Part 1: Full stack technical test

You are to build a project consist of key elements as below:

1. Create a website consist of a login page, registration page, and a content page with navigation
2. A MySQL database that store registration and manage page contents
3. You can demonstrate your front-end and back-end skills in content pages, such as:
    a. (must) Async function using promise method
    b. (must) Advance DB query demonstration
    c. (must) AJAX content loading in modal
    d. (optional) editable table listing
    e. (optional) carousel content card
    f. Etc. (any other special feature you would like to show)
4. Git repo that we can download the project
5. Step by step guide for us to deploy the project to our localhost and run

Requirement:

- The website should use our company logo and our corporate color to prove the work is custom made for this test
- The project must use **PHP, MySQL, and Vue.js**

Notes:

- You may use any additional framework or library in your project
- Aesthetic and creativity of the layout and content is also important

Part 2: Backend Microservice / REST API test (Beginner/Intermediate)

You are to build a project consist of key elements as below:

1. Create a microservice REST API with routing/controller/version such as:
   a. Eg: your_domain/cart/api/v1.0/fetch
2. First fetch cart details from https://dummyjson.com/carts/2, then extract from response data to find “userId”
3. Next, fetch user details by “userId” found in step 2 from https://dummyjson.com/users/<userId>. Replace with actual userId found in step 2.
4. Rebuild final json to include cart & user details and return consolidated payload as API response/output.

Requirement:

- Use NodeJS
- This application should be Dockerized into a Dockerfile

Part 3: Backend Microservice / REST API test (Advanced)

You are to build a project consist of key elements as below:

1. Create a microservice REST API with routing/controller/version such as:
    a. Eg: your_domain/dynamicimage/api/v1.0/generate
2. This API will be have GET parameters appended to url above as part of the request with the following fields
    a. Image Url -> this will be used as background image
    b. Title
    c. Description
3. From GET params in 2, generate a dynamic image using html/css and return API response/output in “image/jpeg” format
4. End result should be something like this:

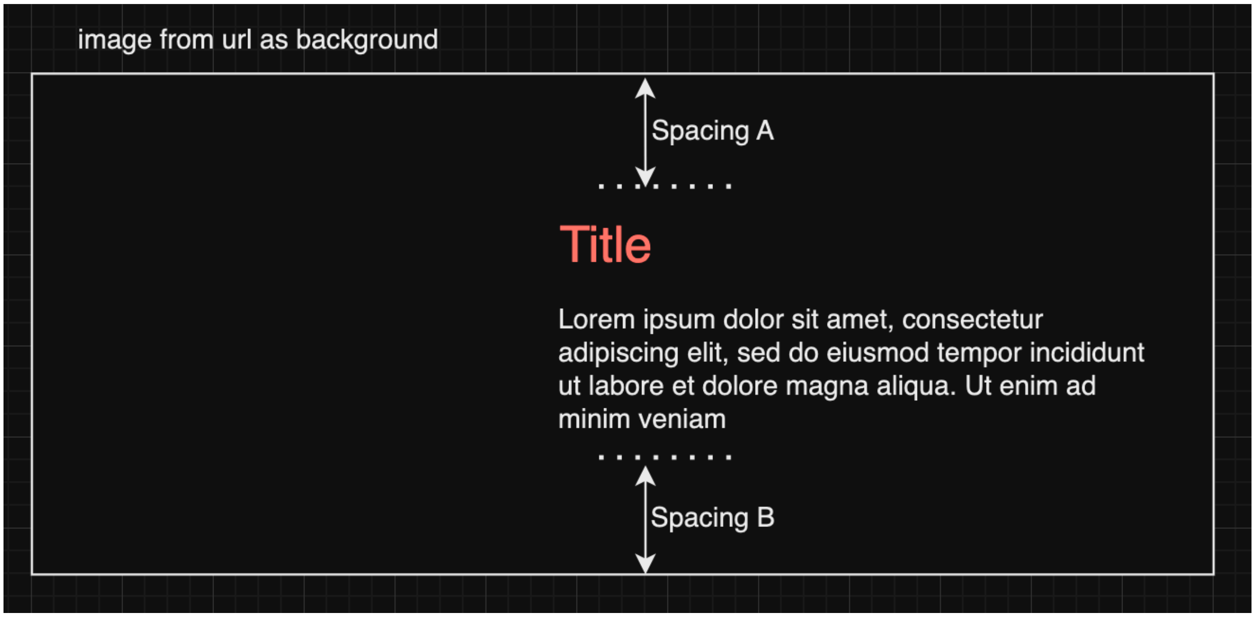
image from url as background
Spacing A
Title
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam
Spacing B

Requirement:

- Spacing A and spacing B (refer to image) must be dynamically adjusted according to content height. The top and bottom spaces should readjust to dynamically centre the content regardless of content height. Image size should be 700px by 400px. Title has to be red colour.
- Use NodeJS, EJS, Puppeteer, HTML, CSS (hint: Bootstrap 5)
- This application should be Dockerized into a Dockerfile