## "Preparing for the Promuex Inc. Global Professional Certificate: Essential Knowledge and Skills Checklist"

**Overview:** The Promuex Inc. (Canada) Global Professional Certificate recognizes expertise across specialized fields like AI, cybersecurity, healthcare, and finance. To excel, you’ll need foundational skills, knowledge of industry tools, and practical experience. Here’s what to focus on before certification:

Instruction plan : Certified Mobile App Developer (CMAD)

### ****Course Overview****

The **Certified Mobile App Developer (CMAD)** course is designed for individuals aiming to develop and deploy mobile applications for both Android and iOS platforms. This course covers essential mobile development skills, from UI/UX design to back-end integration, focusing on popular frameworks like React Native, Flutter, and native development with Android Studio and Xcode. By the end of the course, students will have the skills to build, test, and deploy cross-platform and native mobile applications.

### ****Course Objectives****

By the end of this course, students will be able to:

1. Design and develop mobile app interfaces following UI/UX best practices.
2. Build mobile applications using React Native, Flutter, or native SDKs.
3. Implement user authentication and manage in-app data storage.
4. Integrate RESTful APIs for real-time data exchange.
5. Test and debug applications on both iOS and Android platforms.
6. Optimize app performance and manage device resources.
7. Deploy applications to the App Store and Google Play Store.

### ****Module Breakdown with STAR Examples****

#### ****Module 1: Introduction to Mobile App Development****

* **Objective**: Understand the fundamentals of mobile app development and explore the main tools and platforms.
* **Topics**:
	+ Overview of Mobile Development: Android vs. iOS
	+ Cross-Platform vs. Native Development
	+ Setting Up Development Environments (Android Studio, Xcode, VS Code)
* **Learning Activity**: Set up a “Hello World” app on both Android and iOS simulators.
* **Assignment**: Create a simple app that displays a welcome message and runs on both iOS and Android.

**STAR Example**:

* **Situation**: A company wants to launch its first mobile app and needs advice on whether to go cross-platform or native.
* **Task**: Develop a basic prototype to evaluate the performance and look on both platforms.
* **Action**: Build a “Hello World” app in both Android Studio and with React Native to compare the results.
* **Result**: Helped the company make an informed decision, choosing cross-platform for efficiency and native for feature-specific projects.

#### ****Module 2: UI/UX Design for Mobile Applications****

* **Objective**: Learn the principles of mobile UI/UX design to create intuitive, user-friendly interfaces.
* **Topics**:
	+ Designing with Figma or Sketch
	+ Mobile Design Patterns and Navigation Principles
	+ Accessibility and Responsive Layouts
* **Learning Activity**: Design the UI for a login screen with Figma, considering mobile UI principles.
* **Assignment**: Design and implement a multi-screen mobile app prototype for a simple e-commerce app.

**STAR Example**:

* **Situation**: A retail business needs a mobile app to improve customer engagement.
* **Task**: Design a user-friendly shopping interface that simplifies browsing and checkout.
* **Action**: Use Figma to create intuitive UI elements, ensuring a seamless experience from browsing to purchase.
* **Result**: Delivered an easy-to-navigate interface, leading to higher user satisfaction and more conversions.

#### ****Module 3: Cross-Platform Development with React Native and Flutter****

* **Objective**: Develop cross-platform applications using React Native or Flutter.
* **Topics**:
	+ Introduction to React Native and Flutter
	+ Components, State, and Navigation
	+ Handling User Input and Forms
* **Learning Activity**: Build a basic to-do list app in React Native or Flutter.
* **Assignment**: Create a cross-platform weather app that fetches and displays weather data based on user location.

**STAR Example**:

* **Situation**: A weather service wants an app that provides real-time weather updates on both iOS and Android.
* **Task**: Build a cross-platform weather app with live weather updates and location-based data.
* **Action**: Use React Native to build the app, implement location services, and connect to a weather API.
* **Result**: Delivered a user-friendly weather app that works seamlessly on both platforms, increasing user engagement and app installs.

#### ****Module 4: Native Android Development with Android Studio****

* **Objective**: Learn to develop native Android apps using Android Studio and the Android SDK.
* **Topics**:
	+ Activities, Intents, and UI Components
	+ Layouts (Constraint, Linear, Relative) and XML
	+ Handling Lifecycles and Resource Management
* **Learning Activity**: Create a simple notes app that uses Activities and stores notes locally.
* **Assignment**: Develop a recipe app with multiple screens and activity navigation, using Android Studio.

**STAR Example**:

* **Situation**: A cooking blog wants to offer its readers a mobile app with recipes and instructional videos.
* **Task**: Build a recipe app that stores recipes locally and includes multimedia.
* **Action**: Use Android Studio to create the app, managing different screens for recipes, categories, and videos.
* **Result**: Delivered a responsive and visually appealing app that allows users to browse recipes offline, increasing user engagement.

