

Chapter 1: Getting Started with Next.js

- **Install pnpm:**

```
bash
Copy code
npm install -g pnpm
```

- **Create Project:**

```
bash
Copy code
npx create-next-app@latest nextjs-dashboard --example
"https://github.com/vercel/next-learn/tree/main/dashboard/starter-
example" --use-pnpm
```

Project Structure:

- `/app`: Main app logic and routes.
- `/app/lib`: Reusable functions.
- `/app/ui`: UI components.
- `/public`: Static assets.

Data:

- Placeholder data in `/app/lib/placeholder-data.ts`.

TypeScript:

- Types defined in `/app/lib/definitions.ts`.

Run Project:

```
bash
Copy code
pnpm i
pnpm dev
```

- Visit <http://localhost:3000>.

Chapter 2: CSS Styling

- **Global Styles:** Use `/app/ui/global.css`.
- **Tailwind CSS:** Utility-first classes.

- **CSS Modules:** Scoped styles in `/app/ui/home.module.css`.
- **clsx:** Conditional styling.

Chapter 3: Optimizing Fonts and Images

- **Fonts:** Use `next/font` to preload fonts.
- **Images:** Use `next/image` for optimized images (responsive, lazy-loaded)

Chapter 4: Creating Layouts and Pages

- **Routing:** Use folder structure for routes.
- **Layouts:** Share UI across pages with nested layouts (`/dashboard/layout.tsx`).
- **Root Layout:** `/app/layout.tsx` for global settings.

Chapter 5: Navigating Between Pages

- **<Link> Component:** Enables smooth client-side navigation.
- **Optimized Navigation:** Code splitting, prefetching.

Chapter 6: Setting Up Your Databases

- **Push to GitHub.**
- **Vercel:** Deploy and connect PostgreSQL.
- **Seed Database:** Use `/app/seed` to populate the database.

Chapter 7: Fetching Data

- **Data Fetching:** Use APIs, ORMs, or SQL for backend access.
- **Server Components:** Efficient data fetching without client-side libraries.

Chapter 8: Static and Dynamic Rendering

- **Static Rendering:** Fast load times, ideal for static content.
- **Dynamic Rendering:** Real-time data, personalized content.

Chapter 9: Streaming

- Send web pages in chunks for faster load times.
- Implement with `loading.tsx` for fallback UI.

Chapter 10: Partial Prerendering (PPR)

- Mix static and dynamic rendering on the same page.
- Static shell with dynamic content ("holes").

Chapter 11: Enhancing the Invoices Page

- Implement search, pagination, and URL handling.
- Sync search input with URL using `useSearchParams`.

Chapter 12: Mutating Data

- Use Server Actions to create, update, and delete data.
- Handle forms efficiently with Server Actions and `FormData`.

Chapter 13: Handling Errors

- Use `try/catch` in Server Actions.
- Manage 404 errors with `notFound()` and custom error UI.

Chapter 14: Improving Accessibility

- Follow best practices for accessible design and form validation.
- Use `Zod` for server-side validation.

Chapter 15: Adding Authentication

- Integrate `NextAuth.js` for authentication and authorization.
- Protect routes and handle password hashing.

Chapter 16: Adding Metadata

- Improve SEO and social media shareability with metadata.
- Set dynamic titles and descriptions in `layout.tsx`.

Chapter 17: Next Steps

- Explore `Next.js` documentation and templates.
- Share your project on social media for feedback.