Angular

- Standalone Component -

Angular standalone components are a way to build components without needing an NgModule, making applications simpler and more modular.

What is a Standalone Component?

- Self-contained unit: A standalone component can operate independently without being declared inside an NgModule.
- Introduced in Angular v14: This feature streamlines development by reducing boilerplate code.
- Marked with standalone: true: When defining a component, directive, or pipe, you set this flag to make it standalone.
- Direct imports: Instead of relying on NgModules, standalone components import other components, directives, and pipes directly.
- Incremental adoption: Existing projects can gradually migrate to standalone style without breaking changes.

How to Create One

- import { Component } from '@angular/core';
- import { CommonModule } from '@angular/common';
-
- @Component({
-   selector: 'app-hello',
-   standalone: true,
-   imports: [CommonModule],
-   template: `<h1>Hello Standalone!</h1>`
- })
- export class HelloComponent {}

Clarification

- Standalone flag: standalone: true makes it independent.
- Imports array: You directly import CommonModule or other standalone components.
- No NgModule needed: You can bootstrap this component directly in main.ts.

Benefits

- Simplified structure: Reduces reliance on NgModules.

- Flexibility: Components can be reused more easily.

- Cleaner codebase: Less boilerplate, more focus on functionality.

- Gradual migration: Existing apps can adopt standalone components step by step.

- Lazy Loading possible.

