**MD04\_TS\_Hackathon\_Skill\_Advance\_01**

**ĐÁNH GIÁ CHẤT LƯỢNG KỸ NĂNG NÂNG CAO**

**MODULE 04 – TYPESCRIPT – HACKATHON 06**

**THỜI GIAN: 240 phút**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

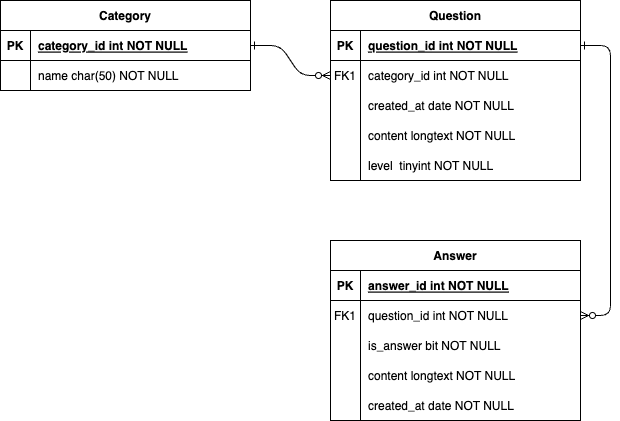
**Yêu cầu:**

* *Đảm bảo clean code, naming convention trong việc đặt tên các biến, phương thức, lớp*
* *Tạo project folder theo hướng dẫn sau:*
* *Tạo folder HN\_JS230213\_AD\_[StudentName] chứa toàn bộ file mã nguồn*
* *Ví dụ: HN\_ JS230213\_AD\_NguyenVanA*
* *Sau khi hoàn thành project, đẩy code lên github và nộp link cho người phụ trách*
* *Công nghệ sử dụng: HTML, CSS, TypeScript, Express, MySQL, React*
* *IDE: Visual Studio Code*

**Thực hành:**

**Bài 1 (20 điểm) :**

Cho sơ đồ mô hình quan hệ giữa các thực thể (ERD - Entity Relationship Diagram).



Viết script thực hiện các yêu cầu sau

* Thiết kế CSDL quan hệ theo ERD trên trong MySQL **[15 điểm]**
* Thêm các bản ghi mẫu vào từng bảng *(Thêm mới ít nhất 6 questions thuộc 2 categories khác nhau và mỗi question có 4 answers)* **[5 điểm]**

**Bài 2 (50 điểm) :**

Tạo các API endpoint sau để tương tác với các bảng đã tạo ở CSDL của **Bài 1**

**Category:**

* Xây dựng một API với phương thức **GET** cho phép lấy một category theo category\_id

**/api/v1/categories/:id[5 điểm]**

* Xây dựng một API với phương thức **GET** cho phép lấy toàn bộ dữ liệu về category

**/api/v1/categories[5 điểm]**

* Xây dựng một API với phương thức **POST** cho phép thêm mới một category

**/api/v1/categories[5 điểm]**

**Question:**

* Xây dựng một API với phương thức **GET** cho phép lấy một question theo question\_id

**/api/v1/questions/:id[5 điểm]**

* Xây dựng một API với phương thức **GET** cho phép lấy về một question với toàn bộ answers của question đó

**/api/v1/questions/:id/answers[10 điểm]**

* Xây dựng một API với phương thức **GET** cho phép lấy về toàn bộ questions

**/api/v1/questions[5 điểm]**

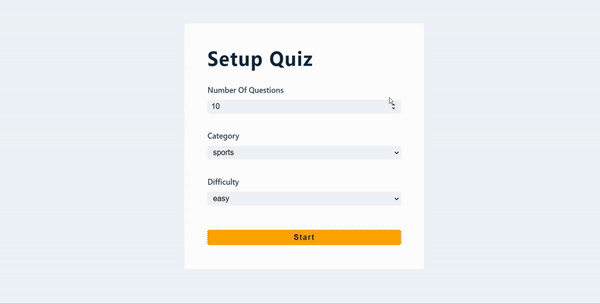
* Xây dựng một API với phương thức **GET** cho phép lấy về toàn bộ questions theo các tiêu chí lọc là **category, level, limit (Số lượng câu hỏi muốn lấy về)**

**/api/v1/questions?category=[category\_id]&level=[easy/medium/hard]&limit=[][10 điểm]**

* Xây dựng một API với phương thức **POST** cho phép thêm mới một question

**/api/v1/questions[5 điểm]**

**Bài 3 (30 điểm + 5) :**



*(Kéo xuống dưới cùng để nhìn giao diện từng trang)*

Tiến hành kết hợp giữa các kiến thức đã học về Express, SQL để clone lại giao diện và tạo ra một ứng dụng quiz-app đầy đủ các tính năng

