

Lab Title:

 **Personal Profile Page – Practicing Angular Components (No Data Binding)**

Objective:

By the end of this lab, you will be able to:

- Create multiple Angular components using the CLI.
 - Organize an application into small, reusable UI components.
 - Render components inside other components without passing data.
-

Scenario:

You are building a **personal profile page** for a portfolio site.
The page will be divided into multiple sections:

1. **ProfileHeaderComponent** – Displays profile photo and name.
2. **ProfileAboutComponent** – Displays a short “About Me” section.
3. **ProfileSkillsComponent** – Displays a list of skills.
4. **ProfileFooterComponent** – Displays contact information.

Each component will have **static HTML** in its template (no variables, no binding).

Steps to Implement:

1. Create Angular App

```
ng new personal-profile
cd personal-profile
ng serve
```

2. Generate Components

```
ng generate component profile-header
ng generate component profile-about
ng generate component profile-skills
ng generate component profile-footer
```

3. Add Static HTML to Each Component

profile-header.component.html

```
<div style="text-align:center;">
  
  <h1>John Doe</h1>
  <h3>Software Engineer</h3>
</div>
```

profile-about.component.html

```
<section>
  <h2>About Me</h2>
  <p>
    I am a passionate software developer with 5 years of experience
    in building modern web applications. I love working with Angular, .NET
    Core, and cloud technologies.
  </p>
</section>
```

profile-skills.component.html

```
<section>
  <h2>Skills</h2>
  <ul>
    <li>Angular</li>
    <li>TypeScript</li>
    <li>.NET Core</li>
    <li>SQL</li>
    <li>Azure</li>
  </ul>
</section>
```

profile-footer.component.html

```
<footer style="text-align:center; margin-top:20px;">
  <p>Contact: johndoe@example.com</p>
  <p>LinkedIn: linkedin.com/in/johndoe</p>
</footer>
```

4. Add Components to the App Component Template

app.component.html

```
<app-profile-header></app-profile-header>
<hr>
<app-profile-about></app-profile-about>
<hr>
<app-profile-skills></app-profile-skills>
<hr>
<app-profile-footer></app-profile-footer>
```

5. Add Styling (Optional)

styles.css

```
body {  
  font-family: Arial, sans-serif;  
  margin: 20px;  
}  
h1, h2, h3 {  
  color: #333;  
}  
ul {  
  list-style-type: square;  
}
```

Expected Output

When you run `ng serve`, the browser should display:

- **Header** with photo, name, and job title.
 - **About Me** section with a short paragraph.
 - **Skills** list.
 - **Footer** with contact info.
-

Extra Practice Ideas

- Add a **ProfileProjectsComponent** with 3 static project descriptions.
- Change the layout using Bootstrap grid system.
- Replace the `<hr>` with better spacing and styling.