

Let's set up a **full NestJS + Handlebars project** step by step with **layouts, header/footer partials, and static assets**, ready for 2026. I'll guide you carefully.

Note:- Here Handlebars (HBS) as the view engine.

---

## Step 0 — Create a new NestJS project

```
npm i -g @nestjs/cli
nest new nest-hbs-app
cd nest-hbs-app
```

Choose **npm** as preferred.

```
D:\nestjs\2026>nest new nest-hbs-app
[?] We will scaffold your app in a few seconds..

√ Which package manager would you [?] to use? npm
CREATE nest-hbs-app/.prettierrc (56 bytes)
CREATE nest-hbs-app/eslint.config.mjs (934 bytes)
CREATE nest-hbs-app/nest-cli.json (179 bytes)
CREATE nest-hbs-app/package.json (2053 bytes)
CREATE nest-hbs-app/README.md (5126 bytes)
CREATE nest-hbs-app/tsconfig.build.json (101 bytes)
CREATE nest-hbs-app/tsconfig.json (702 bytes)
CREATE nest-hbs-app/src/app.controller.ts (286 bytes)
CREATE nest-hbs-app/src/app.module.ts (259 bytes)
CREATE nest-hbs-app/src/app.service.ts (150 bytes)
CREATE nest-hbs-app/src/main.ts (236 bytes)
CREATE nest-hbs-app/src/app.controller.spec.ts (639 bytes)
CREATE nest-hbs-app/test/jest-e2e.json (192 bytes)
CREATE nest-hbs-app/test/app.e2e-spec.ts (754 bytes)

▶■■■■ Installation in progress... [?]
```

```
D:\nestjs\2026>nest new nest-hbs-app
✔ We will scaffold your app in a few seconds..

/ Which package manager would you like to use? npm
CREATE nest-hbs-app/.prettierrc (56 bytes)
CREATE nest-hbs-app/eslint.config.mjs (934 bytes)
CREATE nest-hbs-app/nest-cli.json (179 bytes)
CREATE nest-hbs-app/package.json (2053 bytes)
CREATE nest-hbs-app/README.md (5126 bytes)
CREATE nest-hbs-app/tsconfig.build.json (101 bytes)
CREATE nest-hbs-app/tsconfig.json (702 bytes)
CREATE nest-hbs-app/src/app.controller.ts (286 bytes)
CREATE nest-hbs-app/src/app.module.ts (259 bytes)
CREATE nest-hbs-app/src/app.service.ts (150 bytes)
CREATE nest-hbs-app/src/main.ts (236 bytes)
CREATE nest-hbs-app/src/app.controller.spec.ts (639 bytes)
CREATE nest-hbs-app/test/jest-e2e.json (192 bytes)
CREATE nest-hbs-app/test/app.e2e-spec.ts (754 bytes)

/ Installation in progress... ✔

✔ Successfully created project nest-hbs-app
✔ Get started with the following commands:

$ cd nest-hbs-app
$ npm run start

Thanks for installing Nest ✔
Please consider donating to our open collective
to help us maintain this package.
```

---

## Step 1 — Install Handlebars engine

We'll use **express-handlebars** for layout and partial support:

```
npm install express-handlebars
```

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

---

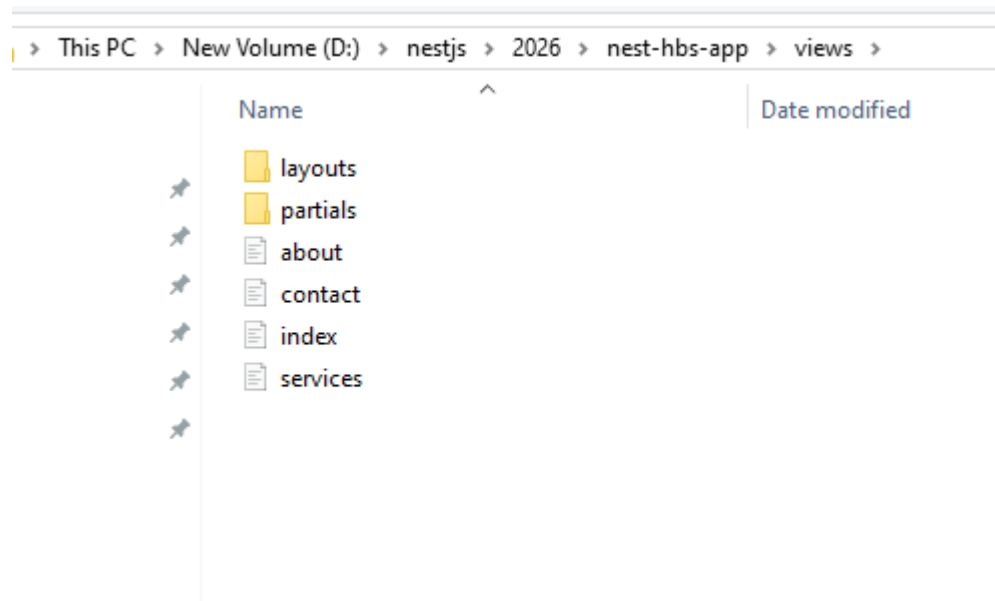
## Step 2 — Set up folder structure for views

