

Hackathon Assignment: Build an Open edX Mobile App for Ubuntu Touch/Lomiri

Title: "Open edX on the Go: A Mobile Learning App for Ubuntu Touch/Lomiri"

Objective

Build a mobile app for **Ubuntu Touch (Clickable)** or **Lomiri (Snap)** that enables users to browse, enroll in, and complete Open edX courses directly from their device. The app should provide a seamless learning experience, including video lectures, quizzes, and progress tracking, while addressing technical constraints and leveraging the most suitable tools.

Background

Open edX is a widely used open-source learning management system (LMS). Ubuntu Touch and Lomiri offer a privacy-focused mobile platform, but lack a dedicated Open edX app. Your task is to create a functional, user-friendly app that integrates with Open edX's REST API, with a focus on **QML/JavaScript** for the frontend and **Python** for backend logic, where feasible.

Requirements

1. Core Features

- **Course Browsing:** View available courses by category, popularity, or search.
 - **Enrollment:** Enroll in courses and track progress.
 - **Content Viewing:** Watch video lectures, read materials, and take quizzes.
 - **Offline Support:** Download course content for offline use.
 - **User Authentication:** Log in using Open edX credentials.
 - **Progress Sync:** Sync progress with the Open edX server.
-

2. Technical Stack

Frontend

- **QML/JavaScript:** Primary UI framework for Ubuntu Touch/Lomiri.
- **UI Toolkit:** Use **QQC2** or **Kirigami** for a native look and feel. Avoid C++ unless necessary.

Backend

- **Python:** For API interactions, but **avoid direct Python-QML integration** due to technical limitations.
- **Alternative Approach:** If Open edX provides a **JavaScript client API**, prioritize using it for direct communication between QML and the Open edX REST API. This avoids the need for a Python backend in QML.

- **Fallback Option:** If a Python backend is unavoidable, implement the **entire app in Python + QML**, but note that Python bindings for **LUITK** may not be available. In this case, rely on **QQC2** or **Kirigami** for UI components.

API

- Use the **Open edX REST API** for all data operations.

Packaging

- **Ubuntu Touch:** Package as a **Clickable** app.
 - **Lomiri:** Package as a **Snap**.
-

3. Deliverables

- A functional Open edX app for Ubuntu Touch or Lomiri.
 - Support for at least **3 Open edX features** (e.g., video playback, quizzes, discussions).
 - A **demo video** (max 2 minutes) showcasing the app.
 - **Documentation** (README) with:
 - Installation instructions.
 - API usage details.
 - Technical decisions (e.g., why JavaScript or Python was chosen for API calls).
-

Getting Started

Step 1: Set Up Your Environment

- Install the **Ubuntu Touch SDK** or **Lomiri development tools**.
- Choose a test Open edX instance (e.g., edX Demo).

Step 2: Design the UI

- Use **QML** to create responsive interfaces for:
 - Course listing.
 - Video player.
 - Quiz/assessment screens.
 - User profile and progress tracking.

Step 3: Integrate the Open edX API

- **Preferred:** Use **JavaScript** to fetch and display:
 - Course lists.
 - User enrollment status.
 - Video content (via YouTube or direct streaming).
 - Quiz questions and submissions.
- **Alternative:** If using Python, ensure it is **decoupled from QML** (e.g., via a local HTTP server or separate process).

Step 4: Implement Offline Mode

- Cache course content using **SQLite** or local storage.
- Allow users to download videos and materials for offline use.

Step 5: Package and Test

- **Ubuntu Touch:** Use **Clickable** to build and install the app.
 - **Lomiri:** Package as a **Snap** and test on a Lomiri device/emulator.
-

Bonus Challenges

- Add push notifications for course updates.
 - Implement dark mode for better readability.
 - Support multiple Open edX instances (e.g., self-hosted).
-

Resources

- [Open edX API Documentation](#)
 - [Ubuntu Touch App Development Guide](#)
 - [Lomiri/Snapcraft Guide](#)
-

Submission Guidelines

- Submit code in a **GitHub repository**.
 - Include a **README** with setup instructions and technical rationale.
 - Provide a **demo video**.
-

Why This Matters

This app will make open education more accessible on Ubuntu Touch/Lomiri, empowering learners to study anytime, anywhere—while respecting the platform’s technical constraints.

Key Adjustments:

- Emphasized **JavaScript** as the preferred method for API calls due to QML’s limitations with Python.
- Clarified the **technical trade-offs** between Python and JavaScript for backend logic.
- Removed assumptions about C++ and highlighted **QQC2/Kirigami** as the UI framework.