Next.js Full Stack Project Structure

Complete app with Prisma, NextAuth, and API routes. Full-stack TypeScript architecture.

#nextjs #react #typescript #prisma #fullstack #auth

PNG PDF Copy Prompt

Project Directory

```
my-app/
  app/
    layout.tsx
    page.tsx
    globals.css
    (auth)/  Auth pages group
      layout.tsx
      login/
        page.tsx
      register/
        page.tsx
    (dashboard)/  Protected routes
      layout.tsx  Auth check here
      dashboard/
        page.tsx
      settings/
        page.tsx
    api/
      auth/
        [...nextauth]/  NextAuth handler
          route.ts
      users/
        route.ts  GET/POST users
        [id]/
          route.ts
      posts/
        route.ts
        [id]/
          route.ts
  components/
    ui/
      button.tsx
      input.tsx
      card.tsx
      modal.tsx
    forms/
      login-form.tsx
      register-form.tsx
    layout/
      header.tsx
      sidebar.tsx
      footer.tsx
    providers/  React context p…
      auth-provider.tsx
      theme-provider.tsx
  lib/
    prisma.ts  Prisma client s…
    auth.ts  NextAuth config
    utils.ts
    validations.ts  Zod schemas
  prisma/  Database layer
    schema.prisma  Database schema
    migrations/
    seed.ts  Seed data script
  actions/  Server Actions
    auth.ts  Auth server act…
    posts.ts  CRUD server act…
  types/
    index.ts
    next-auth.d.ts  NextAuth type a…
  hooks/
    use-auth.ts
    use-toast.ts
  public/
    favicon.ico
    images/
  middleware.ts  Auth middleware
  next.config.js
  package.json
  tsconfig.json
  .env  DATABASE_URL, N…
  .env.example
  .gitignore
```

Why This Structure?

This structure is a complete full-stack application scaffold. Prisma handles the database layer, NextAuth provides authentication, and Server Actions enable type-safe mutations. Route groups separate auth pages from protected dashboard routes, each with their own layouts.

Key Directories

app/(dashboard)/ - Protected routes with auth-checking layout

prisma/ - Database schema, migrations, and seed script

actions/ - Server Actions for mutations

lib/auth.ts - NextAuth configuration

components/providers/ - React context providers

Authentication Setup

middleware.ts - Protects routes before they render

lib/auth.ts - NextAuth config with providers

app/api/auth/[...nextauth]/ - Auth API endpoints

components/providers/auth-provider.tsx - Session context for client

Database Workflow

```
# Edit prisma/schema.prisma
npx prisma migrate dev --name your_migration
npx prisma generate
npx prisma db seed # optional
```

Getting Started

```
npx create-next-app@latest my-app
npm install prisma @prisma/client next-auth zod
npx prisma init
# Configure DATABASE_URL in .env
npx prisma migrate dev
npm run dev
```

When To Use This

- SaaS applications and dashboards
- Apps requiring user authentication
- Projects with database-backed features
- Full-stack TypeScript teams
- Rapid MVP development with auth/DB

Trade-offs

More dependencies - Prisma, NextAuth, Zod add to bundle

Configuration overhead - Auth and DB require initial setup

Hosting requirements - Needs database and persistent hosting