Software Justification

Backend

Programming Language

We have chosen TypeScript and JavaScript, specifically using the Nest.js framework, for the backend. Nest.js provides a modular and structured architecture, making it well-suited for scalable and maintainable applications. TypeScript offers strong type safety, reducing runtime errors and improving code clarity, which is crucial for long-term maintainability. Nest.js also includes built-in support for RESTful APIs and GraphQL, allowing flexibility in data retrieval. To ensure clean architecture and maintainability, we will structure the backend into modules for different functionalities such as user authentication, task management, and notifications, while using dependency injection to enhance scalability.

Database

For data storage, we will use SQLite, as it is lightweight, easy to set up, and well-suited for applications with a smaller user base. Since SQLite is file-based, it eliminates the need for managing a separate database server, making it an ideal choice for our project, where we do not expect high concurrent traffic. We will use TypeORM to interact with the database, allowing us to define models and relationships in a structured manner. This setup ensures efficient data management while keeping the overall system simple and easy to maintain.

Container

To maintain consistency across development and deployment environments, we will use Docker to containerize our application. This guarantees that all dependencies and configurations are standardized, preventing the common “works on my machine” issue. By packaging the backend, database, and necessary dependencies into a Docker container, we ensure that the application runs reliably across different environments. If necessary, we will use Docker Compose to manage multiple services efficiently. This approach simplifies deployment, testing, and collaboration, ensuring a smooth development process.

Frontend

Programming Language

For the frontend of our project, we are using HTML, CSS, and JavaScript to create a structured, visually appealing, and interactive user experience. These languages provide the flexibility and functionality needed to develop a responsive and efficient project management system while ensuring ease of maintenance and scalability.

HTML (HyperText Markup Language)

HTML serves as the backbone of our frontend, defining the structure and content of our web pages. Its universal support across all browsers makes it the ideal choice for creating a solid foundation. Additionally, its semantic elements improve accessibility and search engine optimization (SEO), ensuring clarity and organization. Since HTML integrates seamlessly with CSS and JavaScript, it allows us to build a structured yet dynamic user interface.

CSS (Cascading Style Sheets)

CSS is essential for styling and designing the layout of our application. It enables us to create a clean, modern, and responsive design that adapts to different screen sizes, ensuring accessibility across various devices. By utilizing reusable styles and potential frameworks like Bootstrap or Tailwind CSS, we streamline development while maintaining consistency in the user interface.

JavaScript

JavaScript adds interactivity and dynamic functionality to our web pages. It allows us to implement features such as form validation, real-time updates, and interactive task management. Through event handling and asynchronous updates, JavaScript ensures a seamless user experience by dynamically updating content without requiring full page reloads. Its cross-browser compatibility further ensures that our application runs smoothly across different platforms.