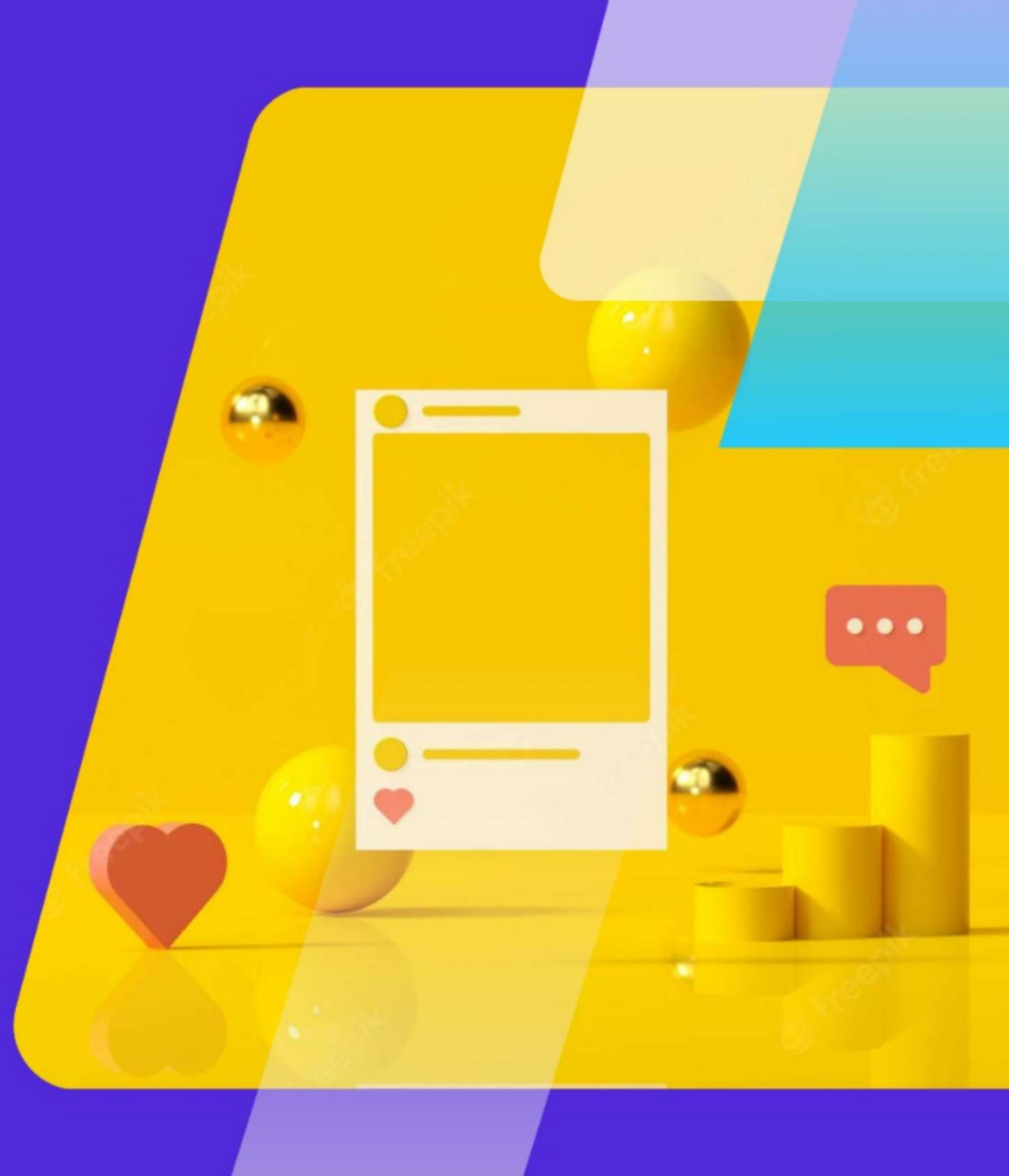# InternEducation App: Project Overview

A minimal runnable QML app demonstrating an educational workflow for interns
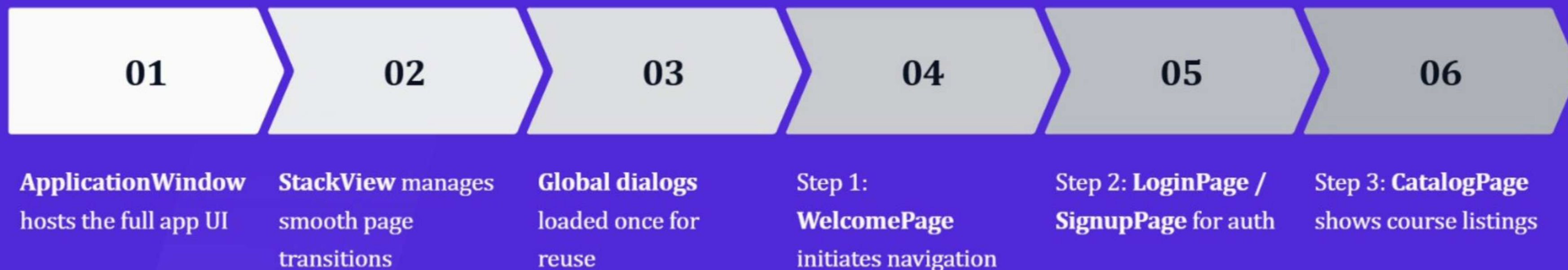
Team BOLD BYTES

Presenter

App Architecture and Navigation Workflow

Detailed stack-based navigation and modular UI components in the app

01

02

03

04

05

06

ApplicationWindow hosts the full app UI

StackView manages smooth page transitions

Global dialogs loaded once for reuse

Step 1: **WelcomePage** initiates navigation

Step 2: **LoginPage / SignupPage** for auth

Step 3: **CatalogPage** shows course listings

Welcome, Login, and Signup Pages

Overview of user access and onboarding screens with validation and navigation

Display WelcomePage with logo and tagline

The WelcomePage features the app logo and tagline 'Learn. Practice. Get certified.' along with navigation buttons for Login, Sign up, and Browse Catalog to guide users.

