Cursor Essential Shortcuts (AI-Powered Editor)

- `Ctrl + ~` – Open/close terminal
- `Cmd + B` – Toggle sidebar/file explorer
- `Cmd + L` – Open AI chat panel
- `Cmd + K` – Show command palette
- `Cmd + K` **then highlight code** – Get AI help for highlighted code
- `Option + Z` – Toggle word wrap
- `Cmd + S` – Save file
- `Cmd + W` – Close tab
- `Cmd + Shift + P` – Command palette (alternate)

Next.js Essential File Structure for Students

Frontend Basics

- `page.tsx` – Creates a page in Next.js (e.g., `app/home/page.tsx` → `/home` route).
- `layout.tsx` – Wraps pages with shared UI (e.g., navbars, footers).
- `loading.tsx` – Displays a loading UI for a page.
- `error.tsx` – Handles errors for a specific page.
- `global.css` **or** `tailwind.css` – Global styles for the app.

Routing & API Basics

- `route.ts` – Defines API routes (e.g., `app/api/user/route.ts` → `/api/user`).
- **Dynamic Routes** – Use `[id]` or `[slug]` in folder names (e.g., `app/blog/[id]/page.tsx` → `/blog/123`).

- **API Handlers** – Use `GET`, `POST`, etc., inside `route.ts` for backend logic.

Backend Basics

- **`server.ts` (or `lib/prisma.ts`)** – Set up a database connection.
- **Middleware (`middleware.ts`)** – Runs code before requests hit routes (e.g., auth checks).
- **Env Variables (`.env.local`)** – Store secrets like API keys (`process.env.SECRET_KEY`).
- **Git Ignore (`.gitignore`)** – Specifies files and directories that Git should ignore (e.g., `node_modules/`, `.env.local`).

Understanding File Paths in Next.js

Next.js follows a structured file system where directories define application routes and component organization. Here’s a breakdown of common file structures:

- **Root Level (`app/`)** – The main directory for all Next.js pages and API routes.
- **Pages (`app/[route]/page.tsx`)** – Defines actual routes in the app (e.g., `app/dashboard/page.tsx` → `/dashboard`).
- **Components (`app/components/[component].tsx`)** – Houses reusable UI components (not directly part of routing).
- **Layouts (`app/layout.tsx`)** – Provides structure across multiple pages (e.g., global navigation).
- **API Routes (`app/api/[endpoint]/route.ts`)** – Defines backend endpoints (e.g., `app/api/user/route.ts` → `/api/user`).
- **Dynamic Routing (`app/[id]/page.tsx`)** – Uses square brackets to define routes that accept parameters (`/user/123`).
- **Middleware (`middleware.ts`)** – Runs before requests reach the handler, often used for authentication.
- **Static Assets (`public/`)** – Stores images, fonts, and other static files.
- **Environment Variables (`.env.local`)** – Stores sensitive information like API keys, database credentials, and tokens. These variables are accessed using `process.env.VARIABLE_NAME`. This file should always be added to `.gitignore` to avoid exposing secrets.

- **Git Ignore (`.gitignore`)** – A file that lists patterns for files and directories that should not be tracked by Git. Common entries include:
  - `node_modules/` (dependency files)
  - `.env.local` (environment variables)
  - `.next/` (build output)
  - `dist/` (compiled files)

Example Next.js Project Structure:

```
my-nextjs-project/
│── app/
│   ├── layout.tsx
│   ├── page.tsx
│   ├── dashboard/
│   │   ├── page.tsx
│   │   ├── settings/
│   │   │   ├── page.tsx
│   ├── api/
│   │   ├── user/
│   │   │   ├── route.ts
│── components/
│   ├── Navbar.tsx
│   ├── Footer.tsx
│── public/
│   ├── logo.png
│── styles/
│   ├── global.css
│── middleware.ts
│── .env.local
│── .gitignore
│── package.json
│── tsconfig.json
│── next.config.js
```

This structure ensures clean separation of concerns and improves maintainability.

Simple Git Commands to Save and Push Commits

- `git status` – Check the current state of the working directory
- `git add .` – Stage all changes
- `git status` – Verify that changes are staged (should appear green)
- `git commit -m "Your commit message"` – Commit changes with a message
- `git push -u origin main` – Push changes to GitHub
- `git pull origin main` – Pull latest changes from GitHub
- `git init` – Initialize a new Git repository
- `git branch -M main` – Rename branch to `main` (if needed)
- `git remote add origin <repository-url>` – Link local repo to a remote repository
