

here's a **ready-to-use full folder structure** for your 5–page NestJS website with EJS and static assets. You can just copy-paste this setup.

Building a 5–6 page website using **NestJS** on a Windows PC is very doable. I'll walk you through it step-by-step—from setup to a working multi-page site.

□ 1. Prerequisites (Install first)

🔗 Install Node.js

- Download from: <https://nodejs.org>
- Verify:

```
node -v  
npm -v
```

🔗 Install NestJS CLI

```
npm install -g @nestjs/cli
```

```
D:\nestjs\2026>npm install -g @nestjs/cli  
  
added 7 packages, removed 39 packages, and changed 206 packages in 2m  
  
39 packages are looking for funding  
run `npm fund` for details
```

Check:

```
nest -v
```

□ 2. Create Your Project

```
nest new my-website
```

- Choose **npm**
- Move into project:

```
D:\nestjs\2026>npm install -g @nestjs/cli
added 7 packages, removed 39 packages, and changed 206 packages in 2m
39 packages are looking for funding
  run `npm fund` for details

D:\nestjs\2026>nest new my-website
👉 We will scaffold your app in a few seconds..

? Which package manager would you 🐱 to use?
> npm
  yarn
  pnpm

↑↓ navigate • 🐱 select
```

Run app:

```
cd my-website
```

```
npm run start
```

```
👉 Donate: https://opencollective.com/nest

D:\nestjs\2026>cd my-website
D:\nestjs\2026\my-website>npm run start
> my-website@0.0.1 start
> nest start

[Nest] 8956 - 04/03/2026, 12:39:52 PM LOG [NestFactory] Starting Nest application...
[Nest] 8956 - 04/03/2026, 12:39:52 PM LOG [InstanceLoader] AppModule dependencies initialized +12ms
[Nest] 8956 - 04/03/2026, 12:39:52 PM LOG [RoutesResolver] AppController {/}: +34ms
[Nest] 8956 - 04/03/2026, 12:39:52 PM LOG [RouterExplorer] Mapped {/, GET} route +3ms
[Nest] 8956 - 04/03/2026, 12:39:52 PM LOG [NestApplication] Nest application successfully started +3ms
```

Open in browser:

```
http://localhost:3000
```

3. Install Template Engine (for pages)

NestJS is backend-focused, so for pages you'll use a template engine like **EJS**:

```
npm install @nestjs/platform-express ejs
```

□ Full Project Structure

```
my-website/  
├── src/  
│   └── main.ts  
├── views/  
│   ├── home.ejs  
│   ├── about.ejs  
│   ├── services.ejs  
│   ├── blog.ejs  
│   ├── contact.ejs  
│   └── partials/  
│       ├── header.ejs  
│       └── footer.ejs  
├── public/  
│   └── css/  
│       └── style.css  
└── package.json
```

This PC > New Volume (D:) > nestjs > 2026 > my-website

	Name
✦	dist
✦	node_modules
✦	public
✦	src
✦	test
✦	views
✦	.gitignore
	.prettierrc
	eslint.config
	nest-cli
	package
	package-lock
	README
	tsconfig.build
	tsconfig

main.ts

```
import { NestFactory } from '@nestjs/core';
import { AppModule } from './app.module';
import { NestExpressApplication } from '@nestjs/platform-express';
import * as path from 'path';

async function bootstrap() {
  const app = await NestFactory.create<NestExpressApplication>(AppModule);

  app.useStaticAssets(path.join(__dirname, '..', 'public'));
  app.setBaseViewsDir(path.join(__dirname, '..', 'views'));
  app.setViewEngine('ejs');

  await app.listen(process.env.PORT ?? 3000);
}
bootstrap();
```

partials/header.ejs

```
<!DOCTYPE html>
<html>
<head>
  <title><%= title %></title>
  <link rel="stylesheet" href="/css/style.css">
</head>
<body>

<nav>
  <a href="/">Home</a> |
  <a href="/about">About</a> |
  <a href="/services">Services</a> |
  <a href="/blog">Blog</a> |
  <a href="/contact">Contact</a>
</nav>

<hr>
```

partials/footer.ejs

```
<hr>
<footer>
  <p>© 2026 My Website</p>
</footer>
</body>
</html>
```

 **home.ejs**

```
<%- include('partials/header', { title: 'Home' }) %>

<h1>Welcome to My Website</h1>
<p>This is the home page.</p>


<%- include('partials/footer') %>
```

 **about.ejs**

```
<%- include('partials/header', { title: 'About' }) %>

<h1>About Us</h1>
<p>We build awesome web applications with NestJS.</p>

<%- include('partials/footer') %>
```

 **services.ejs**

```
<%- include('partials/header', { title: 'Services' }) %>

<h1>Our Services</h1>
<ul>
  <li>Web Development</li>
  <li>Backend Development</li>
  <li>UI/UX Design</li>
  <li>SEO Optimization</li>
  <li>Maintenance & Support</li>
</ul>

<%- include('partials/footer') %>
```

