Assume you're building an adaptive learning portal for students to practice questions relevant to an exam. The hierarchy of how an exam branches out into subjects, topics, chapters and questions is as below :

EXAM—> SUBJECTS—> TOPICS—> CHAPTERS—> QUESTIONS

1. Exam has many subjects. Subject has many topics. Topic has many chapters. And Chapter has many questions.
2. Questions can be either Easy, Medium or Hard category. A Question will have 4 answer options and a skip option too. User can either choose from the 4 options or can even skip the question.
3. A user can start answering the questions in any particular chapter. Once a question has been answered by a user, the user cannot re-answer it.

Requirement:

Primary requirement involves you to create the following API :

1. Create a Database Schema and seed initial content for Users, Exams, Subjects, Topics, Chapters and Questions

2. An API to Answer a particular question

3. An API to show a list of random questions to this user, from within a chapter or topic or subject or Exam. The response should not list questions the user has already answered. If user has answered all questions, then show the questions the user has skipped

4. An API which gives us the user's percentage of correct answers, wrong answers, skipped questions and unseen questions within a chapter, within a topic, within a subject and within an exam

5. Controller / Request Specs to test the above APIs

What we will be looking for:

1. Clean database design (preferably use SQL)
2. Model relationships and validations
3. Modular, clean, refactored, working correct code
4. Extensive test coverage
5. Security aspects in the code. Are there vulnerabilities in the codebase ? Can the data be corrupted easily ?

Note:

1. Use git and commit as much as possible, often using descriptive commits descriptions (required)
2. Use OOP principles
3. Use any external libraries and data storage which you feel necessary

Best of luck!