Test Task for Swift Developer(MacOS) position

Objective

To evaluate the candidate's ability to develop a basic prototype for a video conferencing application on MacOS, demonstrating their knowledge of external device integration and video data processing using Swift/Objective-C.

Task Overview

You are tasked with building a prototype application that implements key features of a video conferencing system, focusing on external device connectivity and video data processing.

Requirements:

1. **Device Integration:**
    - Detect and list all connected video devices (e.g., integrated or external webcams).
    - Provide functionality for users to select specific devices for the video conferencing session.
    - Include a feature to check the connection status of the selected devices and notify the user if a device is disconnected.
2. **Video Preview:**
    - Capture live video from the selected webcam and display it in a video preview window.
    - Implement basic video processing, such as converting the video feed to grayscale or adding a simple filter.
3. **User Interface:**
    - Create a simple and intuitive UI that includes:
        - Dropdown menus to list and select video devices.
        - Buttons to start/stop video preview.
        - A status indicator for each device to display its current state (connected/disconnected).

Time Limit

You have 6 hours to complete the task. Please aim to make it as thorough and as close to the description as possible. Focus on the result application performance, simplicity, and meeting the time constraint.

Bonus

- Implement a feature to record the video and audio streams and save them as separate files.
- Use platform-specific libraries to handle video and audio processing efficiently (AVFoundation).

Submission Instructions

To share the results of your work, please include the following:

1. **Source Code:**
    - Provide all source code files in a structured format.
    - Ensure the code is easy to understand.
2. **Executable File(Optional):**
    - Share a compiled executable file for the platform(s) you developed the application on (MacOS).
    - Ensure the executable can be run without additional configuration or dependencies.
3. **README File:**
    - Include a README file with the following details:
        - Instructions on how to run the application, including any prerequisites or dependencies.
        - An explanation of your approach and any challenges you encountered.
4. **Demonstration(Optional):**
    - If possible, include a short video demonstrating the application in action (e.g., showing the device integration, a video preview, etc.).
5. **Submission Method:**
    - Upload all required files to a GitHub repository or a cloud storage service (e.g., Google Drive, Dropbox, or OneDrive).
    - Share the repository link or download link in your response.

Good luck, and we look forward to seeing your creativity in action!