

OVERVIEW:

CANDIDATES HAVE THE OPTION TO CHOOSE BETWEEN COMPLETING A FRONT-END OR A BACK-END TASK. COMPLETING BOTH TASKS IS NOT MANDATORY BUT COULD INCREASE THE CHANCES OF SELECTION AS IT SHOWCASES THE CANDIDATE'S FULL-STACK SKILLS. THE FRONT-END TASK SHOULD BE COMPLETED USING NEXT.JS, WHILE THE BACK-END TASK SHOULD UTILIZE NODE.JS.

Please submit your task(s) to internship@raycabio.com by the deadline mentioned in the LinkedIn message.

TASK A: FRONT-END DESIGN

Building an Interface

We are requesting that you precisely imitate the UI demonstrated in the image attached to this file. Your task is to replicate this design as accurately as possible.

Instructions:

1. Review the attached image to understand the desired UI elements.
2. Use Next.js to recreate the UI and interaction.
3. Aim for pixel-perfect precision in replicating the design seen in the provided sample design.
4. Encourage you to consider details such as using animations, transitions, and responsiveness.
5. Once you have completed the task, provide a live demo of your implementation. You can host it on a platform of your choice. **Providing a deployed version of your code is necessary for demonstrating.**
6. You must also share your code for a more comprehensive evaluation.

Task Delivery

- ✓ Link to live demo
- ✓ Code GitHub repository

TASK B: BACK-END MICROSERVICE

You are tasked with developing a microservice for managing customer support tickets. This system will allow users to create, update, and track support tickets. The task involves setting up a Node.js environment and implementing various API endpoints for ticket and user management. These endpoints will include functionalities such as creating, updating, deleting, and assigning tickets to users, as well as updating ticket statuses. You are also required to integrate a MongoDB database for data storage and retrieval. Implementing JWT-based authentication is essential to ensure secure access to the endpoints, with role-based access control for different user roles like customers, support agents, and admins. Additionally, the system should have a notification feature to alert users of ticket assignments or status changes, which can be done via email or real-time notifications using WebSockets. Comprehensive API documentation should be provided using tools like Swagger. The goal of this task is to evaluate your skills in backend development, including API design, database integration, and implementing security measures. You will need to deliver a link to the live API, a code repository, and the API documentation.

Evaluation Criteria

- Correct implementation of the required API endpoints.
- Proper database integration and data handling.
- Implementation of secure authentication and authorization mechanisms.
- Notification system functionality.
- Code quality, including proper documentation and clean code practices.

Task Delivery

- ✓ Link to the live API
- ✓ Code GitHub repository
- ✓ API documentation link