

# Configuration Files - Complete Source Code & Documentation

## Table of Contents

- 1. [Package Configuration](#)
- 2. [Next.js Configuration](#)
- 3. [TypeScript Configuration](#)
- 4. [Tailwind Configuration](#)
- 5. [PostCSS Configuration](#)
- 6. [ESLint Configuration](#)
- 7. [Prisma Configuration](#)

## Package Configuration

package.json

**Purpose:** Defines project dependencies, scripts, and metadata **Location:** `/package.json`

```
{
  "name": "wine-store",
  "version": "0.1.0",
  "private": true,
  "scripts": {
    "dev": "next dev --turbo",
    "build": "next build",
    "start": "next start",
    "lint": "next lint",
    "seed": "node prisma/seed.js",
    "postinstall": "prisma generate"
  },
  "dependencies": {
    "@clerk/nextjs": "^6.25.5",
    "@prisma/client": "^6.8.2",
    "@radix-ui/react-checkbox": "^1.3.2",
    "@radix-ui/react-dialog": "^1.1.14",
    "@radix-ui/react-dropdown-menu": "^2.1.15",
    "@radix-ui/react-icons": "^1.3.2",
    "@radix-ui/react-label": "^2.1.7",
    "@radix-ui/react-popover": "^1.1.14",
    "@radix-ui/react-select": "^2.2.5",
    "@radix-ui/react-separator": "^1.1.7",
    "@radix-ui/react-slot": "^1.2.3",
    "@stripe/react-stripe-js": "^3.8.1",
    "@stripe/stripe-js": "^7.7.0",
    "axios": "^1.11.0",
    "class-variance-authority": "^0.7.1",
    "clsx": "^2.1.1",
    "embla-carousel-react": "^8.6.0",
    "lodash": "^4.17.21",
    "lucide-react": "^0.503.0",
    "next": "15.3.1",
    "next-themes": "^0.4.6",
    "react": "^19.0.0",
    "react-dom": "^19.0.0",
    "react-icons": "^5.5.0",
    "sonner": "^2.0.6",
    "stripe": "^18.4.0",
    "tailwind-merge": "^3.2.0",
    "use-debounce": "^10.0.4",
    "vaul": "^1.1.2"
  }
}
```

```
  },
  "devDependencies": {
    "@eslint/eslintrc": "^3",
    "@tailwindcss/postcss": "^4",
    "@types/lodash": "4.17.17",
    "@types/node": "20.17.48",
    "@types/react": "19",
    "@types/react-dom": "19",
    "csv-parse": "5.6.0",
    "eslint": "9",
    "eslint-config-next": "15.3.1",
    "prisma": "6.8.2",
    "tailwindcss": "4",
    "tw-animate-css": "1.2.8",
    "typescript": "5.8.3"
  }
}
```

Key Scripts Explained:

- **dev**: Starts development server with Turbopack for faster builds
- **build**: Creates production build
- **start**: Runs production server
- **seed**: Populates database with sample data
- **postinstall**: Automatically generates Prisma client after npm install

Major Dependencies:

- **@clerk/nextjs**: Authentication and user management
- **@prisma/client**: Database ORM client
- **@radix-ui**: Headless UI primitives for accessible components
- **@stripe**: Payment processing
- **next-themes**: Dark/light mode support
- **tailwind-merge**: Utility for merging Tailwind classes

Next.js Configuration

next.config.ts

**Purpose:** Next.js framework configuration **Location:** /next.config.ts

```
import type { NextConfig } from "next";

const nextConfig: NextConfig = {
  experimental: {
    turbo: {
      rules: {
        "*.svg": {
          loaders: ["@svgr/webpack"],
          as: "*.js",
        },
      },
    },
  },
  images: {
    remotePatterns: [
      {
        protocol: "https",
        hostname: "images.unsplash.com",
      },
      {
        protocol: "https",
        hostname: "plus.unsplash.com",
      },
    ],
  },
}
```

```
    },
  ],
},
};

export default nextConfig;
```

Configuration Explained:

- **Turbo Rules:** Configures SVG handling with SVGR webpack loader
- **Image Domains:** Allows images from Unsplash for remote image optimization
- **Experimental Features:** Enables Next.js 15 experimental features

