

## Assignment 2 - Question Bank Management Application

### Instructions

In this exercise, you will expand and enhance the project from Assignment 01 to build a question bank management application with a user interface using EJS and Handlebars. The exercise will focus on integrating existing APIs from Assignment 01 into the new interface, ensuring CRUD functionality for questions and quizzes.

Template engine is a part of Express that enables us to use static files in our applications. Template engine converts variables to values and changes the template to HTML files to send to the client. The default template engine of Express is Jade, but EJS is the most commonly used engine

In this article we will cover:

- Setting up a project
- Configuring app.js to use EJS
- Creating EJS partials
- Using partials in views
- Passing data to EJS files (views and partials)

### Request

#### User interface (UI)

- Use EJS and Handlebars to create user interfaces.
- Use Bootstrap or CSS to create beautiful and user-friendly interfaces.

#### Main function

- Quiz Management:
  - o Displays a list of quizzes.
  - o Add, edit, delete quiz.
  - o View quiz details including related questions.
- Management Questions:
  - o Show list of questions.
  - o Add, edit, delete questions.
  - o Add questions to a specific quiz.

#### API integration

- Use the APIs created in Assignment 01 to manipulate the MongoDB database.
- Use API endpoints like GET, POST, PUT, DELETE for quizzes and questions.

## Step 1: Prepare the Environment

- Initialize project: Create a new directory for the project and initialize npm:
- Install necessary modules: Install Express, EJS, Handlebars, Body-Parser, Method-Override, Axios
- Install Bootstrap 5: CDN can be used to integrate Bootstrap 5 into the project.

## Step 2: Project Structure

Create necessary folders and files:

```
routes/  
  index.js  
  quiz.js  
  question.js  
views/  
  layouts/  
    main.hbs  
  partials/  
    header.hbs  
    footer.hbs  
    index.ejs  
  quiz/  
    list.ejs  
    details.ejs  
    create.ejs  
    edit.ejs  
  questions/  
    list.ejs  
    details.ejs  
    create.ejs  
    edit.ejs  
public/  
  css/  
    style.css
```

## Step 3: Set up Express and Handlebars

Configuring Express and Handlebars:

- Set up Handlebars as the main view engine and EJS for specific pages.
- Set up necessary middleware such as Body-Parser, Method-Override and Static.

## Step 4: Build API

- Create routes for quiz:
  - o Create routes to handle CRUD for quiz.
  - o Use Axios to call APIs from Assignment 01.
- Create routes for questions:
  - o Create routes to handle CRUD for queries.
  - o Use Axios to call APIs from Assignment 01.

## Step 5: Create User Interface

- Design the main interface with Bootstrap:
  - o Use Bootstrap 5 to create user-friendly interfaces.
  - o Design interface elements such as header, footer, and quiz and question management pages.
- Create EJS pages for quiz:
  - o Create EJS pages to display quiz list, quiz details, create and edit quiz.
  - o Use Bootstrap's CSS classes to create beautiful forms and lists.
- Creating EJS pages for the question:
  - o Create EJS pages to display question list, question details, create new and edit questions.
  - o Use Bootstrap's CSS classes to create beautiful forms and lists.

## Step 6: Test the Application

Run the application

### With /questions

- List of Question

#### Questions

Create New Question

<a href="#">What is the capital of France?</a>	Delete
<a href="#">What is a Control in WinForms?</a>	Delete

- Create a Question

## Create New Question

Question Text:

Options (comma separated):

Correct Answer Index:

Create

### - Display Question

What is the capital of France?

Paris, London, Berlin, Madrid (Correct)

Edit Question

### - Edit a Question

## Edit Question

Question Text:

What is the capital of France?

Options (comma separated):

Paris, London, Berlin, Madrid

Correct Answer Index:

0

Update

## With /quizzes

### - List of Quiz

## Quizzes

Create New Quiz

History Quiz	Delete
What is Windows Forms (WinForms)?	Delete
What language is commonly used to develop WinForms applications?	Delete

### - Display Quiz

## History Quiz

Test your knowledge of scientific concepts!

### Questions

What is the capital of France?

Edit Quiz

- Create a Quiz

### Create New Quiz

Title:

Description:

Create

- Edit a Quiz

### Edit Quiz

Title:

History Quiz

Description:

Test your knowledge of scientific concepts!

Update

- Check the routes in the browser:
  - Visit routes like /quizzes and /questions to test CRUD functions.
  - Ensure that the APIs function properly and the user interface displays correctly.

## Conclude

This exercise helps you get familiar with using EJS and Bootstrap 5 to create a user interface for a question bank management application, integrating with APIs built in Assignment 01.

### Note that the API is used with HTTPS

Using HTTPS for secure communication between web applications and APIs is important. In the context of Assignment 01, when you want to use HTTPS for API requests, you need to understand how to configure and use HTTPS in Express and Axios.

Example:

```
// routes/question.js
const https = require('https');
const express = require('express');
const router = express.Router();
const axios = require('axios');
```

```
// URL của API
const apiUrl = 'https://localhost:3443';

const axiosInstance = axios.create({
  httpsAgent: new https.Agent({ rejectUnauthorized: false })
});
// GET all questions
router.get('/', async (req, res) => {
  try {
    const response = await axiosInstance.get(`${apiUrl}/questions`);
    res.render('question/list', { questions: response.data });
  } catch (err) {
    console.error(err);
    res.status(500).send('Internal Server Error');
  }
});
```