**Software Development Document: Notes Application**

1. **Project Overview**

The Notes Application is a web-based application developed using Next.js, TypeScript and MongoDB. It provides users with the ability to create, delete, update, and view individual notes. Each note includes an id, title, content, created date, and updated date. The application uses Prisma Client to interact with a MongoDB database for data storage. Additionally, libraries such as Axios, React Icons, React Paginate, and Tailwind CSS are employed to enhance functionality and styling. The application is designed to be fully responsive, providing a seamless user experience across devices.

1. **Architecture and Technologies**

The application follows a client-server architecture. The frontend is built using Next.js, utilizing React components for dynamic UI rendering. Prisma Client acts as the middleware to communicate with the MongoDB database. Axios is used for making HTTP requests, while React Icons enhances visual elements. Pagination is achieved using the React Paginate library, and Tailwind CSS provides responsive styling.

1. **Database Design**

The database is structured to store notes, which consist of the following fields:

* Id: Auto generated and unique identifier for each note.
* Title: The title of the note.
* Content: The content of the note.
* Created Date: The date when the note was created.
* Updated Date: The date when the note was last updated.

Prisma Client is used to define the data models, perform database queries, and ensure seamless interaction between the application and MongoDB.

1. **UI/UX Design**

Tailwind CSS is used to style the user interface, ensuring a clean and modern design. The application is responsive, adapting to various screen sizes. The use of React Icons enhances the visual representation of actions.

1. **Development Process**

The development process involves several key steps:

* **Project Setup**: Setting up the Next.js project, installing dependencies, and configuring environment variables.
* **Frontend Development**: Creating React components for creating, deleting, updating, and viewing notes. Implementing form validations to prevent empty titles and content.
* **Backend Development**: Defining Prisma data models, setting up Prisma Client, and implementing CRUD operations for notes.
* **Pagination**: Utilizing React Paginate to implement pagination for displaying notes.
* **Modal Popup**: Creating a modal component for delete confirmation and integrating it with backend logic to delete notes.
* **Success and Error Messages**: Implementing mechanisms to display success and error messages to users based on their actions.

1. **Project setup step by step**

* Created the next js project using npx create-next-app@latest nextjs\_taskand initialize the project.
* **npm run dev** to run the project.
* installed the needed dependencies using npm install <dependency\_name> such as @prisma/client, prisma, mongodb, axios, react-icons, react-paginate and tailwindcss.
* Configured the tailwindcss inside the global.css.
* Make sure prisma and prisma client are installeed and run npx prisma init to initialize the prisma to our application.
* The above code automatically created the .env file and prisma folder.
* After that we have to configure mongodb and create username and password. No need to create a database. Just copy the cluster URL and paste it into .env file and enter the database name end of the URL.
* Inside the Prisma folder we have schema.prisma file and that includes database provider and database URL.
* Then add model inside the schema file. The model will be the collection name and it will include all collection fields. (Id, title, content, …)
* Then run npx prisma to generate Prisma client from schema.prisma and .env file.
* Pushed the schema filed to mongodb using npx prisma db push.
* Created the API end points using Prisma client.
* Once completed the CRUD API end point then started to implement the user interface.
* In-between used the version control (GitHub) to push the code.

1. **Deployment**

The application is deployed to <https://nextjstask-d186a00d901d.herokuapp.com/notes>. I have used the Heroku platform to deploy the next-js project. It will provide CI pipeline whenever push to GitHub it will automatically deploy.