

Project Overview: Biotech Data Management System

In this assignment, you are tasked with building a simple data management system for a biotech research lab. The system will consist of both a back-end API service and a front-end application. The goal is to demonstrate your proficiency in using FastAPI for the back-end and React with Next.js for the front-end.

Requirements:

Backend (FastAPI):

Implement a FastAPI server with CRUD operations for managing data related to biological samples. The data should include at least the following fields:

- Sample ID
- Sample Name
- Date Collected
- Experiment Type
- Storage Location

Use a database for data persistence. You can choose either SQLite, PostgreSQL, or similar.

- Implement validation for input data on the server side. Include appropriate error handling for invalid requests.

Create API routes for:

- Retrieving a list of all samples
- Retrieving details of a specific sample
- Adding a new sample
- Updating an existing sample
- Deleting a sample

Include API documentation using FastAPI's built-in capabilities or Swagger.

Frontend (React with Next.js):

- Implement a React application using Next.js to interact with the FastAPI back end.
- Create a home page that displays a list of all samples retrieved from the API.
- Include a form to add a new sample with the necessary fields.
- Implement functionality to view the details of a specific sample and edit its information.
- Provide a way to delete a sample from the system.

- Ensure proper error handling and user feedback for CRUD operations.

Additional Considerations:

- Use a responsive design for the front end, ensuring a good user experience on both desktop and mobile devices.
- Employ state management on the front end (e.g., Redux or React Context) as needed.
- Demonstrate best practices for code organization, readability, and maintainability.

Submission:

- Provide clear instructions on how to set up and run your application.
- Include any necessary scripts for database initialization.
- Submit your code in a version-controlled repository (GitHub, GitLab, Bitbucket, etc.).
- Include a README file documenting your design decisions, any trade-offs made, and potential future improvements.

Evaluation Criteria:

- Adherence to the specified technologies (FastAPI, React, Next.js).
- Correct implementation of CRUD operations with proper error handling.
- Code organization, readability, and maintainability.
- Responsive design and overall user experience on the front end.
- Documentation quality and completeness.