 **blog.ejs**


```
<%- include('partials/header', { title: 'Blog' }) %>

<h1>Blog</h1>

<div>
  <h2>Getting Started with NestJS</h2>
  <p>Learn how to build scalable backend applications step by step.</p>
  <a href="#">Read More</a>
</div>

<div>
  <h2>Web Development Tips</h2>
  <p>Best practices to improve your workflow.</p>
  <a href="#">Read More</a>
</div>

<%- include('partials/footer') %>
```

 **contact.ejs**

```
<%- include('partials/header', { title: 'Contact' }) %>

<h1>Contact Us</h1>

<form method="POST" action="/contact">
  <input type="text" name="name" placeholder="Your Name"><br><br>
  <input type="email" name="email" placeholder="Your Email"><br><br>
  <textarea name="message" placeholder="Message"></textarea><br><br>
  <button type="submit">Send</button>
</form>

<%- include('partials/footer') %>
```

 **public/css/style.css**

```
body {
  font-family: Arial, sans-serif;
  margin: 0;
  padding: 0 20px;
  background: #f5f5f5;
}

nav {
  background: #333;
```

```
padding: 10px;
}

nav a {
  color: white;
  margin-right: 10px;
  text-decoration: none;
}

nav a:hover {
  text-decoration: underline;
}

h1 {
  color: #333;
}

div {
  background: white;
  padding: 15px;
  margin-bottom: 15px;
  border-radius: 10px;
  box-shadow: 0 0 5px rgba(0,0,0,0.1);
}
```

Ready to Run

1. Make sure you installed packages(`@nestjs/platform-express` is the **adapter that tells NestJS to use Express.**) :-

```
npm install @nestjs/platform-express ejs
```

2. Start server:

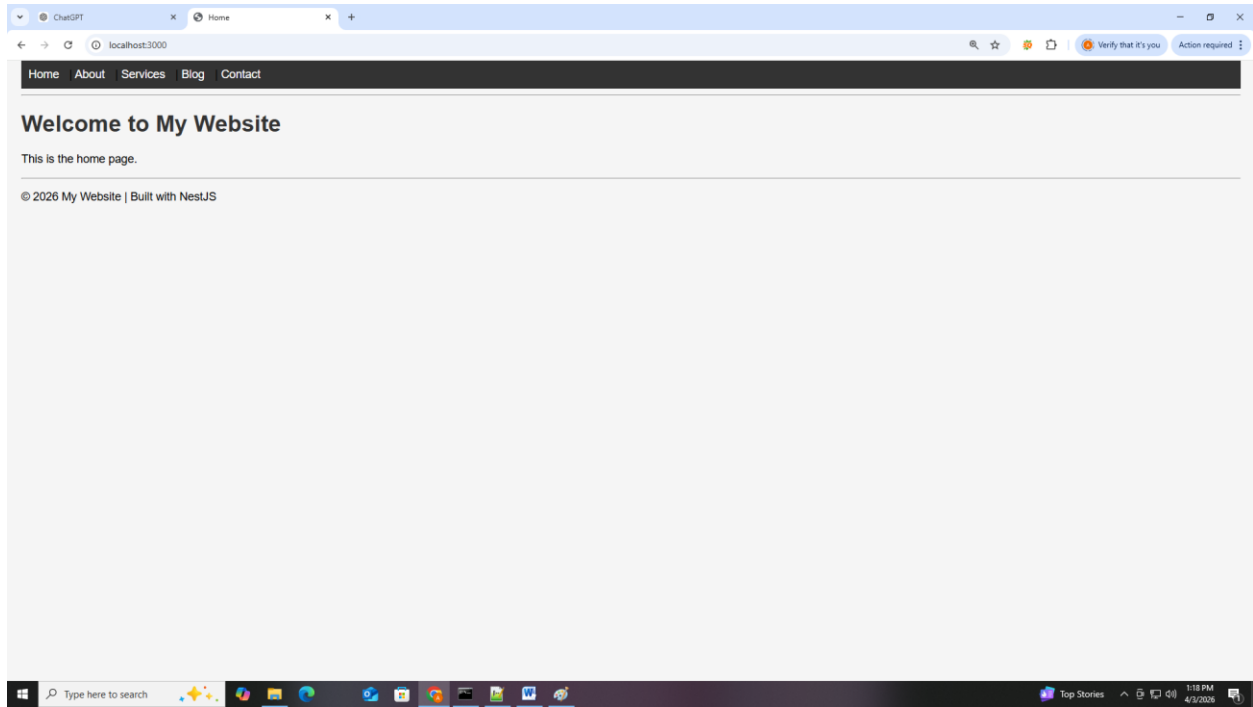
```
npm run start:dev
```

3. Open browser:

```
http://localhost:3000/
```

You now have a **fully working 5-page NestJS website** with header/footer, CSS, and static assets!

Output:-



Let's make a **step-by-step guide** for creating your **5-page NestJS website with EJS, header/footer, dynamic titles, and content** on a Windows PC.