## TypeScript Configuration

tsconfig.json

**Purpose:** TypeScript compiler configuration **Location:** /tsconfig.json

```
{
  "compilerOptions": {
    "lib": ["dom", "dom.iterable", "es6"],
    "allowJs": true,
    "skipLibCheck": true,
    "strict": true,
    "noEmit": true,
    "esModuleInterop": true,
    "module": "esnext",
    "moduleResolution": "bundler",
    "resolveJsonModule": true,
    "isolatedModules": true,
    "jsx": "preserve",
    "incremental": true,
    "plugins": [
      {
        "name": "next"
      }
    ],
    "baseUrl": ".",
    "paths": {
      "@/*": ["./*"]
    }
  },
  "include": ["next-env.d.ts", "**/*.ts", "**/*.tsx", ".next/types/**/*.ts"],
  "exclude": ["node_modules"]
}
```

Key Settings Explained:

- **baseUrl:** Sets root directory for imports
- **paths:** Maps @/\* to root directory for clean imports
- **strict:** Enables strict type checking
- **jsx:** Preserves JSX for Next.js processing
- **moduleResolution:** Uses bundler for modern module resolution

## Tailwind Configuration

tailwind.config.mjs

**Purpose:** Tailwind CSS configuration and customization **Location:** /tailwind.config.mjs

```
import { fontFamily } from "tailwindcss/defaultTheme";

/** @type {import('tailwindcss').Config} */
const config = {
  darkMode: ["class"],
  content: [
    "./pages/**/*.{ts,tsx}",
    "./components/**/*.{ts,tsx}",
    "./app/**/*.{ts,tsx}",
    "./src/**/*.{ts,tsx}",
  ],
  prefix: "",
  theme: {
    container: {
      center: true,
      padding: "2rem",
      screens: {
        "2xl": "1400px",
      },
    },
    extend: {
      colors: {
        border: "hsl(var(--border))",
        input: "hsl(var(--input))",
        ring: "hsl(var(--ring))",
        background: "hsl(var(--background))",
        foreground: "hsl(var(--foreground))",
        primary: {
          DEFAULT: "hsl(var(--primary))",
          foreground: "hsl(var(--primary-foreground))",
        },
        secondary: {
          DEFAULT: "hsl(var(--secondary))",
          foreground: "hsl(var(--secondary-foreground))",
        },
        destructive: {
          DEFAULT: "hsl(var(--destructive))",
          foreground: "hsl(var(--destructive-foreground))",
        },
        muted: {
          DEFAULT: "hsl(var(--muted))",
          foreground: "hsl(var(--muted-foreground))",
        },
        accent: {
          DEFAULT: "hsl(var(--accent))",
          foreground: "hsl(var(--accent-foreground))",
        },
        popover: {
          DEFAULT: "hsl(var(--popover))",
          foreground: "hsl(var(--popover-foreground))",
        },
        card: {
          DEFAULT: "hsl(var(--card))",
          foreground: "hsl(var(--card-foreground))",
        },
      },
      borderRadius: {
        lg: "var(--radius)",
        md: "calc(var(--radius) - 2px)",
        sm: "calc(var(--radius) - 4px)",
      },
      fontFamily: {
        sans: ["var(--font-sans)", ...fontFamily.sans],
      },
      keyframes: {
```

```
      "accordion-down": {
        from: { height: "0" },
        to: { height: "var(--radix-accordion-content-height)" },
      },
      "accordion-up": {
        from: { height: "var(--radix-accordion-content-height)" },
        to: { height: "0" },
      },
    },
    animation: {
      "accordion-down": "accordion-down 0.2s ease-out",
      "accordion-up": "accordion-up 0.2s ease-out",
    },
  },
  plugins: [require("tailwindcss-animate")],
};

export default config;
```

Configuration Explained:

- **Dark Mode:** Enables class-based dark mode switching
- **Content Paths:** Specifies where Tailwind should look for classes
- **Custom Colors:** Defines CSS custom properties for theming
- **Custom Animations:** Adds accordion animations
- **Container:** Centers content with responsive padding
- **Plugins:** Includes animation utilities

PostCSS Configuration

postcss.config.mjs

**Purpose:** PostCSS processing configuration **Location:** /postcss.config.mjs

```
export default {
  plugins: {
    tailwindcss: {},
    autoprefixer: {},
  },
};
```

Configuration Explained:

- **tailwindcss:** Processes Tailwind CSS directives
- **autoprefixer:** Automatically adds vendor prefixes

ESLint Configuration

eslint.config.mjs

**Purpose:** Code linting and formatting rules **Location:** /eslint.config.mjs

```
import { dirname } from "path";
import { fileURLToPath } from "url";
import { FlatCompat } from "@eslint/eslintrc";

const __filename = fileURLToPath(import.meta.url);
const __dirname = dirname(__filename);

const compat = new FlatCompat({
```

```
    baseDirectory: __dirname,
  });

  const eslintConfig = [
    ...compat.extends("next/core-web-vitals"),
    {
      rules: {
        // Add custom rules here
      },
    },
  ];

  export default eslintConfig;
```