Inside `nest-hbs-app`:

```
nest-hbs-app/  
  public/                <-- static assets like CSS/JS/images  
    style.css  
  views/  
    layouts/            <-- main layout with header/footer  
      main.hbs  
    partials/  
      header.hbs  
      footer.hbs  
    index.hbs  
    about.hbs  
    services.hbs  
    contact.hbs
```

---

As shown below



## Step 3 — Create partials

`views/partial/header.hbs`

```
<header>  
  <nav>  
    <a href="/">Home</a> |  
    <a href="/about">About</a> |  
    <a href="/services">Services</a> |
```

```
    <a href="/contact">Contact</a>
  </nav>
  <hr />
</header>
```

#### **views/partials/footer.hbs**

```
<hr />
<footer>
  <p>&copy; 2026 My NestJS Website</p>
</footer>
```

---

## **Step 4 — Create layout**

#### **views/layouts/main.hbs**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>{{title}}</title>
  <link rel="stylesheet" href="/style.css">
</head>
<body>
  {{> header}}

  {{{body}}} <!-- page content will be injected here -->

  {{> footer}}
</body>
</html>
```

---

## **Step 5 — Create pages**

#### **views/index.hbs**

```
<h1>{{title}}</h1>
<p>{{message}}</p>
```

### views/about.hbs

```
<h1>{{title}}</h1>
<p>{{description}}</p>
```

### views/services.hbs

```
<h1>{{title}}</h1>
<ul>
  {{#each services}}
    <li>{{this}}</li>
  {{/each}}
</ul>
```

### views/contact.hbs

```
<h1>{{title}}</h1>
<form action="/contact" method="post">
  <input type="text" name="name" placeholder="Your Name" required />
  <input type="email" name="email" placeholder="Your Email" required />
  <textarea name="message" placeholder="Message" required></textarea>
  <button type="submit">Send</button>
</form>
<p>{{feedback}}</p>
```

---

## Step 6 — main.ts

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

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

  // Serve static assets
  app.useStaticAssets(join(__dirname, '..', 'public'));

  // Set views directory
  app.setBaseViewsDir(join(__dirname, '..', 'views'));

  // Handlebars engine with layouts and partials
  app.engine(
    'hbs',
    exphbs.engine({
      extname: 'hbs',
      defaultLayout: 'main',
      layoutsDir: join(__dirname, '..', 'views/layouts'),
      partialsDir: join(__dirname, '..', 'views/partials'),
    }),
  ),
```

```
);  
  
// Set view engine  
app.setViewEngine('hbs');  
  
await app.listen(3000);  
console.log('Server running on http://localhost:3000');  
}  
bootstrap();
```

---

## Step 7 — AppController

### app.controller.ts

```
import { Controller, Get, Post, Body, Render } from '@nestjs/common';  
  
@Controller()  
export class AppController {  
  @Get()  
  @Render('index')  
  home() {  
    return { title: 'Home Page', message: 'Welcome to our NestJS site!' };  
  }  
  
  @Get('about')  
  @Render('about')  
  about() {  
    return { title: 'About Us', description: 'We provide amazing services.' };  
  };  
}  
  
  @Get('services')  
  @Render('services')  
  services() {  
    return { title: 'Our Services', services: ['Web Development', 'API  
Development', 'UI/UX Design'] };  
  }  
  
  @Get('contact')  
  @Render('contact')  
  contact() {  
    return { title: 'Contact Us', feedback: '' };  
  }  
  
  @Post('contact')  
  @Render('contact')  
  handleContact(@Body() body: any) {  
    const { name, email, message } = body;  
    // Here you could save to DB or send email  
    return { title: 'Contact Us', feedback: `Thanks ${name}, we received your  
message!` };  
  }  
}
```

