PROOF OF WORK

Course: AI Application Development

Instructor: Dr. Nimrita Koul

Team Number: Team 11

Student Name: Riya K Janali

SRN: R22EN065

Project Title: AI Interviewer App

Date of Submission: May 15, 2025

**1. Project Overview**

The AI Interviewer App is a smart platform designed to help candidates practice job interview questions. It provides personalized, real-time feedback, making the preparation process more effective and engaging. Voice-driven features are optional, allowing flexibility based on user preferences.

**Week 1 (Apr 16 2025)**

* Initiated AI Interviewer App project planning phase
* Conducted initial team meeting to assign responsibilities
* Completed preliminary research on project requirements
* Set up project management tools and documentation structure
* Created initial project timeline and milestones

**Week 2 (April 21, 2025)**

* **Proposed Solution Architecture**
* Web-based UI for selecting interview types and durations (Streamlit)
* **Groq API:** Advanced question generation and evaluation
* converted into speech using gTTS
* User response recording and transcription using sounddevice, Whisper, and audio preprocessing with scipy
* Answer analysis with LLaMA to provide feedback and improvement suggestions

**Week 3 (Apr 28, 2025)**

**Implementation Guide: Correction of rendering of html code**

After reviewing the code and the issue description, I've identified that div class in code appears as part of the reward display in the interview results page for users who receive the "Better luck next time!" badge (the lowest score tier).

**Issue Analysis**

**A screenshot of a computer

AI-generated content may be incorrect.**

The code is being displayed through the get\_reward\_badge() function when a user receives a score below 50%. This function returns a tuple containing:

1. The code (in this case,confetti part)
2. The badge title ("Better luck next time!")
3. A description message

The code is then displayed in the reward card on the results page through this HTML formatting:

A screenshot of a computer

AI-generated content may be incorrect.

**Solution**

To remove the code and its container from the "Interview Results" section, I've modified the get\_reward\_badge() function to return an empty string for the badge icon in the lowest tier case, while preserving the badge title and description.

**Modified Code**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Implementation Notes**

1. The change is minimal and focused on just removing the code snippet itself
2. The "Better luck next time!" text and description are preserved
3. When the badge icon is an empty string, the <span class="reward-badge"> element will still exist but will be effectively empty
4. The change does not affect other functionality of the results page

**Testing Recommendations**

**A screenshot of a computer

AI-generated content may be incorrect.**

To verify this change:

1. Complete an interview with deliberately low scores (below 50%)
2. Confirm the results page displays "Better luck next time!" without the code part.
3. Verify that other score tiers still display their respective icons correctly

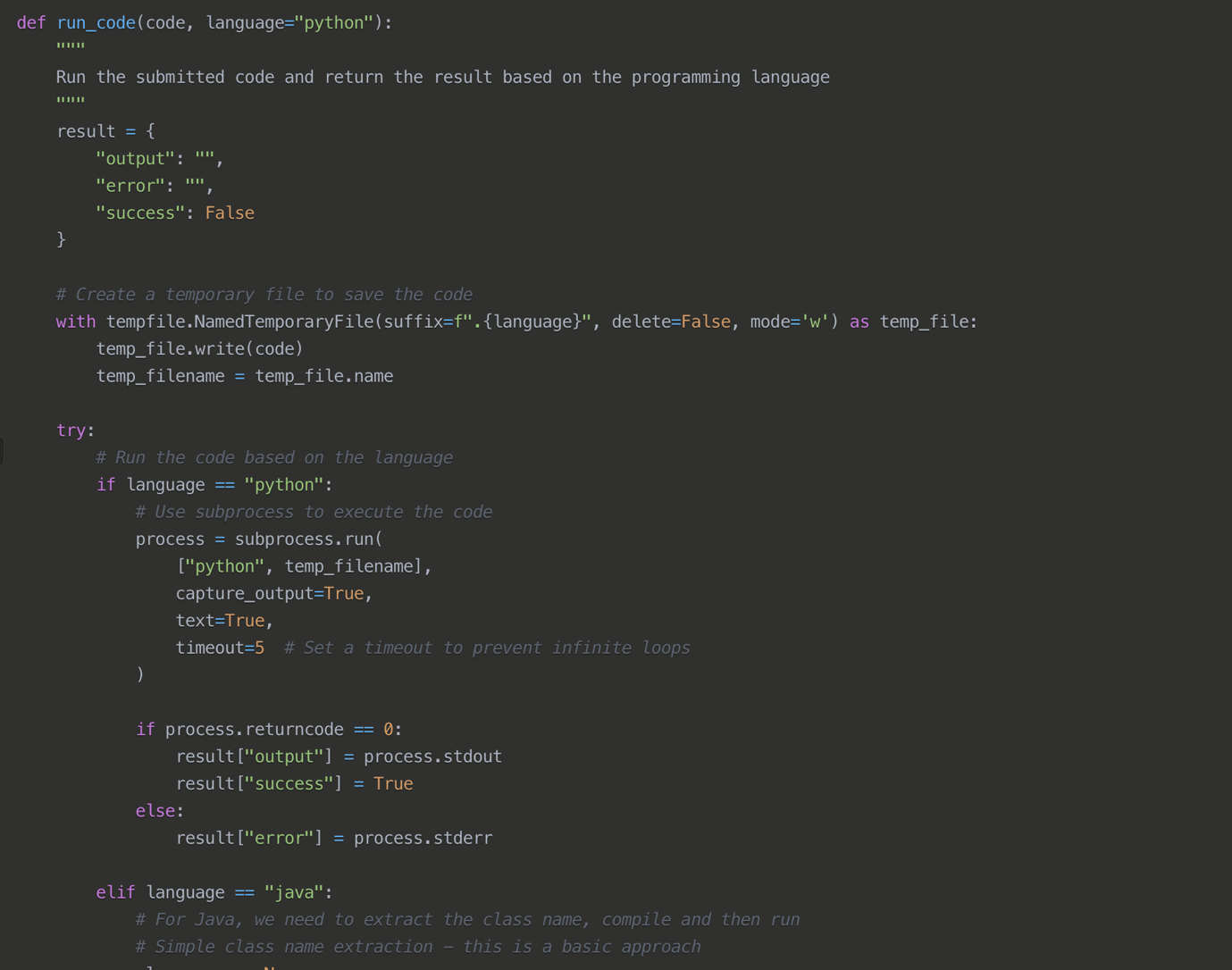
This change ensures the div part no longer appears below the "Interview Results" heading while maintaining the rest of the application's functionality

