- Define the data models:
- User: Representing user details such as name, email, and password.
- Quiz: Representing quiz details such as title, description, and duration.
- Question: Representing individual quiz questions with options and correct answer.
- Result: Representing user's quiz results, including the score.

- Create the necessary endpoints:
- User Endpoints:
- Register: POST request to create a new user account.
- Login: POST request to authenticate and generate a user session token.
- Logout: POST request to invalidate the user session token.

- Quiz Endpoints:
- Create Quiz: POST request to create a new quiz.
- Get Quiz: GET request to retrieve quiz details by ID.
- List Quizzes: GET request to retrieve a list of all available quizzes.

- Question Endpoints:
- Add Question: POST request to add a new question to a quiz.

- Result Endpoints:
- Submit Quiz: POST request to submit user's quiz answers and calculate the score.
- Get Result: GET request to retrieve user's quiz result by ID.

- Implement the business logic for each endpoint:
- User Service: Implement user registration, authentication, and session management.
- Quiz Service: Implement quiz creation, retrieval, and listing.
- Question Service: Implement question addition, retrieval, and listing for a specific quiz.
- Result Service: Implement quiz submission, result calculation, and retrieval.

- Configure the database and data access:
- Set up the database connection and define the necessary entities (User, Quiz, Question, Result).
- Create repositories/interfaces for each entity to perform CRUD operations.

- Implement the controllers and map the endpoints:
- Create controllers for each endpoint and map the request methods and paths accordingly.

- Handle the request parameters, body, and headers to call the corresponding service methods.
- Implement necessary security measures:
- Implement authentication and authorization mechanisms to protect sensitive endpoints.
- Secure user registration, login, and quiz submission using tokens or session management.

- Test and debug the API endpoints:
- Use tools like Postman or curl to test each endpoint and verify the expected behavior.
- Handle and debug any errors or exceptions that occur during testing.