

อธิบายโครงสร้างไฟล์ Nuxt 3 Project - watchman_app

โฟลเดอร์หลัก

1. `.nuxt/` (Auto-generated)

```
.nuxt/
├── tsconfig.json
├── types/
├── components.d.ts
└── ...
```

หน้าที:

- โพลเดอร์ที่ Nuxt สร้างขึ้นอัตโนมัติ
- เก็บไฟล์ที่ compile แล้วและ type definitions
- **ไม่ควรแก้ไข** ไฟล์ในนี้
- จะถูกสร้างใหม่ทุกครั้งที่เรา `bun dev`

2. `.vscode/`

```
.vscode/  
└─ settings.json
```

settings.json

```
json
{
  "files.associations": {
    "*.css": "tailwindcss"
  },
  "editor.quickSuggestions": {
    "strings": "on"
  },
  "tailwindCSS.classAttributes": ["class", "ui"],
  "tailwindCSS.experimental.classRegex": [
    ["ui:\\s*{([\\^]*}\\s*}", "(?:'|\\")([\\^]*)(?:'|\\")"]
  ]
}
```

หน้าที่:

- การตั้งค่า VS Code เฉพาะ project นี้
- เปิดใช้ Tailwind CSS IntelliSense
- Auto-completion สำหรับ `ui` prop ของ Nuxt UI

3. `assets/`

```
assets/  
└── css/  
    └── main.css
```

`main.css`

css

```
@import "tailwindcss";  
@import "@nuxt/ui-pro";  
  
/* Custom CSS Variables */  
:root {  
  --color-primary: rgb(59 130 246);  
  --color-secondary: rgb(71 85 105);  
}  
  
/* Custom Styles */  
.custom-gradient {  
  background: linear-gradient(135deg, var(--color-primary) 0%, var(--color-secondary) 100%);  
}  
  
/* Responsive breakpoints */  
@media (max-width: 640px) {  
  .mobile-hidden {  
    display: none;  
  }  
}
```

หน้าที่:

- เก็บ CSS, SCSS, images, fonts
- Import Tailwind CSS และ Nuxt UI Pro styles
- กำหนด CSS variables และ custom styles

- ไฟล์ในนี้จะถูก process โดย Nuxt

4. layouts/

layouts/

└─ default.vue

default.vue

vue

```
<template>
  <div class="min-h-screen bg-gray-50 dark:bg-gray-900">
    <!-- Header -->
    <UHeader>
      <template #logo>
        <UIcon name="i-lucide-shield-check" class="w-8 h-8" />
        <span class="ml-2 text-xl font-bold">{{ $config.public.appName }}</span>
      </template>

      <template #right>
        <UColorModeButton />
      </template>
    </UHeader>

    <!-- Main Content -->
    <main class="container mx-auto px-4 py-8">
      <slot />
    </main>

    <!-- Footer -->
    <UFooter>
      <template #left>
        <p class="text-sm text-gray-500">
          © 2025 {{ $config.public.appName }}. All rights reserved.
        </p>
      </template>
    </UFooter>
  </div>
</template>

<script setup>
const { $config } = useNuxtApp()
</script>
```

หน้าที่:

- กำหนดโครงสร้างหน้าเว็บทั้งหมด (Header, Footer, Navigation)
- `<slot />` คือที่ที่เนื้อหาจาก pages จะแสดง
- Layout นี้จะถูกใช้โดยทุกหน้าถ้าไม่กำหนดเป็นอื่น

5. `pages/`

pages/
└─ index.vue

index.vue


```

<template>
  <div class="space-y-8">
    <!-- Hero Section -->
    <UCard class="p-8">
      <div class="text-center">
        <UIcon name="i-lucide-shield-check" class="w-16 h-16 mx-auto mb-4 text-primary-500" />
        <h1 class="text-4xl font-bold mb-4">Welcome to {{ $config.public.appName }}</h1>
        <p class="text-lg text-gray-600 dark:text-gray-300 mb-6">
          A modern watchman application built with Nuxt 3 and UI Pro
        </p>
        <div class="flex justify-center gap-4">
          <UButton size="lg" icon="i-lucide-play">
            Get Started
          </UButton>
          <UButton size="lg" variant="outline" icon="i-lucide-book-open">
            Documentation
          </UButton>
        </div>
      </div>
    </UCard>
  </div>

```

```

    <!-- Features Grid -->
    <div class="grid grid-cols-1 md:grid-cols-3 gap-6">
      <UCard v-for="feature in features" :key="feature.title" class="p-6">
        <div class="text-center">
          <UIcon :name="feature.icon" class="w-12 h-12 mx-auto mb-4 text-primary-500" />
          <h3 class="text-xl font-semibold mb-2">{{ feature.title }}</h3>
          <p class="text-gray-600 dark:text-gray-300">{{ feature.description }}</p>
        </div>
      </UCard>
    </div>
  </div>
</template>