#### ****Module 5: Native iOS Development with Xcode and Swift****

* **Objective**: Develop native iOS applications using Xcode and Swift.
* **Topics**:
	+ Views, View Controllers, and Storyboards
	+ UIKit and SwiftUI Basics
	+ Managing User Input and Core Data
* **Learning Activity**: Create a simple app with SwiftUI that displays user information from a form.
* **Assignment**: Build an expense tracker app that saves data locally using Core Data.

**STAR Example**:

* **Situation**: A financial advisory firm wants an iOS app that helps clients track personal expenses.
* **Task**: Develop an app with a user-friendly interface to add and manage expenses.
* **Action**: Use Xcode and Swift to build the app, implementing Core Data for local storage.
* **Result**: Provided a reliable expense tracker, helping clients manage finances more effectively with a smooth iOS experience.

#### ****Module 6: Backend Integration and RESTful APIs****

* **Objective**: Integrate mobile apps with back-end services for real-time data handling and user interaction.
* **Topics**:
	+ Making API Calls with Axios (React Native) or Retrofit (Android)
	+ Parsing JSON Data and Updating UI
	+ Authentication and Authorization (OAuth, JWT)
* **Learning Activity**: Connect an app to a sample REST API and display data dynamically.
* **Assignment**: Build a blog app that fetches posts from an API, allowing users to read and post comments.

**STAR Example**:

* **Situation**: A news platform wants a mobile app that fetches the latest articles in real time.
* **Task**: Develop an app that connects to the platform’s API to display articles dynamically.
* **Action**: Use Axios to fetch data, manage the app state, and implement a refresh feature.
* **Result**: The app allowed users to access live updates, boosting engagement and retention rates.

#### ****Module 7: Testing and Debugging Mobile Applications****

* **Objective**: Test and debug mobile applications for a reliable, bug-free experience.
* **Topics**:
	+ Unit Testing and UI Testing with Jest, Espresso, and XCTest
	+ Debugging Tools in Android Studio and Xcode
	+ Device Testing and Simulator Testing
* **Learning Activity**: Write unit tests for core functionality in a mobile app.
* **Assignment**: Conduct end-to-end testing on an e-commerce app, identifying and fixing bugs before deployment.

**STAR Example**:

* **Situation**: A retail company needs a stable app for their launch event, free of any critical issues.
* **Task**: Test the app thoroughly to ensure a smooth user experience.
* **Action**: Use Espresso for UI testing and Jest for unit testing, debugging any issues identified.
* **Result**: Delivered a stable, high-performance app, achieving high user ratings and fewer support tickets.

#### ****Module 8: Performance Optimization and Resource Management****

* **Objective**: Improve app performance and efficiently manage device resources.
* **Topics**:
	+ Reducing App Load Time and Memory Usage
	+ Image and Resource Optimization
	+ Caching Strategies and Data Persistence
* **Learning Activity**: Optimize images and assets to reduce load time for an app.
* **Assignment**: Profile and optimize an existing app, reducing memory usage and increasing responsiveness.

**STAR Example**:

* **Situation**: A photo-sharing app is experiencing high memory usage, causing crashes on low-memory devices.
* **Task**: Optimize memory usage and resource handling.
* **Action**: Implement image caching and lazy loading for smoother performance on all devices.
* **Result**: Improved app stability and responsiveness, ensuring a seamless user experience even on older devices.

#### ****Module 9: Deploying Mobile Apps to App Stores****

* **Objective**: Prepare and deploy applications to the Google Play Store and Apple App Store.
* **Topics**:
	+ App Store Guidelines and Play Store Policies
	+ Signing, Versioning, and Uploading Builds
	+ Beta Testing and App Store Optimization (ASO)
* **Learning Activity**: Prepare a sample app for deployment to the Google Play Store.
* **Assignment**: Package and deploy a fully functional app to either the App Store or Play Store, optimizing the app listing.

**STAR Example**:

* **Situation**: A startup wants to launch its new productivity app and reach a large audience quickly.
* **Task**: Package the app, optimize the listing, and submit it for approval.
* **Action**: Follow App Store guidelines for packaging, sign the app, and submit with optimized screenshots and keywords.
* **Result**: Successfully launched the app on both stores, reaching a broad user base and gaining positive feedback.

### ****Conclusion****

The **Certified Mobile App Developer (CMAD)** course offers in-depth training for building, testing, and deploying mobile applications. Students gain hands-on experience with cross-platform and native development, back-end integration, testing, optimization, and app deployment. Through STAR examples and practical assignments, students develop the skills needed to create and launch high-quality mobile applications.