Configuration Explained:

- **Flat Config:** Uses new ESLint flat config format
- **Next.js Rules:** Extends Next.js recommended rules
- **Custom Rules:** Placeholder for project-specific linting rules

Prisma Configuration

prisma/schema.prisma

**Purpose:** Database schema definition and configuration **Location:** /prisma/schema.prisma

```
// This is your Prisma schema file,
// learn more about it in the docs: https://pris.ly/d/prisma-schema

generator client {
  provider = "prisma-client-js"
}

datasource db {
  provider = "postgresql"
  url      = env("DATABASE_URL")
}

model User {
  id          String   @id @default(cuid())
  email       String   @unique
  name        String?
  role        Role      @default(USER)
  createdAt   DateTime @default(now())
  updatedAt   DateTime @updatedAt

  // Relations
  orders      Order[]
  reviews     Review[]
  favorites    Favorite[]
}

model Product {
  id          String   @id @default(cuid())
  name        String
  description  String?
  price       Decimal  @db.Decimal(10, 2)
  type        String
  featured    Boolean  @default(false)
  images       Json     // Array of image URLs
  region       Json     // Object with name and country
  averageRating Decimal? @db.Decimal(3, 2)
  createdAt   DateTime @default(now())
  updatedAt   DateTime @updatedAt
```

```
// Relations
orderItems OrderItem[]
reviews      Review[]
favorites    Favorite[]
}

model Order {
  id          String    @id @default(cuid())
  userId      String
  status      OrderStatus @default(PENDING)
  total       Decimal    @db.Decimal(10, 2)
  createdAt   DateTime   @default(now())
  updatedAt   DateTime   @updatedAt

  // Relations
  user        User        @relation(fields: [userId], references: [id])
  items       OrderItem[]
}

model OrderItem {
  id          String    @id @default(cuid())
  orderId     String
  productId   String
  quantity    Int
  price       Decimal    @db.Decimal(10, 2)
  createdAt   DateTime   @default(now())

  // Relations
  order       Order    @relation(fields: [orderId], references: [id])
  product     Product  @relation(fields: [productId], references: [id])
}

model Review {
  id          String    @id @default(cuid())
  userId      String
  productId   String
  rating      Int
  comment     String?
  createdAt   DateTime   @default(now())

  // Relations
  user        User    @relation(fields: [userId], references: [id])
  product     Product  @relation(fields: [productId], references: [id])
}

model Favorite {
  id          String    @id @default(cuid())
  userId      String
  productId   String
  createdAt   DateTime   @default(now())

  // Relations
  user        User    @relation(fields: [userId], references: [id])
  product     Product  @relation(fields: [productId], references: [id])

  @@unique([userId, productId])
}

enum Role {
  USER
  ADMIN
}

enum OrderStatus {
  PENDING
}
```

```
    CONFIRMED
    SHIPPED
    DELIVERED
    CANCELLED
  }
```

Schema Explained:

- **User Model:** Stores user information and relationships
- **Product Model:** Wine product details with images and metadata
- **Order Models:** Handles order processing and line items
- **Review Model:** User product reviews and ratings
- **Favorite Model:** User favorite products
- **Enums:** Defines role and order status options
- **Relations:** Establishes database relationships between models

## Environment Configuration

.env.local (Template)

**Purpose:** Environment variables configuration **Location:** Create this file locally (not committed to git)

```
# Database
DATABASE_URL="postgresql://username:password@localhost:5432/wine_store"

# Clerk Authentication
NEXT_PUBLIC_CLERK_PUBLISHABLE_KEY=pk_test...
CLERK_SECRET_KEY=sk_test...

# Stripe
STRIPE_SECRET_KEY=sk_test...
NEXT_PUBLIC_STRIPE_PUBLISHABLE_KEY=pk_test...

# App
NEXTAUTH_SECRET=your-secret-here
NEXTAUTH_URL=http://localhost:3000
```

Environment Variables Explained:

- **DATABASE\_URL:** PostgreSQL connection string
- **CLERK\_KEYS:** Authentication service credentials
- **STRIPE\_KEYS:** Payment processing credentials
- **NEXTAUTH:** NextAuth.js configuration (if used)

These configuration files provide the foundation for the Wine Store application. Each file serves a specific purpose in the development, build, and runtime processes.