Angular TS – Modules Components and Templates

[**✅ Task 2: Angular Modules, Components, Templates** 1](#_Toc204787116)

[**1️⃣ 🎯 Purpose** 1](#_Toc204787117)

[**2️⃣ 📚 Theory** 1](#_Toc204787118)

[**3️⃣ 🧱 Prerequisites** 1](#_Toc204787119)

[**4️⃣ 👨‍💻 Code Example (Angular 18 Standalone)** 1](#_Toc204787120)

[**🔹 Step-by-Step Code** 2](#_Toc204787121)

[**5️⃣ 📷 Project Snapshot** 3](#_Toc204787122)

[**6️⃣ 📌 Summary** 3](#_Toc204787123)

**✅ Task 2: Angular Modules, Components, Templates**

**1️⃣ 🎯 Purpose**

To understand the **core building blocks** of Angular:  
✅ Components (UI units),  
✅ Templates (HTML + Binding),  
✅ and how **Standalone Components** remove the need for NgModules.

**2️⃣ 📚 Theory**

| **Concept** | **Description** |
| --- | --- |
| **Component** | A TypeScript class with metadata (selector, template, styles) that controls a view |
| **Template** | The HTML that defines the component's UI |
| **Standalone** | Since Angular 14+, you can skip NgModules using standalone: true in the component decorator |
| **Bootstrap** | In Angular 18, components are bootstrapped via app.config.ts, not AppModule |

**3️⃣ 🧱 Prerequisites**

Ensure you’ve already done Task 1 and your app has:

* main.ts
* app.config.ts
* Angular version ≥ 17+

**4️⃣ 👨‍💻 Code Example (Angular 18 Standalone)**

**✅ Project Structure:**

src/

├── app/

│ ├── app.config.ts ✅ Angular 18 Routing + Bootstrap

│ ├── app.component.ts ✅ Root Component (Standalone)

│ ├── components/

│ │ └── student/

│ │ ├── student.component.ts ✅ Feature Component (Standalone)

│ │ ├── student.component.html

│ │ └── student.component.css

├── main.ts

**🔹 Step-by-Step Code**

**📁 app/app.component.ts ✅ Root Standalone Component**

import { Component } from '@angular/core';

@Component({

selector: 'app-root',

standalone: true,

template: `

<h1>Welcome to Angular 18</h1>

<app-student></app-student> <!-- using student component -->

`,

imports: [],

})

export class AppComponent {}

**📁 components/student/student.component.ts ✅ Standalone Component**

import { Component } from '@angular/core';

@Component({

selector: 'app-student',

standalone: true,

templateUrl: './student.component.html',

styleUrls: ['./student.component.css']

})

export class StudentComponent {

studentName: string = 'Kiran Kumar';

}

**📁 components/student/student.component.html**

<p>Hello {{ studentName }}! This is the Student Component.</p>

**📁 components/student/student.component.css**

p {

font-weight: bold;

color: darkgreen;

}

**📁 app/app.config.ts ✅ Angular 18 Bootstrap Config**

import { ApplicationConfig } from '@angular/core';

import { provideRouter } from '@angular/router';

import { bootstrapApplication } from '@angular/platform-browser';

import { AppComponent } from './app.component';

import { StudentComponent } from './components/student/student.component';

export const appConfig: ApplicationConfig = {

providers: [

provideRouter([]) // no routing yet

]

};

bootstrapApplication(AppComponent, {

providers: appConfig.providers,

}).catch(err => console.error(err));

**📁 main.ts**

import './app/app.config';

**5️⃣ 📷 Project Snapshot**

src/

├── app/

│ ├── app.component.ts

│ ├── app.config.ts

│ └── components/

│ └── student/

│ ├── student.component.ts

│ ├── student.component.html

│ └── student.component.css

├── main.ts

**6️⃣ 📌 Summary**

| **✅ Topic** | **💡 Covered** |
| --- | --- |
| Components | ✔️ |
| Templates | ✔️ |
| Standalone Setup | ✔️ |
| Angular 18 Structure | ✔️ |

This prepares your app to scale using only **standalone components**, no NgModules needs.