

Project Handover: Knowledge Sharing & Interactive Quiz Platform

1. Project Overview

ระบบ Web Application สำหรับจัดกิจกรรมอบรมและทดสอบความรู้พนักงาน (50-100 คน) แบบ Real-time เน้นความเร็วในการประมวลผลและการวัดผล (Pre-test vs Post-test) โดยมีฟีเจอร์หลักคือ Live Quiz แข่งขันตอบคำถามพร้อม Leaderboard

2. Technology Stack

- **Backend:** Node.js, Express.js
- **Database:** MongoDB (ใช้ Mongoose ODM)
- **Real-time:** Socket.io (v4)
- **Frontend:** React.js (Vite Build Tool)
- **Reporting:** ExcelJS (สำหรับ Export Report)

3. Database Schema (`server/models/Schemas.js`)

โครงสร้างฐานข้อมูลล่าสุด (แก้ไขเรื่อง Array ของ Options แล้ว)

JavaScript

```
// 1. Course (หลักสูตร)
const CourseSchema = {
  title: String,
  description: String,
  isActive: Boolean,
  // เก็บลิงก์ความรู้ภายนอก
  resourceLinks: [{
    title: String,
    url: String,
    type: { type: String, enum: ['video', 'app', 'pdf', 'website'] }
  }]
};

// 2. Question (คำถาม)
const QuestionSchema = {
  courseId: ObjectId (Ref: Course),
  text: String,
  type: enum['pre-test', 'post-test'],
  options: [String], // *แก้ไขล่าสุด: เก็บเป็น Array of Strings
  correctIndex: Number, // 0-3
```

```

    timeLimit: Number // default 20s
  };

// 3. User (ผู้เล่น)
const UserSchema = {
  employeeId: String,
  nickname: String,
  currentCourseId: ObjectId (Ref: Course),
  totalScore: Number // คะแนนรวมจากการแข่ง Live Quiz
};

// 4. Result (Log การตอบ)
const ResultSchema = {
  userId: ObjectId,
  questionId: ObjectId,
  isCorrect: Boolean,
  scoreEarned: Number,
  responseTimeMs: Number,
  type: enum['pre-test', 'post-test']
};

```

4. API Endpoints (server/server.js)

ฝั่ง User

- GET /api/courses : ดึงรายชื่อคอร์สทั้งหมด (ใช้แสดงหน้าเลือกคอร์ส)
- POST /api/join-course : ลงทะเบียน User เข้าสู่คอร์ส (Body: {employeeId, nickname, courseId})
- GET /api/pre-test/:courseId : ดึงข้อสอบ Pre-test ของคอร์สนั้นๆ
- POST /api/submit-pretest : ส่งคำตอบ Pre-test (Body: {userId, answers: [{questionId, answerIndex}], courseId})

ฝั่ง Admin

- GET /api/questions/:courseId : ดึงข้อสอบ **Post-test** ทั้งหมดของคอร์ส (สำหรับหน้า Control Panel)
 - GET /api/admin/export-report : ดาวน์โหลดไฟล์ Excel สรุปผลการอบรม (Learning Gain)
-

5. Socket.io Events (Real-time Flow)

การเชื่อมต่อ

- **Room:** ใช้ courseId เป็นชื่อ Room เพื่อแยกห้องสอบแต่ละวิชา

Events

Event Name	Direction	Payload	Description
join_course_room	Client -> Server	{ courseId, userId }	User/Admin แจ้งเข้าห้อง สอบ
admin_start_question	Admin -> Server	{ courseId, questionId }	Admin กดปุ่ม ปล่อยโจทย์
receive_question	Server -> All Clients	{ _id, text, options, timeLimit, startTime }	ส่งโจทย์ไปที่ จอมือถือทุก คน
submit_answer	Client -> Server	{ questionId, answerIndex, clientStartTime }	User ส่งคำ ตอบกลับมา
answer_result	Server -> Client	{ isCorrect, score }	แจ้งผลถูก/ผิด กลับไปหาคน ตอบทันที
update_leaderboard	Server -> All Clients	[Array of Top 5 Users]	อัปเดตตาราง คะแนน Real- time

6. Project Structure & Files

/server (Backend)

- server.js : Main Application, API Routes, Socket.io Logic, Scoring Logic.
- models/Schemas.js : Mongoose Models.
- seed.js : สคริปต์สร้างข้อมูลและสร้างข้อมูลจำลอง (ตัวเก่า).
- repair.js : **(สำคัญ)** สคริปต์ซ่อมข้อมูลล่าสุด ใช้สำหรับ Reset Database เมื่อมีการแก้ Schema หรือ Data หาย.

/client (Frontend)

- src/main.jsx : ตั้งค่า Router (/ = User, /admin = Admin).

- `src/App.jsx` : หน้าจอฝั่ง **User** (Login -> Select Course -> Pre-test -> Lobby -> Live Quiz).
 - `src/Admin.jsx` : หน้าจอฝั่ง **Admin** (Control Panel, Live Leaderboard).
-

7. Current Content (Data)

ปัจจุบันระบบมีข้อมูลชุด "ภาษีเงินได้บุคคลธรรมดา" ประกอบด้วย:

- **Resource Links:** ลิงก์ไปยัง App TaxPro และคลิป YouTube.
- **Pre-test:** 5 ข้อ (ความรู้พื้นฐาน).
- **Post-test:** 5 ข้อ (การคำนวณและเทคนิค).

8. How to Run (วิธีใช้งาน)

1. Start Backend:

Bash

```
cd server
npm install
node repair.js # รันครั้งแรก หรือเมื่อต้องการรีเซ็ตข้อมูลใหม่
node server.js
```

2. Start Frontend:

Bash

```
cd client
npm install
npm run dev
```

3. Access:

- **User:** `http://localhost:5173`
- **Admin:** `http://localhost:5173/admin`