

Next.js SaaS Starter Project Structure

Complete SaaS boilerplate with auth, Stripe billing, and team management.

#nextjs #react #typescript #saas #stripe #auth

PNG

PDF

Copy

Prompt

Project Directory

my-saas/

```
> app/
  layout.tsx
  page.tsx Marketing landi...
  globals.css
  (marketing)/ Public marketin...
    layout.tsx
    pricing/
      page.tsx
    features/
      page.tsx
  (auth)/ Auth flow pages
    layout.tsx
    login/
      page.tsx
    signup/
      page.tsx
    forgot-password/
      page.tsx
  (app)/ Authenticated a...
    layout.tsx App shell with ...
    dashboard/
      page.tsx
    settings/
      page.tsx
      billing/
        page.tsx
      team/
        page.tsx
  api/
    auth/
      [...nextauth]/
        route.ts
      webhooks/ External servic...
        stripe/
          route.ts
  lib/
    auth.ts
    prisma.ts
    stripe.ts Stripe client a...
  prisma/
    schema.prisma User, Team, Sub...
    migrations/
  components/
    ui/
      button.tsx
      card.tsx
    billing/
      pricing-card.tsx
      checkout-button.tsx
    dashboard/
      sidebar.tsx
      stats-card.tsx
  middleware.ts
  next.config.js
  package.json
  tsconfig.json
  .env.local DB, auth, Strip...
  .gitignore
```

Why This Structure?

A production-ready SaaS starter with the core features every SaaS needs: authentication, subscription billing with Stripe, team/org management, and protected app routes. Route groups separate marketing, auth, and app sections with distinct layouts.

Key Directories

app/(marketing)/ - Public pages: landing, pricing, features

app/(app)/ - Protected app with dashboard and settings

app/api/webhooks/stripe/ - Stripe webhook handler

lib/stripe.ts - Stripe client, checkout, portal helpers

components/billing/ - Pricing cards, checkout buttons

Billing Integration

components/billing/checkout-button.tsx - Creates Stripe Checkout session

app/api/webhooks/stripe/route.ts - Handles subscription events

app/(app)/settings/billing/ - Customer portal link

prisma/schema.prisma - Subscription model with status

When To Use This

- New SaaS products with subscription billing
- B2B apps with team/organization features
- Projects needing auth + payments from day one
- MVPs that need to monetize quickly

Trade-offs

Opinionated stack - Prisma + Stripe + NextAuth baked in

Complexity - More moving parts than simpler starters

Stripe dependency - Tightly coupled to Stripe billing