

SDE 1 Fullstack Assessment - Todo List Application

Overview

This assessment requires you to build a fully functional Todo List application. Your task is to implement the application requirements shared below in both the frontend and backend. This assessment is designed to evaluate your fullstack development skills, code organization, attention to detail, and how you extend beyond the basic requirements. You may reference the provided frontend folder, backend.md, and database.md files to set up your project, or you can choose to create everything from scratch based on the requirements. The reference files can be downloaded from: [Reference Files](#)

Application Requirements

The application should be a Todo List with the following features:

1. Todo Management

- Create new todos with titles and descriptions
- Add tags and priorities to todos (High, Medium, Low)
- Tag/mention other users in todos (@username)
- Edit existing todos
- Delete todos

2. Todo Details

- Click on a todo to view its details (tags, priority, notes, users)
- Add notes to a todo via a modal when clicking an icon next to the todo

3. List View Features

- List all todos with basic information
- Implementation of pagination (either infinite scroll or numbered pagination)
- Filter todos by tags, priority, or users
- Sort todos by creation date, priority, etc.

4. Data Export [Optional]

- Export all todos of a user (JSON or CSV format)

5. User Management [Optional]

- **Backend:** Pre-create at least 5 different users so that when tagging, the system can validate if the user exists.
- **Frontend:** Provide a dedicated tab that displays all users, allowing you to switch between users. When switching to a different user, the application should display all todos associated with that user.

Note: Implement as many features as you can. We expect you not spend more than 2-3 hours on this assignment. Whatever is not implemented can be mentioned as 'Not Implemented' in your documentation.

Technical Requirements

Frontend

The basic HTML and CSS structure has been provided. You need to:

- Choose either React or Angular for the frontend implementation
- Implement the required features listed above
- Follow proper component structure and state management
- Create responsive design ensuring the application works on mobile devices
- Add appropriate error handling and loading states

Backend

You should implement the backend using one of the following technologies:

- Next.js
- Node.js (Express)
- Flask (Python)

The backend should:

- Provide RESTful API endpoints for all required functionality
- Implement proper error handling and validation
- Include authentication for user-specific operations
- Follow best practices for your chosen technology

Database

Choose either:

- MongoDB
- PostgreSQL

Design and implement an appropriate database schema for the application.

Assessment Criteria

You will be evaluated on:

1. **Functionality:** All features work as expected
2. **Code Quality:** Clean, maintainable code with consistent style
3. **Architecture:** Proper separation of concerns and project organization
4. **Git Usage:** Meaningful commit messages and logical commits
5. **Documentation:** Clear documentation on how to run and use the application
6. **Testing:** Appropriate tests for key functionality

Getting Started

1. Review the provided documentation
2. Choose your preferred tech stack from the options above
3. Implement the required features
4. Document your implementation decisions in the IMPLEMENTATION.md file
5. Submit your completed assignment

Submission

Your submission should include:

1. All source code in the repository
2. An IMPLEMENTATION.md file explaining:
 - Your chosen tech stack
 - How to run the application
 - Any assumptions or design decisions
 - Any additional features or improvements you made
3. Any necessary setup scripts or instructions

Timeline

Please complete this assessment and share it as instructed.

Questions

If you have any questions or need clarification, please reach out to your hiring contact.

Good luck!