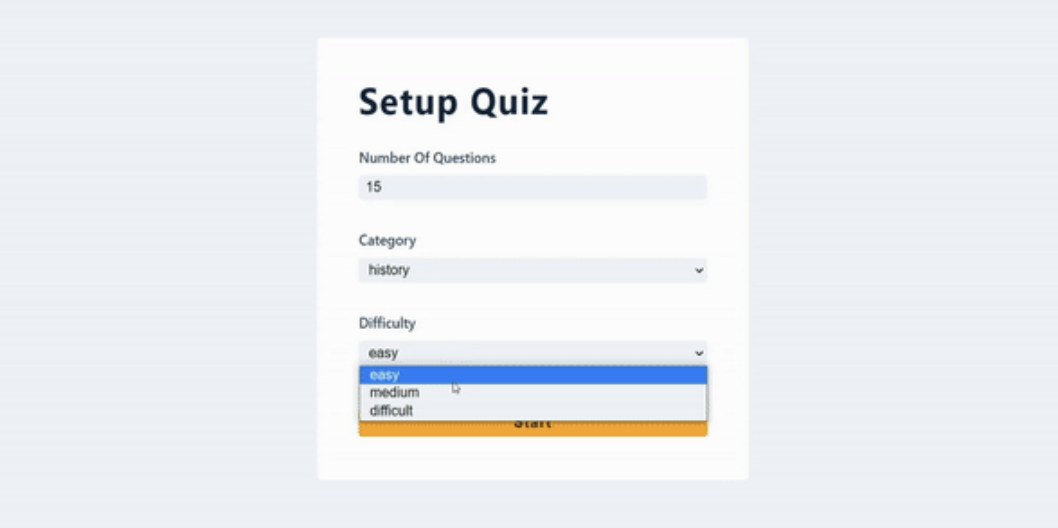
* Cắt giao diện đẹp, đúng → Giao diện gần đúng với giao diện được giao, có thể tùy biến màu sắc, font chữ, icon trong bài**[5 điểm]**
* **Trang chủ:**
* Xây dựng trang chủ với một form submit các option → Lấy dữ liệu từ endpoint ***GET all questions with filters*** và điều hướng sang trang question detail**[5 điểm]**
* **Trang Question detail:**
* Xây dựng render câu hỏi và câu trả lời ra màn hình trang question detail

**[5 điểm]**

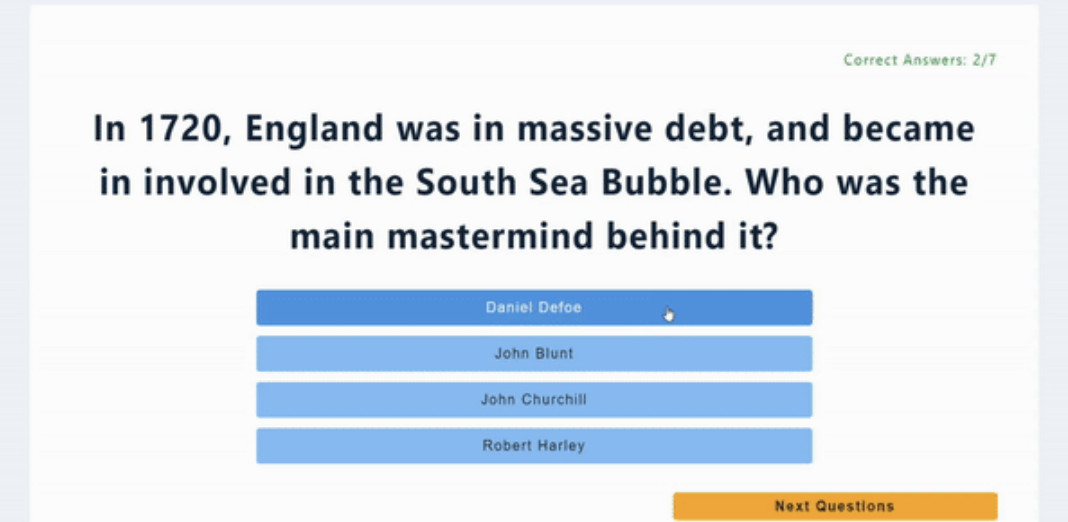
* Xây dựng chức năng trả lời câu hỏi → Với mỗi câu trả lời đúng hoặc sai thì sẽ tính tổng số câu trả lời đúng trực tiếp lên màn hình *(Chú ý: Khi next sang câu hỏi tiếp theo cũng tính là trả lời sai 1 câu)***[10 điểm]**
* Xây dựng chức năng pop-up lên % số câu trả lời đúng sau khi hoàn thành quiz và điều hướng về trang chủ sau khi nhấn ok**[5 điểm]**
* **BONUS:** Sử dụng React để code frontend**[5 điểm]**

*Lưu ý: Chỉ tính điểm khi chức năng phải thực hiện đúng theo yêu cầu*

*Trang chủ:*



*Trang Question Detail:*



*Pop-up*