**Week 4( May 5, 2025)**

**Implementation Guide: Domain specific code runner**

**1. run\_code()**

* Executes code in multiple languages (Python, Java, JavaScript, C++, HTML, CSS).
* Handles compilation, runtime errors, and timeouts.
* Cleans up temporary files after execution.

****

**2. Domain-to-Language Mapping (get\_language\_from\_domain)**

**Core Purpose:**

* Maps a given interview domain to the most appropriate programming language.

**Key Mappings:**

* **Python:** Data Science, Machine Learning, DevOps, Algorithms.
* **Java:** Traditional Object-Oriented Programming (OOP) interviews.
* **JavaScript:** Full Stack, Node.js, React.
* **C++:** High-performance and system-level programming.
* **SQL:** Database questions (placeholder for future integration).
* **Pseudocode:** System design (not executable, for architectural discussions).

**A screen shot of a computer program

AI-generated content may be incorrect.**

**3. Question State Management (next\_question)**

**Core Purpose:**

* Controls the interview flow by generating and resetting the current question based on the domain and question type (coding or non-coding).

**Key Features:**

* **DomainSpecific Question Generation:** Integrates with external AI models for domain-appropriate question prompts.
* **State Reset:** Resets key session variables like spoken, answered, score, and feedback to maintain a clean interview flow.

A computer screen shot of a program

AI-generated content may be incorrect.

**Week 5 (May 6 , 2025)**

**Implementation Guide: Non-Coding Question Generator**

**A screenshot of a computer

AI-generated content may be incorrect.**

**What's Changed**

The original function was generating technical questions with no restrictions, which could include coding challenges. The modified version now:

1. **Explicitly prevents coding questions** by instructing the LLM not to ask candidates to write code
2. **Refocuses the questions** toward conceptual understanding, theory, and design principles
3. **Maintains relevance to the selected domain** while ensuring questions test knowledge rather than coding ability

A screen shot of a computer

AI-generated content may be incorrect.

**How to Implement**

1. Locate the current generate\_question() function in your codebase (around line 370-385)
2. Replace it with the modified version provided
3. No other changes are needed, as all function calls, parameters, and return values remain the same

**Benefits**

* **More accessible interviews**: Not everyone is comfortable coding on the spot
* **Focus on knowledge**: Tests deeper understanding rather than syntax recall
* **Broader question range**: Encourages questions about architecture, design patterns, and theory
* **Reduces anxiety**: Candidates won't feel pressured to produce working code during the interview

**Examples of Generated Questions**

Instead of coding questions like:

* "Write a Python function to implement a binary search tree"
* "Implement a quicksort algorithm from scratch"

The modified function will generate conceptual questions like:

* "Compare and contrast virtual memory and physical memory. How does the operating system manage the translation between them?"
* "Explain the differences between REST and GraphQL APIs, and describe scenarios where each would be more appropriate."
* "Describe the CAP theorem and its implications when designing distributed database systems."

**Testing**

To verify the function is working correctly, you can temporarily add a test section to your application that calls generate\_question() with various domains and prints the results to ensure they follow the non-coding guidelines.

**Week 6(May 12, 2025)**

**Implementation Guide: Read out loud feature**

Modidfied the functions needed to make the read-aloud feature optional for both regular questions and coding questions. The key modifications will be to the *speak\_question()* function and the integration of a microphone button for user control.

1. **Added new state variables** to track TTS status:

* tts\_enabled: Flag to control whether TTS is active or not
* tts\_speaking: Flag to track if TTS is currently speaking
* tts\_thread: Thread for TTS processing

A screen shot of a computer program

AI-generated content may be incorrect.

1. **Created new control functions**:

* toggle\_tts(): Toggles the TTS feature on/off
* stop\_tts(): Stops any ongoing speech playback
* speak\_text(): A wrapper function that only speaks if TTS is enabled

A screenshot of a computer program

AI-generated content may be incorrect.

1. **Updated the speak\_question() function**:

* Now only speaks if TTS is enabled and not already spoken
* Uses the new speak\_text() function for thread handling

A computer screen shot of text

AI-generated content may be incorrect.

1. **Created a new display\_question() function**:

* Displays the question text with TTS controls
* Handles both regular and coding questions
* Resets the spoken flag when a new question is displayed

A computer screen shot of a computer code

AI-generated content may be incorrect.