Implement LoginPage with input validation

LoginPage includes email and password fields with basic minimum length validation. On successful login, a token is saved, and the user is redirected to the CatalogPage. Invalid credentials trigger an error dialog.

Design SignupPage with minimal validation

SignupPage requires full name, email, and password inputs. It validates that all fields are completed. On success, it saves a token and navigates to CatalogPage; missing fields prompt an error dialog.

Enable secure user access and onboarding

These pages collectively provide a secure and smooth onboarding experience by managing user authentication and guiding users through the app entry process.

Course Catalog and Course Details

Overview of course listings and detailed course information dialogs for user selection

Component	Description	Key Features	Sample Data / Actions
CatalogPage	Lists courses using a ListView bound to a ListModel	ListView with course items, 'Open' and 'Details' buttons, Profile and Sync buttons	Courses: Introduction to Python (5 lessons), Web Development (7 lessons), Data Structures (6 lessons)
CourseDetailsDialog	Displays course title and description in a dialog	Triggered by 'Details' button on course items	Shows detailed course information to users

Course Content and Lesson Interaction

Structured learning flow with multimedia and quiz integration

01

CourseContentPage displays course details

Shows selected course title and description; lists lessons and quizzes as selectable items.

02

Selecting lesson opens LessonPage

Navigates from course overview to detailed lesson content screen for focused study.

03

LessonPage shows lesson title and content

Displays read-only lesson text with buttons for video playback, quiz, and back navigation.

04

VideoDialog enables in-app video playback

Supports multimedia engagement by playing educational videos without leaving the app.

05

"Take Quiz" navigates to QuizPage

Allows users to assess knowledge by taking quizzes following lessons.

Quiz Functionality and Scoring

Overview of quiz mechanics, scoring, and user navigation flow

01 Implement a simple 3-question multiple-choice quiz

QuizPage features three questions such as 'What is 2+2?', 'Which is a programming language?', and 'HTML stands for?'. Users select answers using RadioButtons to facilitate interaction.

02 Track quiz progress and manage navigation

The system tracks the current question index and score. Users advance through questions using the 'Next' button, and finalize the quiz with the 'Submit' button, which directs them to the ScorePage.

03 Display results and offer certification options

ScorePage shows the user's quiz score alongside total questions. It provides a 'Get Certificate' option that opens a CertificateDialog and includes a button to return to the CatalogPage for further navigation.

User Profile and Certificate Management

Enhancing user engagement through profile display and certificate handling features

01 Display user name and email on ProfilePage

ProfilePage shows demo data for user name and email, providing a clear user identity interface.

02 Implement buttons for certificate viewing and logout

ProfilePage includes buttons for viewing certificates, which opens CertificateDialog, and logout, which clears tokens and returns to WelcomePage.

03 Show certificate text with download option in CertificateDialog

CertificateDialog presents the certificate text clearly and offers a download option for credential management.

04 Simulate file download with progress bar in DownloadDialog

DownloadDialog visually simulates the file download process using a progress bar to enhance user experience.

05 Encourage user engagement and credential management

These integrated features promote active user interaction and effective management of digital certificates.

Backend Integration and API Overview

Key components and functions supporting data synchronization and persistence

Minimal Flask backend

01

Provides endpoints for health check (/api/ping) and course data (/api/courses) in JSON format.

Odoo API integration stub

02

Mock functions in utils/OdooAPI.js for fetching courses and sending sync payloads, designed for real API replacement.

SettingsManager

03

Manages token storage and clearing with stub implementations for secure authentication handling.

Data synchronization and persistence

04

Backend and API layer enable data syncing and persistent storage across systems.

Additional Features and Dialogs

Reusable dialogs and utility scripts enhance user experience and maintainability

01 Implement reusable dialogs for consistent user experience

Dialogs such as VideoDialog, CertificateDialog, CourseDetailsDialog, DownloadDialog, and SyncDialog are implemented once and reused to play videos, display certificates, show course details, simulate downloads, and sync with backend respectively.

02 Utilize utility scripts for backend simulation and management

Utility scripts like CertificateManager generate simple certificate text, while SettingsManager manages token persistence using demo stubs, supporting maintainability and demo functionality.

03 Enhance UX consistency and maintainability with reusable components

These reusable dialogs and utility scripts collectively improve the overall user experience consistency and simplify the app's maintenance process.

Thank You for Engaging with Our App

We appreciate your interest and look forward to supporting your learning journey

Express sincere appreciation for your attention

Thank you for taking the time to explore our app and its features, showing genuine interest and engagement throughout this presentation.

Encourage questions and feedback

We invite you to share any questions or feedback you have, helping us improve and tailor the app to better support your needs.

Commit to supporting your learning journey

Our team looks forward to assisting you as you continue to grow and achieve your goals with our app's innovative features.