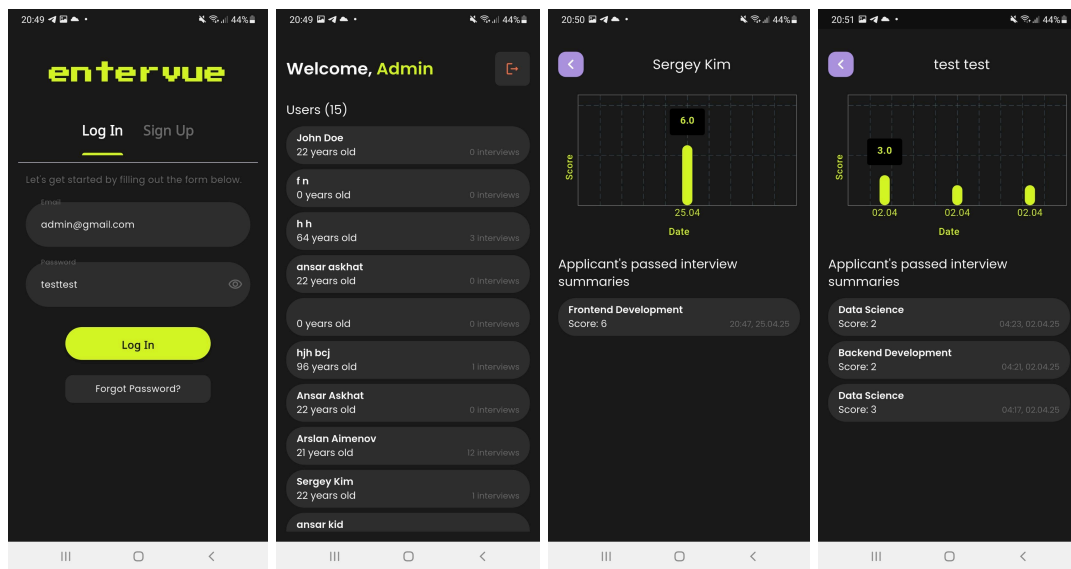
Figures 36 & 37. Finishing the interview.

The profile page of the user shows the list of all interviews of the user including the time and date when it was conducted and score. Pressing to this list item opens the new page with the full feedback (strengths, weaknesses and suggestions).



Figures 38 & 39. Feedback on Profile page.

When everything was more or less ready, the team implemented a special interface for the admin user, where he/she can see the list of all users, their personal info and interview sessions. The admin user can also see the progress of the user in the form of a graph diagram where the x-axis is the date of the conducted interview session and y-axis is the score for that particular session.



Figures 40, 41, 42 & 43. Interface for the admin user.

Evaluation (20%)

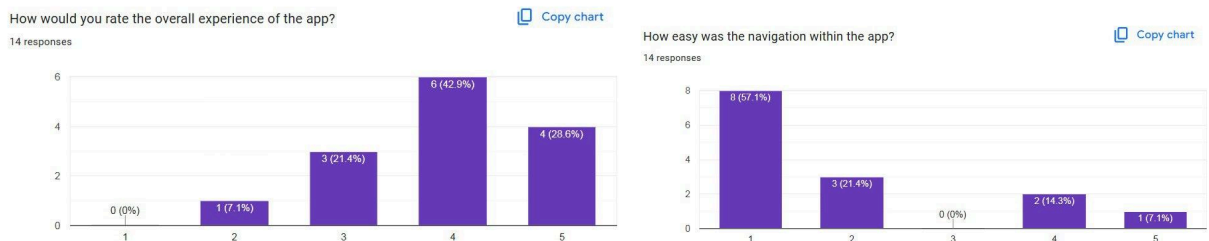
After the majority of application functionality was completed and checked by team members during an alpha testing, we corrected major and minor bugs we encountered. Among the most prominent ones are adding a “mute” button to the interview, if the user finds AI speaking distracting,

Moreover, we conducted a Google Form questionnaire among volunteers who agreed to test our application during the development process. There were 14 people total who helped us with their feedback and evaluation of the application.



Figures 44 & 45. 1st and 2nd questions of the questionnaire.

As shown above, these 14 people are not only experienced people in interviews (intermediate and advanced), but also people who have never done a technical interview before (beginners).



Figures 46 & 47. 3rd and 4th questions of the questionnaire.

Next, two questions were asked about general difficulty of use of our application and experience and most responses were positive. 10 out of 14 people rated their experience with the score 4 and higher and 11 people found it easy to navigate inside of an application.



Figures 48, 49 & 50. 5th, 6th & 7th questions of the questionnaire.

After that we asked the users about the bugs they faced during the testing of the application and even if more than 60% of users stated they did not face any bug, some of them did which helped us a lot to improve the performance and experience of using our application. People who faced bugs mostly stated that there

was a problem with voice recognition during voice messages and it led us to fix bugs about speech-to-text functionality.