```

```

<script setup>
const { $config } = useNuxtApp()

// SEO
useHead({
  title: `Home - ${$config.public.appName}`,
})

```

```

// Features data

```

```
const features = [  
  {  
    title: 'Modern UI',  
    description: 'Built with Nuxt UI Pro components',  
    icon: 'i-lucide-palette'  
  },  
  // ... more features  
]  
</script>
```

หน้าที่:

- เป็นหน้าแรก (/) ของเว็บไซต์
- ใช้ Nuxt UI Pro components
- แสดง Hero section และ Features
- **File-based routing:** ชื่อไฟล์ = URL path

6. public/

```
public/  
├── favicon.ico  
└── og-image.jpg
```

หน้าที่:

- เก็บไฟล์ static ที่เข้าถึงได้โดยตรง
- `favicon.ico` - ไอคอนเว็บไซต์
- `og-image.jpg` - รูปสำหรับ social media sharing
- เข้าถึงได้ที่ root URL เช่น `/favicon.ico`

ไฟล์การตั้งค่า

7. .env

env

Nuxt UI Pro License Key

NUXT_UI_PRO_LICENSE=your_license_key_here

Public Environment Variables

PUBLIC_APP_NAME=Watchman App

PUBLIC_API_BASE_URL=http://localhost:3000

Private Environment Variables

DATABASE_URL=your_database_url_here

JWT_SECRET=your_jwt_secret_here

หน้าที่:

- เก็บค่าตัวแปรสภาพแวดล้อม
- ข้อมูลลับ (database, API keys)
- ค่าที่แตกต่างกันระหว่าง dev/prod

8. `.env.example`

env

Template สำหรับทีม

NUXT_UI_PRO_LICENSE=

PUBLIC_APP_NAME=Watchman App

PUBLIC_API_BASE_URL=

DATABASE_URL=

JWT_SECRET=

หน้าที่:

- Template สำหรับทีมงาน
- แสดงตัวแปรที่ต้องมี **ไม่มีค่าจริง**
- ถูก commit ใน Git

9. `.gitignore`

gitignore

Nuxt dev/build outputs

.output

.data

.nuxt

.nitro

.cache

dist

Node dependencies

node_modules

Environment files

.env

.env.local

.env.*.local

Bun lockfile

bun.lockb

OS generated files

.DS_Store

Thumbs.db

หน้าที่:

- กำหนดไฟล์ที่ไม่ต้อง commit ใน Git
- ป้องกัน sensitive data หลุด
- ไม่ commit auto-generated files

10. `app.config.ts`

typescript

```
export default defineAppConfig({
  ui: {
    colors: {
      primary: 'blue',
      secondary: 'slate',
      success: 'green',
      warning: 'yellow',
      error: 'red'
    },
    icons: {
      copy: 'i-lucide-copy',
      copyCheck: 'i-lucide-copy-check',
      close: 'i-lucide-x',
      menu: 'i-lucide-menu'
    }
  },

  uiPro: {
    prose: {
      codeIcon: {
        terminal: 'i-lucide-terminal',
        javascript: 'i-lucide-file-text',
        typescript: 'i-lucide-file-text'
      }
    }
  }
})
```

หน้าที่:

- การตั้งค่า Nuxt UI theme และ colors
- กำหนด default icons
- ตั้งค่า UI components behavior
- สามารถ HMR (Hot Module Reload) ได้

11. `app.vue`

vue

```
<template>
  <UApp>
    <NuxtLayout>
      <NuxtPage />
    </NuxtLayout>
  </UApp>
</template>

<script setup>
// Global meta tags
useHead({
  title: 'Watchman App',
  meta: [
    { name: 'description', content: 'A modern watchman application' },
    { name: 'viewport', content: 'width=device-width, initial-scale=1' }
  ],
  link: [
    { rel: 'icon', type: 'image/x-icon', href: '/favicon.ico' }
  ]
})

// Global SEO
useSeoMeta({
  title: 'Watchman App',
  ogTitle: 'Watchman App',
  description: 'A modern watchman application',
  ogDescription: 'A modern watchman application',
  ogImage: '/og-image.jpg',
  twitterCard: 'summary_large_image'
})
</script>
```

หน้าที่:

- **Root component** ของแอปพลิเคชัน
- `<UApp>` - จำเป็นสำหรับ Nuxt UI Pro
- `<NuxtLayout>` - เรียกใช้ layout
- `<NuxtPage>` - แสดง page ปัจจุบัน
- กำหนด global meta tags และ SEO

12. bun.lockb

หน้าที่:

- Lock file ของ Bun package manager
- เก็บ version ที่แน่นอนของ dependencies
- **Binary file** - ไม่สามารถอ่านได้
- ทำให้ทีมได้ dependencies version เดียวกัน

13. bunfig.toml

toml

[install]

Auto install peer dependencies

auto = true

Use exact versions

exact = true

Registry configuration

registry = "https://registry.npmjs.org/"

[run]

Enable hot reload

hot = true

[build]

Enable minification

minify = true

หน้าที่:

- การตั้งค่า Bun package manager
- กำหนดพฤติกรรมการติดตั้ง package
- เทียบเท่า .npmrc ของ npm

14. nuxt.config.ts

typescript

```
export default defineNuxtConfig({
  modules: ['@nuxt/ui-pro'],

  // เปิดใช้งาน future compatibility
  future: {
    compatibilityVersion: 4
  },

  // CSS Imports
  css: ['~/assets/css/main.css'],

  // DevTools และ Debug
  devtools: { enabled: true },

  // Environment variables
  runtimeConfig: {
    // Private keys (server-side only)
    nuxtUiProLicense: process.env.NUXT_UI_PRO_LICENSE,

    // Public keys (client + server)
    public: {
      appName: process.env.PUBLIC_APP_NAME || 'Watchman App',
      apiUrl: process.env.PUBLIC_API_BASE_URL || 'http://localhost:3000'
    }
  },

  // Nuxt UI Pro Configuration
  uiPro: {
    license: process.env.NUXT_UI_PRO_LICENSE
  },

  // UI Configuration
  ui: {
    theme: {
      colors: ['primary', 'secondary', 'success', 'warning', 'error'],
      transitions: true
    },
    fonts: true,
    colorMode: true
  }
})
```

หน้าที่:

- หัวใจหลักของ Nuxt configuration
- กำหนด modules ที่ใช้
- ตั้งค่า runtime config
- CSS imports และ build settings
- การตั้งค่า development tools

15. package.json

json

```
{
  "name": "watchman-app",
  "private": true,
  "type": "module",
  "scripts": {
    "build": "bun run nuxt build",
    "dev": "bun run nuxt dev",
    "generate": "bun run nuxt generate",
    "preview": "bun run nuxt preview",
    "postinstall": "bun run nuxt prepare",
    "typecheck": "bun run nuxt typecheck"
  },
  "dependencies": {
    "@iconify-json/lucide": "^1.2.18",
    "@nuxt/ui-pro": "^3.0.0",
    "nuxt": "^3.16.0"
  },
  "devDependencies": {
    "typescript": "^5.8.2",
    "vue-tsc": "^2.2.10"
  }
}
```

หน้าที่:

- ข้อมูลพื้นฐานของ project
- กำหนด scripts สำหรับรัน commands
- รายการ dependencies และ devDependencies
- เทียบเท่า "หนังสือพาสปอร์ต" ของ project

16. **README.md**

markdown

Watchman App

A modern watchman application built with Nuxt 3 and UI Pro.

Setup

```
```bash
```

```
bun install
```

## Development Server

```
bash
```

```
bun dev
```

## Production

```
bash
```

```
bun run build
```

**\*\*หน้าที่:\*\***

- เอกสารอธิบาย project
- วิธีการติดตั้งและใช้งาน
- ข้อมูลสำหรับ developers ใหม่

### 17. `tsconfig.json`

```
```json
```

```
{  
  "extends": "../.nuxt/tsconfig.json"  
}
```

หน้าที่:

- การตั้งค่า TypeScript
- Extends จาก Nuxt auto-generated config
- **สั้นมาก** เพราะ Nuxt จัดการให้แล้ว

Flow การทำงาน

1. เริ่มต้น: `bun dev` → อ่าน `nuxt.config.ts`
2. **Load Modules:** โหลด `@nuxt/ui-pro` และ modules อื่น
3. **Load Environment:** อ่านค่าจาก `.env`
4. **Generate Types:** สร้างไฟล์ใน `.nuxt/`
5. **Load App:** `app.vue` → `layouts/default.vue` → `pages/index.vue`
6. **Styling:** Load `assets/css/main.css` → Apply theme จาก `app.config.ts`
7. **Ready:** เว็บไซต์พร้อมใช้งาน

สรุปการใช้งาน

- **แก้ไขเนื้อหา:** แก้ในโฟลเดอร์ `pages/`
- **แก้ไข Layout:** แก้ในโฟลเดอร์ `layouts/`
- **แก้ไข Styling:** แก้ใน `assets/css/main.css` หรือ `app.config.ts`
- **เพิ่ม Dependencies:** `bun add package-name`
- **การตั้งค่า:** แก้ใน `nuxt.config.ts`
- **Environment Variables:** แก้ใน `.env`