

Interactive Learning Hub - Technical Part

Project Overview

Project Name: Interactive Learning Hub

Course: COMP5241 - Semester 1

Group: 17

Date: October 2025

Project Description

The Interactive Learning Hub is a comprehensive web-based platform designed to facilitate real-time interaction between teachers and students. The system enables educators to create engaging learning experiences through interactive activities, real-time feedback, and comprehensive course management.

Key Objectives

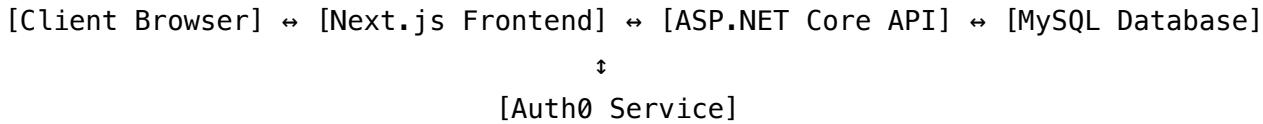
- **Interactive Learning Activities:** Real-time polls, quizzes, Q&A sessions, and brainstorming activities
- **Course Management:** Complete CRUD operations for courses and student enrollment
- **Real-time Feedback:** Live student engagement tracking and participation analytics
- **User-Friendly Interface:** Responsive design with accessibility compliance
- **Scalable Architecture:** Modular backend with clean separation of concerns

System Architecture

Architecture Pattern

- **Frontend:** Single Page Application (SPA) with Next.js
- **Backend:** RESTful API with [ASP.NET](#) Core
- **Database:** MySQL with Entity Framework Core ORM
- **Authentication:** JWT-based authentication with Auth0 integration

System Flow



🔧 Technologies Used

Frontend Technologies

Core Framework

- **Next.js 15.4.6** - React framework with App Router
- **React 19.1.1** - Frontend library
- **React DOM 19.1.1** - DOM rendering

UI & Styling

- **Material-UI (MUI) 7.3.1** - Component library
 - `@mui/material` - Core components
- **Emotion** - CSS-in-JS styling
- **Framer Motion 12.23.12** - Animation library

State Management & Data

- **Redux Toolkit 2.9.0** - State management
- **React Redux 9.1.2** - React-Redux bindings
- **Axios 1.11.0** - HTTP client
- **React Hook Form 7.62.0** - Form handling
- **Zod 4.0.15** - Schema validation

Development Tools

- **ESLint 9.32.0** - Code linting
- **Prettier 3.6.2** - Code formatting
- **TypeScript Support** - Type checking via JSConfig

Backend Technologies

Core Framework

- **.NET 9.0** - Runtime and framework
- **ASP.NET Core** - Web API framework
- **C#** - Programming language

Database & ORM

- **MySQL** - Primary database
- **Entity Framework Core 9.0.6** - ORM
- **MySql.EntityFrameworkCore 9.0.6** - MySQL provider

API & Documentation

- **Swagger/OpenAPI** - API documentation
- **Swashbuckle.AspNetCore 9.0.4** - Swagger integration

Authentication & Security

- **JWT Bearer Authentication** - Token-based auth
- **Auth0 Integration** - Identity provider
- **Microsoft.AspNetCore.Authentication.JwtBearer 9.0.9**

Development & DevOps

Version Control

- **Git** - Source control
- **GitHub** - Repository hosting
- **GitHub Actions** - CI/CD (configured)

Development Environment

- **Visual Studio Code** - Primary IDE
- **Node.js 20+** - Frontend runtime
- **.NET CLI** - Backend development
- **Docker Support** - Containerization ready

Project Structure

Root Directory

```
groupproject-team_17/
├── .github/                      # GitHub Actions workflows
├── .vscode/                      # VS Code settings
└── frontend/                     # Frontend application
├── backend/                      # Backend services
├── COMP5241-sem1-grp17.sln      # .NET solution file
├── README.md                      # Project overview
├── RelationMapping.md            # Database relationships
└── *.csv                          # Test data files
```

Frontend Structure (frontend/web-app/)

```
src/
├── _mock/                         # Mock data for development
├── api/                           # API integration layer
└── app/                            # Next.js App Router pages
├── assets/                         # Static assets
├── auth/                           # Authentication context
├── components/                    # Reusable UI components
│   ├── error-dialog/              # Global error handling
│   ├── iconify/                  # Icon components
│   └── [other-components]/
├── contexts/                      # React contexts
├── layouts/                        # Page layouts
├── lib/                            # Utility libraries
├── redux/                          # State management
├── routes/                         # Route definitions
└── sections/                      # Page-specific components
    ├── course/                   # Course management
    ├── dashboard/                 # Dashboard components
    └── [other-sections]/
├── theme/                          # MUI theme configuration
└── utils/                          # Utility functions
```

Backend Structure (backend/)

```
└── InteractiveHub.WebAPI/          # Main API project
    ├── Controllers/                # API controllers
    ├── ControllerBase/           # Base controller classes
    ├── Filters/                   # Action filters
    ├── Middleware/                # Custom middleware
    └── Program.cs                 # Application entry point

    └── InteractiveHub.Service/      # Business logic layer
        ├── ClassRelated/           # Course management
        |   ├── Db/                  # Database contexts
        |   ├── Migrations/          # EF migrations
        |   └── Model/                # Data models
        ├── Logger/                 # Logging services
        └── Shared/                  # Shared utilities

    └── InteractiveHub.Service.ClassManagement/ # Course management service
    └── InteractiveHub.Service.Logger/         # Logging service
    └── TestingCore/                      # Test projects
```

🚀 Key Features

1. Course Management

- **CRUD Operations:** Create, read, update, delete courses
- **Student Enrollment:** CSV import/export functionality
- **Course Settings:** Enable/disable, archive courses
- **Bulk Operations:** Mass student management

2. Interactive Classroom (next stage)

- **Real-time Activities:** Polls, quizzes, Q&A sessions
- **Student Participation:** Live engagement tracking
- **Activity Analytics:** Participation metrics and insights
- **Session Management:** Start/stop interactive sessions

Security Features

Authentication & Authorization

- **JWT Tokens:** Secure token-based authentication
- **Auth0 Integration:** Third-party identity provider
- **Role-based Access:** Different permissions for teachers/students
- **Session Management:** Secure session handling

Data Protection

- **Input Validation:** Comprehensive data validation
- **SQL Injection Prevention:** Parameterized queries via EF Core
- **CORS Configuration:** Controlled cross-origin requests

Database Design

Core Entities

- **Courses:** Course information and settings
- **Students:** Student profiles and enrollment
- **Activities:** Interactive learning activities
- **Responses:** Student activity responses
- **Sessions:** Classroom session tracking

Relationships

- **One-to-Many:** Course → Students (enrollment)
- **One-to-Many:** Course → Activities
- **Many-to-Many:** Students ↔ Activities (responses)

Project Repository: [GitHub - groupproject-team_17](#)

Documentation: Available in repository README and inline comments

Last Updated: October 2025