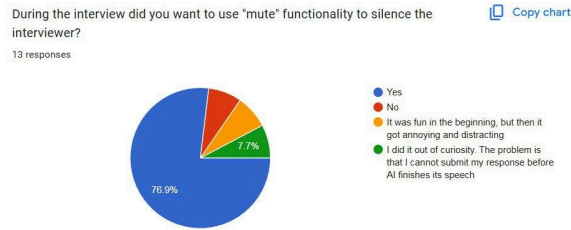


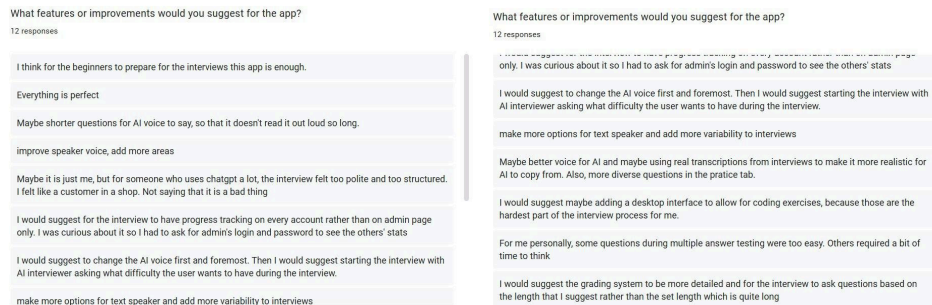
Figure 51. 8th question of the questionnaire.

More than half of people used “mute” functionality to turn off the AI-assistant’s voice and some of the testers said it was annoying and distracting. If there was no such button, our application would lose its User Experience and that AI-assistant’s voice would simply distract the user.



Figures 52 & 53. 9th & 10th questions of the questionnaire.

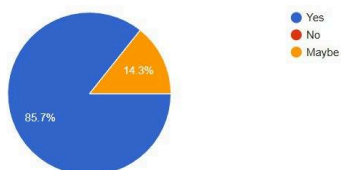
Next, we asked the opinions of the users about the provided feedback after the interview. Almost all users stated that the feedback generated by AI-assistant was clear and understandable and it helped to understand user’s strong and weak sides.



Figures 54 & 55. 11th & 12th questions of the questionnaire.

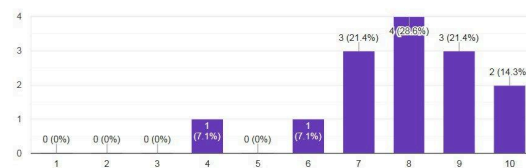
Moreover, we asked about suggestions about improving our application. Some users stated they would like us to add more job positions, which gave us great motivation to expand the scope of our app. Also, one of the users stated questions of the multiple choice quiz were too easy which made us add up to 100 questions for each job position. Many suggestions were about AI-assistant’s voice which was fixed as explained above.

Would you recommend this app to your friend who is preparing for interviews?
14 responses



Copy chart

Overall, how would you rate this app?
14 responses



Figures 56 & 57. 13th & 14th questions of the questionnaire.

Lastly, we asked the users if they would recommend our application to the people preparing for them - and none of the answers was negative and 12 out of 14 ratings were higher than 6 out of 10 which made us sure our application would help a lot of people preparing for the technical interviews and leave positive experience to the users and we can state confidently that we achieved our goal that we set at the beginning of the project.

Conclusion and Possible Future Work (5%)

The interview preparation application that supports the development of both technical and soft skills was successfully developed. Key contributions include integration of real customizable question banks along with manual difficulty selection for each question, real-time AI feedback and progress tracking sections. Entervue is an application that provides a perfect environment for job seekers to enhance their interview taking skills in a realistic fashion.

It was identified that the app's core functionalities operate as intended and the app has a lot of potential for expansion. The following are the areas of improvement that we found. The current question bank for each job position has 100 questions. For a total of 500 questions, which sounds like a big number. However, in the long-term it could prove to be too little. Additionally, the AI grading and interview conduct could be further enhanced, where a definitive grading formula could be introduced for the feedback. For the interview conduct, the AI could choose the difficulty of questions during interview training based on the respondent's answer in a clearer way with a behavior formula.

To conclude, the future plan is to increase the question diversity, enhance the AI behavior and functions, and conduct further testing on these enhancements before the public release of the app. The application in its current state has potential that can help a lot of interview takers to practice in a semi-realistic environment.

References (5%)

- Jebreen, I. (2021, September). An analysis of job advertisements for software engineering employability skills and knowledge: Jordan as a case study. https://www.researchgate.net/publication/354687495_An_analysis_of_job_advertisements_for_software_engineering_employability_skills_and_knowledge_Jordan_as_a_case_study
- Anton, R. (n.d.). Interview and Job Search Tips for Software Engineers. https://www.ranton.org/g/software_eng_job_searches

- Marinovici, C. (2014, March 10). The Hidden Job Requirements for a Software Engineer. https://www.researchgate.net/publication/261960733_The_Hidden_Job_Requirements_for_a_Software_Engineer
- Rusu, A. (2011, August 25). Simulating the software engineering interview process using a decision-based serious computer game. <https://ieeexplore.ieee.org/abstract/document/6000345>
- Leetcode "Top Interview Questions". <https://leetcode.com/explore/interview/card/top-interview-questions-easy/>
- Udemy "Ace the Coding Interview". <https://www.udemy.com/course/ace-the-coding-interview/?srsltid=AfmBOoqeEfBNX1wQFBz9AMhGkUL8nzTNBtBGr44XsuHui8rrXmjXrUfZ>
- Quizlet "Technical Interviews". <https://quizlet.com/763344917/technical-interviews-flash-cards/>