**Kế hoạch sơ bộ**

**1. Phân Tích Yêu Cầu**

* Xác định các yêu cầu chính cho ứng dụng quiz:
  + Nhiều cấp độ câu hỏi (Easy, Medium, Hard)
  + Nhiều chủ đề câu hỏi (SQL, C#, Algorithm, v.v.)
  + Các loại câu hỏi (trắc nghiệm, tự luận, điền đáp án, chọn nhiều đáp án đúng, v.v.)
  + Giao diện người dùng hỗ trợ HTML và rich media (hình ảnh, video, âm thanh)
  + Tạo quiz với thông tin về tiêu đề, nội dung, và đáp án.

**2. Thiết Kế Cơ Sở Dữ Liệu**

* Sử dụng các bảng đã được tạo trước đó để lưu trữ thông tin quiz, câu hỏi, đáp án, người dùng, và mối quan hệ giữa chúng.
* Xác định các ràng buộc cần thiết (khóa chính, khóa ngoại).

**3. Tạo Dự Án C#**

* Tạo một dự án mới trong Visual Studio hoặc IDE tương tự:
  + Chọn **ASP.NET Core Web API** hoặc **.NET Core** cho backend.

**4. Cài Đặt Thư Viện và Công Cụ**

* Cài đặt các gói NuGet cần thiết, như:
  + **Entity Framework Core** (nếu bạn quyết định sử dụng nó cho ORM).
  + **MSSQL Server** cho kết nối với SQL Server.
  + **Moq** cho kiểm thử đơn vị (Unit Testing).

**5. Triển Khai Cơ Sở Dữ Liệu**

**6. Xây Dựng Các Mô Hình (Models)**

**7. Triển Khai Controllers**

* Tạo các Controller để xử lý các yêu cầu API:

|  |  |
| --- | --- |
| **Controller** | **Main methods** |
| QuizController | CreateQuiz, GetAllQuizzes, GetQuizById, UpdateQuiz, DeleteQuiz |
| QuestionController | CreateQuestion, GetAllQuestions, GetQuestionById, UpdateQuestion, DeleteQuestion |
| AnswerController | CreateAnswer, GetAnswersByQuestionId, UpdateAnswer, DeleteAnswer |
| UserController | RegisterUser, UpdateUser, AuthenticateUser |
| UserQuizController | StartQuiz, GetUserQuizzes, CompleteQuiz, GetQuizResults |
| UserAnswerController | SubmitAnswer, GetUserAnswersByQuiz, EvaluateAnswer |
| TopicController | CreateTopic, GetAllTopics, UpdateTopic, DeleteTopic |
| LevelController | GetAllLevels, CreateLevel, UpdateLevel, DeleteLevel |

**8. Xây Dựng Repositories và Services**

**Repositories**

Repositories chịu trách nhiệm giao tiếp với cơ sở dữ liệu cho từng thực thể riêng biệt. Mỗi repository sẽ có các phương thức CRUD để truy cập và thao tác dữ liệu trong database.

* **UserRepository**: Quản lý dữ liệu người dùng.
  + AddUser, GetUserById, GetUserByEmail, UpdateUser, DeleteUser
* **QuizRepository**: Quản lý các quiz.
  + AddQuiz, GetQuizById, GetAllQuizzes, UpdateQuiz, DeleteQuiz
* **QuestionRepository**: Quản lý câu hỏi.
  + AddQuestion, GetQuestionById, GetQuestionsByCriteria, UpdateQuestion, DeleteQuestion
* **AnswerRepository**: Quản lý câu trả lời.
  + AddAnswer, GetAnswerById, GetAnswersByQuestionId, UpdateAnswer, DeleteAnswer
* **UserQuizRepository**: Quản lý kết quả làm quiz của người dùng.
  + AddUserQuiz, GetUserQuizById, GetUserQuizzesByUserId, UpdateUserQuiz, DeleteUserQuiz
* **RoleRepository**: Quản lý vai trò của người dùng.
  + AddRole, GetRoleById, GetAllRoles, UpdateRole, DeleteRole
* **UserRoleRepository**: Quản lý quan hệ giữa người dùng và vai trò.
  + AddUserRole, GetUserRoleById, GetRolesByUserId, DeleteUserRole
* **TagRepository**: Quản lý các tag của quiz.
  + AddTag, GetTagById, GetAllTags, UpdateTag, DeleteTag
* **QuizTagRepository**: Quản lý quan hệ giữa quiz và tag.
  + AddQuizTag, GetQuizTagsByQuizId, DeleteQuizTag
* **TypeRepository**: Quản lý các type của câu hỏi.
  + AddType, GetTypeById, GetAllTypes, UpdateType, DeleteType

