### **Journal Application Setup Documentation (Next.js)**

This documentation will guide you through the steps to set up and run the Journal application locally. It includes instructions for environment setup, dependencies and necessary configurations.

### **1. Prerequisites**

Before setting up the project, ensure you have the following tools installed:

* **Node.js** (version 16 or higher): The backend runs on Node.js, so make sure it's installed. You can download it from [here](https://nodejs.org/).
* **npm** (comes with Node.js): npm is used to install project dependencies.
* **Git**: For version control and cloning the repository.

To verify your installations, run the following commands:

bash

Copy

node -v

npm -v

git --version

### **2. Cloning the Repository**

Start by cloning the project repository from GitHub:

bash

Copy

git clone <https://github.com/TinaKareh/personal_journal.git>

cd personal\_journal

### **3. Installing Dependencies**

After cloning the repository, you need to install the necessary dependencies. Run the following command in the project directory:

bash

Copy

npm install

This will install all the required libraries and packages listed in the package.json file.

### **4. Environment Setup**

Create a .env.local file in the root of the project. This file will store environment-specific variables needed to run the application locally.

Here is an example .env.local setup:

env

Copy

NEXT\_PUBLIC\_API\_URL=http://localhost:3000/api

DATABASE\_URL=postgres://username:password@localhost:5432/journal\_app

JWT\_SECRET=your-secret-key

* **NEXT\_PUBLIC\_API\_URL**: The API base URL. By default, it points to the local server.
* **DATABASE\_URL**: Connection string for your PostgreSQL database (or other databases based on your setup).
* **JWT\_SECRET**: A secret key for signing JWT tokens. You should generate a secure, random string for this.

### **5. Setting Up the Database**

The application uses MySQL by default for storing user and journal data. If you're using MySQL locally, make sure you have it installed.

Once installed, create a new database:

bash

Copy

CREATE DATABASE personal\_journal;

Then, apply any migrations you should use a migration tool like Prisma:

bash

Copy

npx prisma migrate dev

If you don't have Prisma set up, follow the setup instructions for Prisma (or your preferred ORM tool) to handle migrations and schema definitions.

### **6. Running the Application Locally**

Once everything is set up, you can start the development server. Run the following command in the project directory:

bash

Copy

npm run dev

This will start the application at http://localhost:3000. You can visit this URL in your browser to see the Journal application running locally.

### **7. Additional Configuration (Optional)**

Depending on the features you need to enable or modify, you might need to configure additional services:

* **CORS Configuration**: If you’re testing API routes separately, you may need to configure CORS (Cross-Origin Resource Sharing) in your Next.js API routes.

### **8. Testing and Debugging**

For local testing, you can use the built-in Next.js features to test the application. Some debugging tips:

* Use the browser’s developer tools (F12) to view the console and network logs.
* Check the terminal where you started the server for any errors or warnings.

### **9. Stopping the Application**

To stop the development server, press Ctrl + C in the terminal.