

Technical Assessment (Intern - Fullstack Developer)

Part 1: Multi-Tenancy in NextJS Project

Objective: Research and prepare a detailed document on implementing multi-tenancy in a NextJS project. Your document should cover the concept of multi-tenancy, why it is important, and how you would approach implementing it in a NextJS application.

Instructions:

- Start by explaining the concept of multi-tenancy and its benefits to web applications.
- Detail the architectural changes needed in a NextJS project to support multi-tenancy. Consider aspects such as database design, application structure, and data isolation.
- Discuss how to handle tenant identification (e.g., via subdomains, URL paths, or headers).
- Explain how you would configure NextJS (and any other necessary technologies) to serve different content or styles based on the tenant.
- Include any potential challenges you anticipate and how you would address them.
- Suggest tools or libraries that could facilitate the implementation of multi-tenancy in NextJS.
- Your document should be clear, concise, and targeted at a technical audience.

Submission: Submit your document in PDF format. Make sure it is well-organized and easy to read.

Part 2: React.js Card Component Using TailwindCSS (Bonus: ShadCn)

Objective: Demonstrate your frontend development skills by coding a responsive card component in React.js using TailwindCSS. If you're familiar with ShadCn, leveraging it for shadow effects will earn extra points.

Card Requirements:

- See attached card design.
- The image should be responsive and maintain its aspect ratio.
- The title should be bold and slightly larger than the description text.
- Implement a hover effect on the buttons for better user interaction.
- Ensure the card is styled using TailwindCSS, and the layout is responsive to different screen sizes.
- Extra Points: Use ShadCn for adding shadows to your card for a more polished look.

Instructions:

- Create a new React.js component for the card.
- Use TailwindCSS for styling. Ensure your code is clean and well-organized.
- If you use ShadCn, document how you incorporated it and the effect it had on your design.
- Your code should be written with readability and maintainability in mind.

Deployment & Submission: Deploy your application to Vercel. Ensure that your deployment is public and accessible.

Submit the following:

A link to your GitHub repository containing the source code.

A live link to your deployed application on Vercel.

Assessment Criteria:

- **Functionality:** The card component meets all the specified requirements.
- **Code Quality:** Your code should be clean, well-organized, and properly commented.
- **Design:** The component should be visually appealing and responsive.
- **Innovation:** Extra points for creativity and the effective use of ShadCn for shadows.
- **Deadline:** Please submit your document and code by 15th March, 2024.