# Practical Test Interview Question for MAUI Developer

**Scenario:**
You are tasked with migrating a Xamarin.Forms WebView application to .NET MAUI. This involves not only moving the codebase but also ensuring that the project follows best practices for architecture, folder structure, and platform-specific configurations.

Your task is to take the given Xamarin.Forms project and begin the migration process, focusing on setting up the MAUI project with proper architecture and folder structure while handling platform-specific configurations as well.

---

## Steps for the Candidate:

1. **Setup the Project:**
   - Create a new .NET MAUI project to replace the existing Xamarin.Forms project.
     Please refer to this link to clone the project
     https://github.com/progresitydaz/MAUI_Migration

   - Transfer any reusable code or assets from the Xamarin.Forms project to the appropriate locations in the MAUI project.
2. **Define Folder Structure:**
   - Maintain a clean and modular folder structure. For example:
     - **Views**: Contains all UI elements.
     - **ViewModels**: For MVVM pattern.
     - **Services**: For business logic or data fetching.
     - **Resources**: For styles, fonts, and images.
     - **Platforms**: For platform-specific code (e.g., handling WebView configurations).
3. **Demonstrate Platform-Specific Customizations:**
   - Configure project for both Android and iOS to support platform-specific behaviors via Handlers.
4. **Write a README:**
   - Document the steps you followed, explaining your architecture decisions, folder structure, and platform-specific configurations.

---

## Submission Requirements:

- A runnable .NET MAUI project showcasing the migrated WebView functionality.
- A well-organized folder structure aligned with MAUI best practices.
- Proper implementation of platform-specific configurations for Android and iOS.
- A README file describing the migration process and rationale for architectural choices.

---

**Evaluation Criteria:**

1. **Sound Initial Approach:**
   - Is the folder structure modular and clean?
   - Are architectural patterns like MVVM properly implemented?
2. **Platform-Specific Configuration:**
   - Is the WebView configured correctly for Android and iOS?
   - Are platform-specific customizations implemented cleanly within the `Platforms` folder?
3. **Maintainability:**
   - Is the codebase easy to understand and maintain?
   - Are configurations and dependencies well-documented?

---