**Services**

Services chịu trách nhiệm thực hiện logic nghiệp vụ bằng cách phối hợp các repository. Các service này đảm bảo rằng logic nghiệp vụ không bị trùng lặp trong các controller và dễ bảo trì.

* **UserService**: Quản lý các tác vụ liên quan đến người dùng.
  + RegisterUser, AuthenticateUser, GetUserProfile, UpdateUserProfile, AssignRoleToUser, GetUserRoles
* **QuizService**: Quản lý logic của quiz.
  + CreateQuiz, GetQuiz, GetQuizzesByCriteria, AddQuestionToQuiz, SaveQuizResult, GetQuizResultsByUser
* **QuestionService**: Xử lý câu hỏi, chọn câu hỏi theo tiêu chí.
  + CreateQuestion, GetQuestionsByLevel, GetQuestionsByTopic, GetQuestionTypes, UpdateQuestion, DeleteQuestion
* **AnswerService**: Quản lý câu trả lời và logic kiểm tra đáp án.
  + SaveUserAnswer, GetCorrectAnswersForQuestion, GetUserAnswersForQuiz, EvaluateAnswers
* **RoleService**: Quản lý vai trò người dùng.
  + CreateRole, GetRoles, AssignRoleToUser, GetUsersByRole
* **TagService**: Quản lý tag cho quiz và câu hỏi.
  + CreateTag, AssignTagToQuiz, GetQuizzesByTag, GetAllTags
* **UserQuizService**: Quản lý kết quả làm quiz của người dùng.
  + GetUserQuizHistory, GetQuizScore, SaveQuizCompletionTime, CalculateQuizResult
* **TypeService**: Quản lý các loại câu hỏi.
  + GetAllTypes, AddType, UpdateType, DeleteType

**9. Triển Khai Kiểm Thử Đơn Vị (Unit Testing)**

* Tạo các bài kiểm thử đơn vị cho từng phần của ứng dụng:
  + Kiểm thử các phương thức trong QuizService, UserService, v.v.
  + Sử dụng Moq để kiểm thử các phương thức tương tác với cơ sở dữ liệu.

**10. Tài Liệu và Hướng Dẫn Sử Dụng**

* Viết tài liệu hướng dẫn cho người dùng cuối để sử dụng ứng dụng quiz.
* Tạo hướng dẫn cho nhà phát triển để bảo trì và phát triển thêm tính năng mới.

**11. Triển Khai Ứng Dụng**

* Triển khai ứng dụng lên server hoặc cloud service (Azure, AWS, Heroku, v.v.).
* Đảm bảo rằng ứng dụng có thể truy cập từ xa và hoạt động như mong đợi.

**12. Bảo Trì và Cải Tiến**

* Theo dõi ứng dụng sau khi triển khai để phát hiện và sửa lỗi.
* Thu thập phản hồi từ người dùng để cải tiến và phát triển tính năng mới.

**Lưu Ý**

* **Quản lý phiên bản:** Sử dụng Git để quản lý mã nguồn và theo dõi các thay đổi.
* **Bảo mật:** Cân nhắc các biện pháp bảo mật cho API và thông tin người dùng.