

Project Template ? Next.js + TypeScript + Tailwind v4 (config-less) + ESLint (flat) + Vitest + Copilot

A reusable starter you can clone for new apps. Batteries included:

- Next.js (App Router), TypeScript, **Tailwind v4** (no 'tailwind.config.*')
- ESLint (flat config) + Prettier wired to VS Code on save
- Vitest + React Testing Library (+ providers helper)
- VS Code workspace files ('.vscode')
- Copilot prompt playbook ('.copilot/prompts')
- Dual dev servers: **Turbopack** (fast) and **Webpack** (fallback)

Requirements

- Node LTS (? 18)
- pnpm (? 8)

> Tip: keep pnpm in sync

> 'corepack enable && corepack prepare pnpm@latest --activate'

Getting Started

```
““bash
pnpm install
pnpm dev      # Turbopack (fast)
# or:
pnpm dev:web  # Webpack (fallback)
““
```

Open <http://localhost:3000>

Scripts

```
““json
{
  "dev": "next dev --turbopack",
  "dev:web": "next dev --no-turbo",
  "build": "next build --turbopack",
  "start": "next start",
  "lint": "eslint .",
  "lint:fix": "eslint . --fix",
  "test": "vitest",
}
```

```
"test:run": "vitest run",
"test:coverage": "vitest run --coverage"
}
```

VS Code

This repo includes workspace files so everyone shares the same tooling:

- `**.vscode/extensions.json` ? recommended extensions (Copilot, ESLint, Prettier, Tailwind CSS IntelliSense, Vitest Explorer, GitLens, etc.)
- `**.vscode/settings.json` ? use project TS SDK, run ESLint fixes + organize imports on save, Tailwind IntelliSense (incl. `'cva()' / 'tv()'` regex)
- `**.vscode/tasks.json` ? run `'dev'`, `'dev:web'`, `'build'`, `'start'` from the Task Runner
- `**.vscode/launch.json` ? debug `**Next.js dev**` and `**Vitest current file**`

Tailwind v4 (config-less)

- This template uses `**Tailwind v4**` in `**zero-config**` mode ? there is `**no**` `'tailwind.config.*'` file.
- Tailwind v4 directives are already present in `'src/app/globals.css'`.
- If you later need deep customization, you can add a `'tailwind.config.*'` file; it's not required for most apps.

> We intentionally `**do not**` include `'eslint-plugin-tailwindcss'` because current releases conflict with Tailwind v4 internals. VS Code's Tailwind IntelliSense still provides class completion and validation.

Linting & Formatting

- Flat config: `'eslint.config.mjs'`
- Prettier: `'prettierrc'`

On save, VS Code:

- Formats with Prettier
- Applies ESLint fixes and organizes imports

Run manually:

```
“bash
pnpm run lint
```

```
pnpm run lint:fix
```

```
““
```

```
---
```

Testing

- Runner: **Vitest** (JSDOM)
- Config: 'vitest.config.ts' (uses '@vitejs/plugin-react' + 'vite-tsconfig-paths')
- Global setup: 'src/test/setup.ts'
- Helper: 'src/test/renderWithProviders.tsx'
- Example: 'src/app/__tests__/page.test.tsx'

First run:

```
““bash
pnpm test
# or non-watch:
pnpm test:run
# with coverage:
pnpm test:coverage
““
```

Notes

- JSX in tests works via '@vitejs/plugin-react' (no need to 'import React').
- 'next-themes' is mocked in 'src/test/setup.ts' to avoid hook-context errors during unit tests.

```
---
```

Copilot Prompt Playbook

Reusable prompts live in **copilot/prompts/**:

- 'refactor-guardrails.md' ? safer refactors (types, a11y, Tailwind ordering)
- 'tests-vitest.md' ? generate RTL tests (AAA structure, a11y-first)
- 'commit-style.md' ? Conventional Commits

How to use

In Copilot Chat:

```
““
```

```
@workspace /open .copilot/prompts/refactor-guardrails.md
```

```
““
```

Click **Use as system prompt**, then ask for the refactor and apply the diff.

Inline Chat (**Cmd+Shift+I**) works great for small/local changes.

Project Structure

```
““txt
.vscode/          # workspace settings, tasks, launch, extension recs
.copilot/prompts/  # prompt playbook for Copilot
src/
  app/            # Next.js App Router
  app/globals.css  # Tailwind v4 directives live here (no tailwind.config.*)
  test/
    renderWithProviders.tsx
    setup.ts       # JSDOM, RTL cleanup, minimal mocks (e.g., next-themes)
eslint.config.mjs  # flat config (typed TS rules + React/a11y/imports)
vite.config.ts     # Vite + vite-tsconfig-paths + @vitejs/plugin-react
postcss.config.mjs # PostCSS pipeline (used by Tailwind v4)
tsconfig.json      # baseUrl + paths + test typings (vite/jest-dom)
““
```

2-Minute Smoke Test

1. **Dev server** ? ‘pnpm dev’ ? open the app.
2. **Format/Lint** ? add sloppy code and **Save** ? auto-format + ESLint fixes.
3. **Tests** ? ‘pnpm test’ ? example spec passes.
4. **Copilot** ? select a component ? **Cmd+Shift+I** ?
?Refactor into smaller components using ‘@workspace /open .copilot/prompts/refactor-guardrails.md’ ? ? apply diff.
5. **Tailwind IntelliSense** ? in ‘className=’ and inside ‘cva()’/‘tv()’, type ‘bg-’ and confirm suggestions.