Step 1: Install Node.js and NestJS CLI

1. Make sure **Node.js** is installed:

```
node -v
npm -v
```

2. Install NestJS CLI globally:

```
npm install -g @nestjs/cli
```

Step 2: Create a new NestJS project

```
nest new my-website
cd my-website
```

- Choose `npm` when asked for the package manager.
-

Step 3: Install required packages

```
npm install @nestjs/platform-express ejs
```

- `@nestjs/platform-express` → allows using `useStaticAssets`, `setViewEngine`
 - `ejs` → template engine
-

Step 4: Folder structure

Inside your project, create:

```
my-website/
├── src/
│   └── main.ts
├── views/
│   ├── home.ejs
│   ├── about.ejs
│   └── services.ejs
```



Step 5: Configure `main.ts`

```

import { NestFactory } from '@nestjs/core';
import { AppModule } from './app.module';
import { NestExpressApplication } from '@nestjs/platform-express';
import * as path from 'path';

async function bootstrap() {
  const app = await NestFactory.create<NestExpressApplication>(AppModule);

  app.useStaticAssets(path.join(__dirname, '..', 'public'));
  app.setBaseViewsDir(path.join(__dirname, '..', 'views'));
  app.setViewEngine('ejs');

  await app.listen(3000);
}
bootstrap();

```

1 Controller Setup (`app.controller.ts`)

We'll include `title` for header and any page-specific variables like `companyName` or `services`.

```

import { Controller, Get, Render } from '@nestjs/common';

@Controller()
export class AppController {

  @Get()
  @Render('home')
  home() {
    return { title: 'Home', companyName: 'My Website' };
  }

  @Get('about')
  @Render('about')
  about() {
    return { title: 'About', companyName: 'My Website' };
  }
}

```

```

}

@Get('contact')
@Render('contact')
contact() {
  return { title: 'Contact', companyName: 'My Website' };
}

@Get('services')
@Render('services')
services() {
  return {
    title: 'Services',
    companyName: 'My Website',
    servicesList: ['Web Development', 'Backend Development', 'UI/UX
Design', 'SEO Optimization', 'Maintenance & Support']
  };
}

@Get('blog')
@Render('blog')
blog() {
  return {
    title: 'Blog',
    companyName: 'My Website',
    blogPosts: [
      { title: 'Getting Started with NestJS', description: 'Learn how to
build scalable backend applications step by step.' },
      { title: 'Web Development Tips', description: 'Best practices to
improve your workflow.' }
    ]
  };
}
}

```

□ Explanation:

- title → used in header.ejs for <title>
- companyName → can be displayed anywhere in page
- servicesList → dynamic list for services.ejs
- blogPosts → dynamic blog content for blog.ejs

2 □ Header (views/partials/header.ejs)

```

<!DOCTYPE html>
<html>
<head>
  <title><%= title || 'My Website' %></title>
  <link rel="stylesheet" href="/css/style.css">
</head>

```

```
<body>

<nav>
  <a href="/">Home</a> |
  <a href="/about">About</a> |
  <a href="/services">Services</a> |
  <a href="/blog">Blog</a> |
  <a href="/contact">Contact</a>
</nav>

<hr>
```

- `title || 'My Website'` → ensures a default title if missing

3 Footer (`views/partials/footer.ejs`)

```
<hr>
<footer>
  <p>© 2026 <%= companyName %></p>
</footer>
</body>
</html>
```

- `companyName` must be passed from controller

4 Home Page (`views/home.ejs`)

```
<%- include('partials/header', { title }) %>

<h1>Welcome to <%= companyName %></h1>
<p>This is the home page.</p>

<%- include('partials/footer', { companyName }) %>
```

5 About Page (`views/about.ejs`)

```
<%- include('partials/header', { title }) %>

<h1>About <%= companyName %></h1>
<p>We build awesome web applications with NestJS.</p>

<%- include('partials/footer', { companyName }) %>
```

6 Contact Page (views/contact.ejs)

```
<%- include('partials/header', { title }) %>

<h1>Contact <%= companyName %></h1>

<form method="POST" action="/contact">
  <input type="text" name="name" placeholder="Your Name"><br><br>
  <input type="email" name="email" placeholder="Your Email"><br><br>
  <textarea name="message" placeholder="Message"></textarea><br><br>
  <button type="submit">Send</button>
</form>

<%- include('partials/footer', { companyName }) %>
```

7 Services Page (views/services.ejs)

```
<%- include('partials/header', { title }) %>

<h1>Our Services</h1>
<ul>
  <% servicesList.forEach(service => { %>
    <li><%= service %></li>
  <% }) %>
</ul>

<%- include('partials/footer', { companyName }) %>
```

8 Blog Page (views/blog.ejs)

```
<%- include('partials/header', { title }) %>

<h1>Blog</h1>

<% blogPosts.forEach(post => { %>
  <div>
    <h2><%= post.title %></h2>
    <p><%= post.description %></p>
  </div>
<% }) %>

<%- include('partials/footer', { companyName }) %>
```

Key Points

1. Always pass **required variables** to partials (header needs title, footer needs companyName)
2. Dynamic lists (services, blog) are rendered with `<% ... %>` loops
3. This setup **avoids 500 errors** and works for all 5 pages

Run your app

```
npm run start:dev
```

Output:-