---

## Step 8 — Add CSS

### public/style.css

```
body {
  font-family: Arial, sans-serif;
  margin: 20px;
}

header nav a {
  margin-right: 10px;
  text-decoration: none;
  color: #333;
}

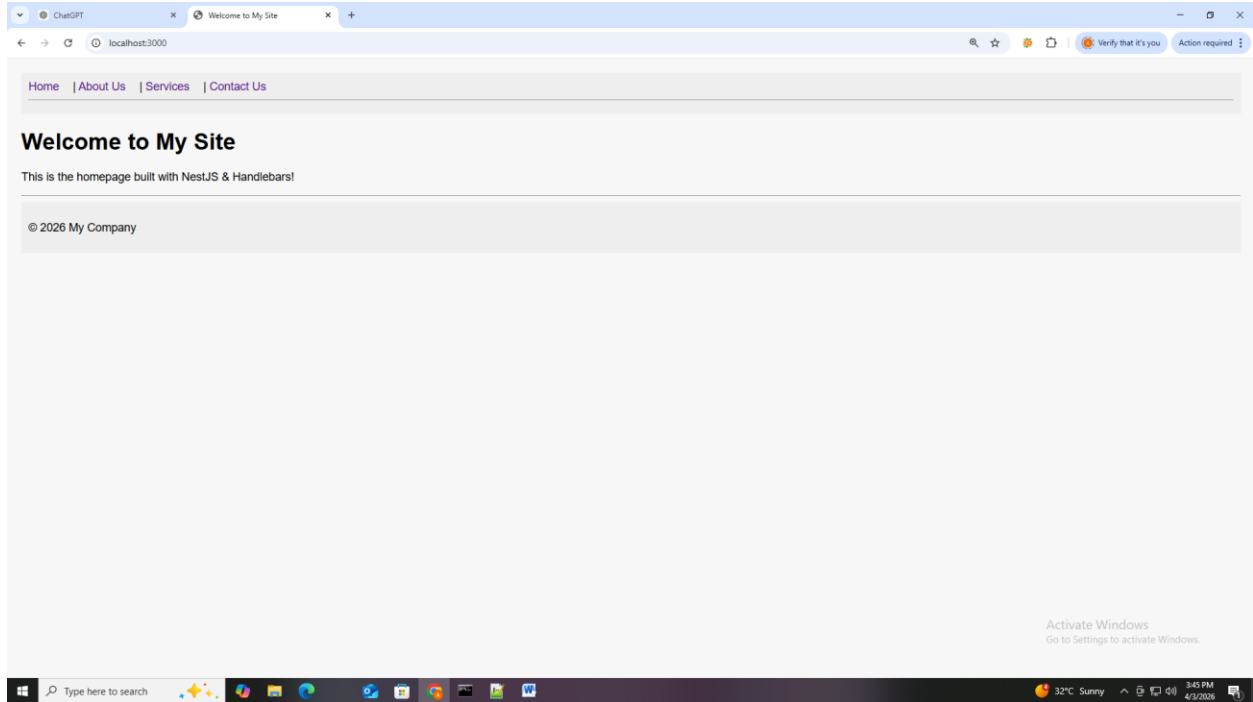
footer {
  text-align: center;
  margin-top: 20px;
  font-size: 0.9em;
  color: #666;
}
```

---

## Step 9 — Run the app

```
npm run start:dev
```

- Open <http://localhost:3000>
- Navigate to `/about`, `/services`, `/contact`
- Header/footer and CSS should appear on every page automatically



**Now you have a fully working NestJS + Handlebars website with:**

- Layout (`main.hbs`)
- Partials (`header.hbs`, `footer.hbs`)
- Static assets (CSS)
- Home, About, Services, Contact pages with working contact POST