Example

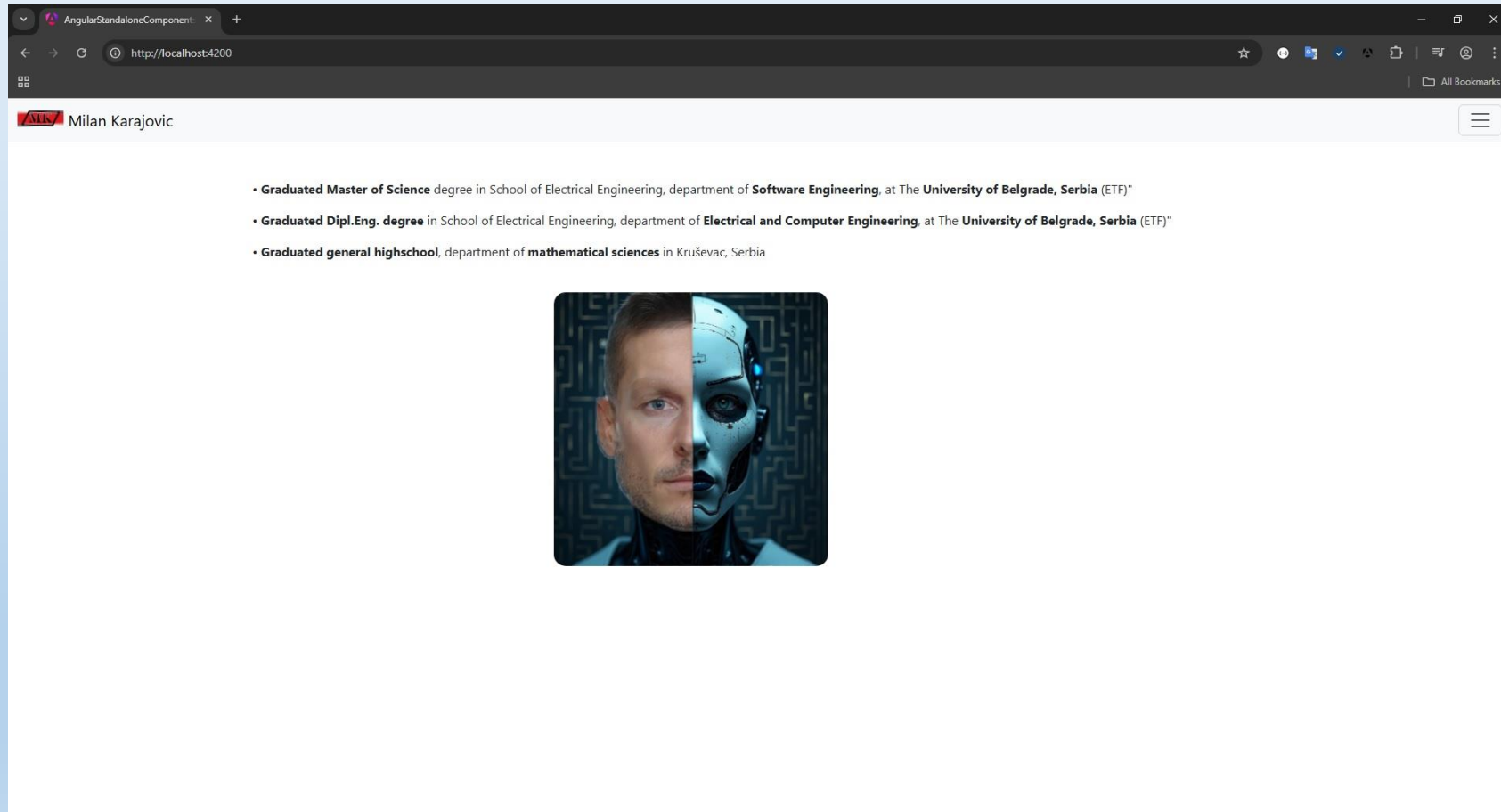- Example: Portfolio - Example Angular Standalone components.
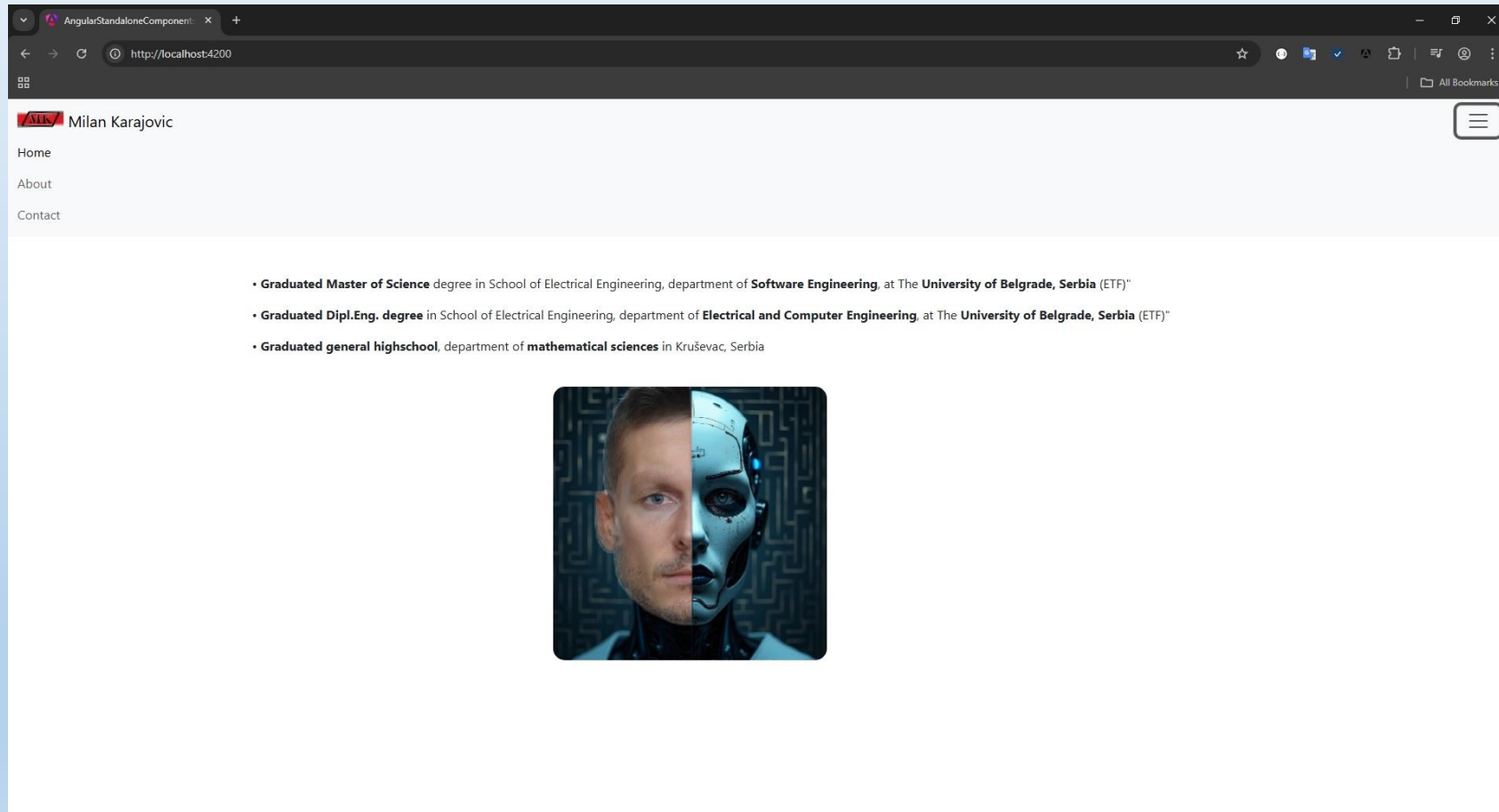
- GitHub with Source Code:

https://github.com/Milan-Karajovic/Angular-Standalone-Components

- LinkedIn:

https://www.linkedin.com/in/milan-karajovic-java-angular-security-cloud-architect/
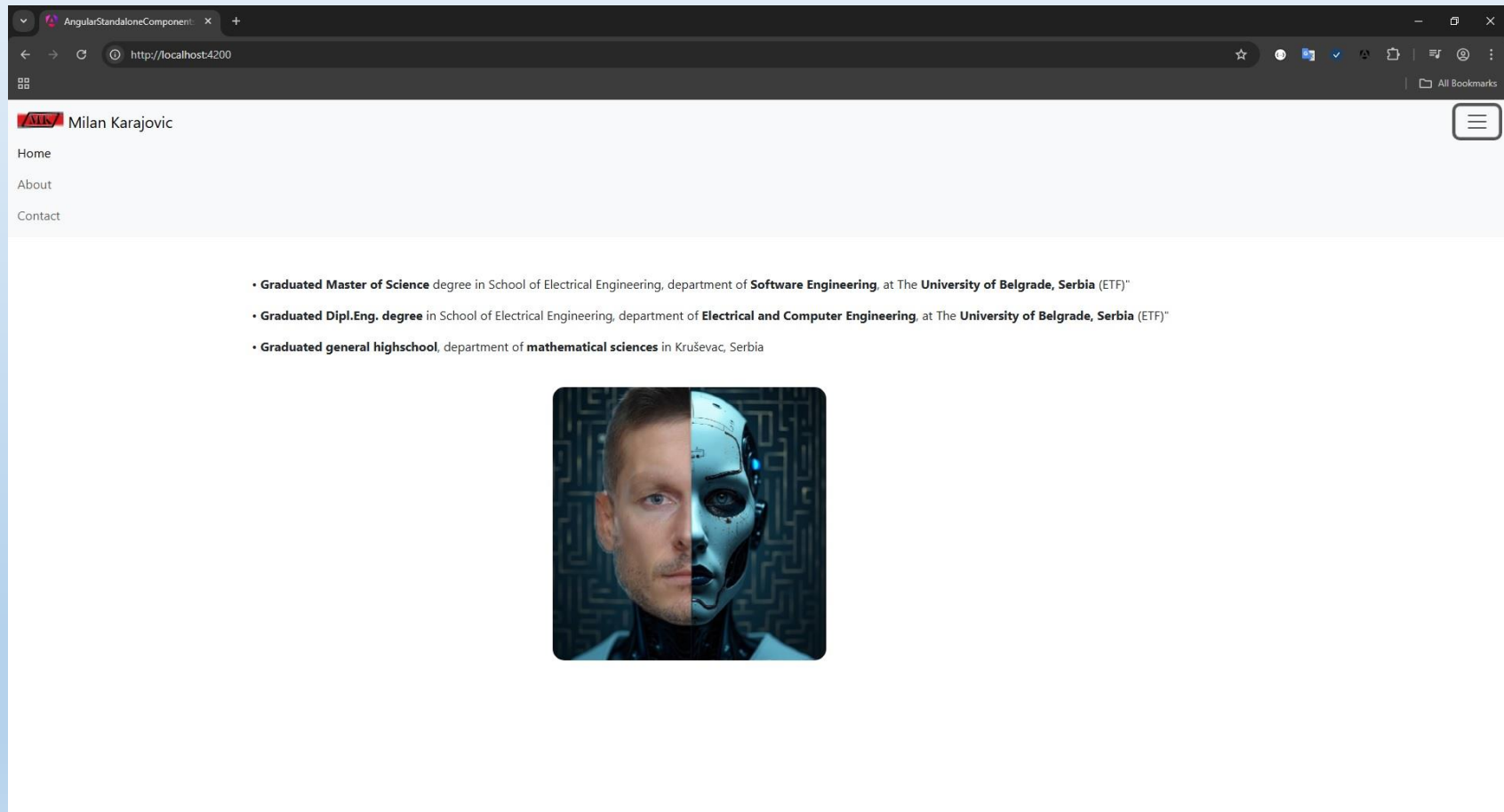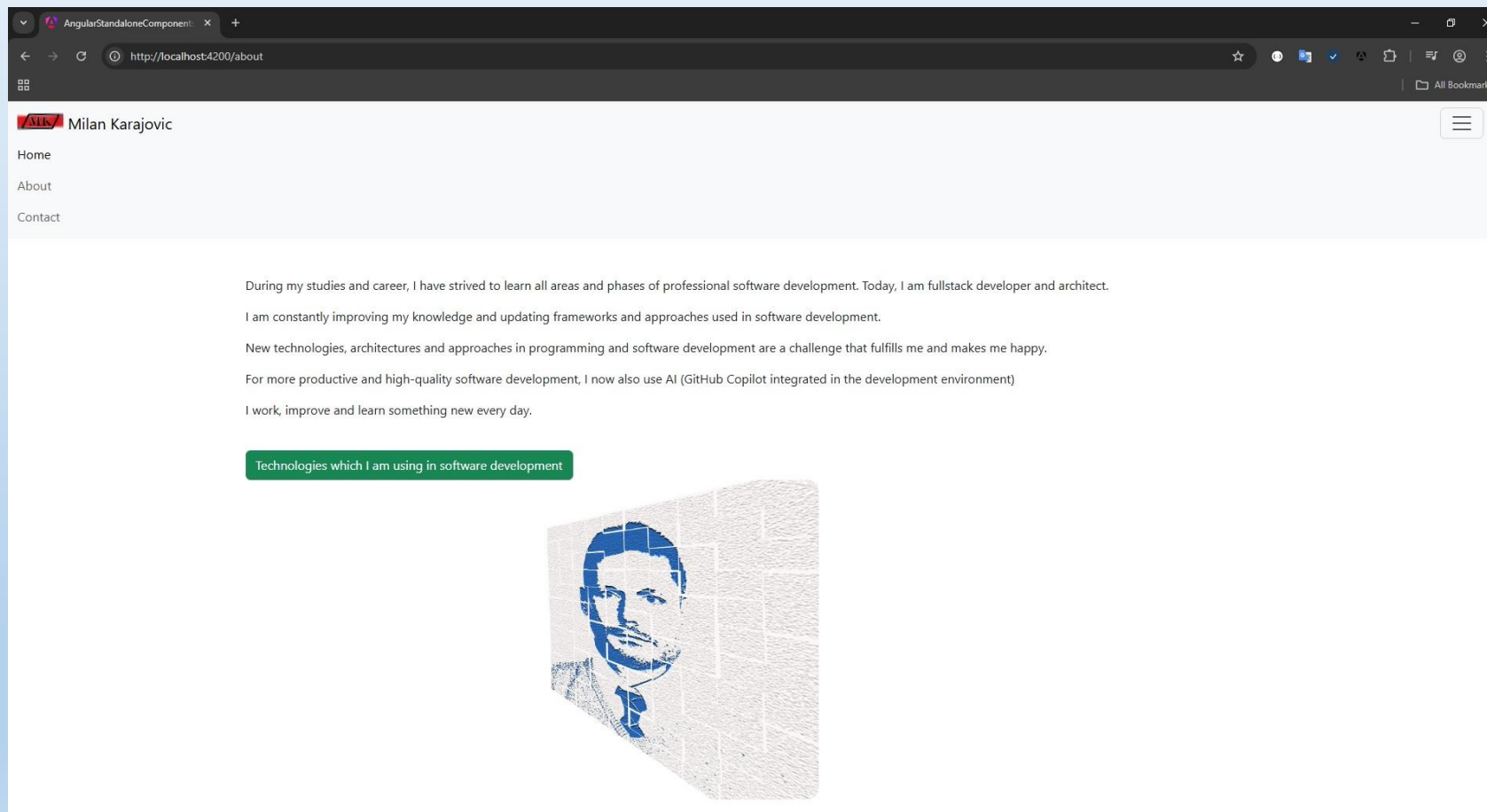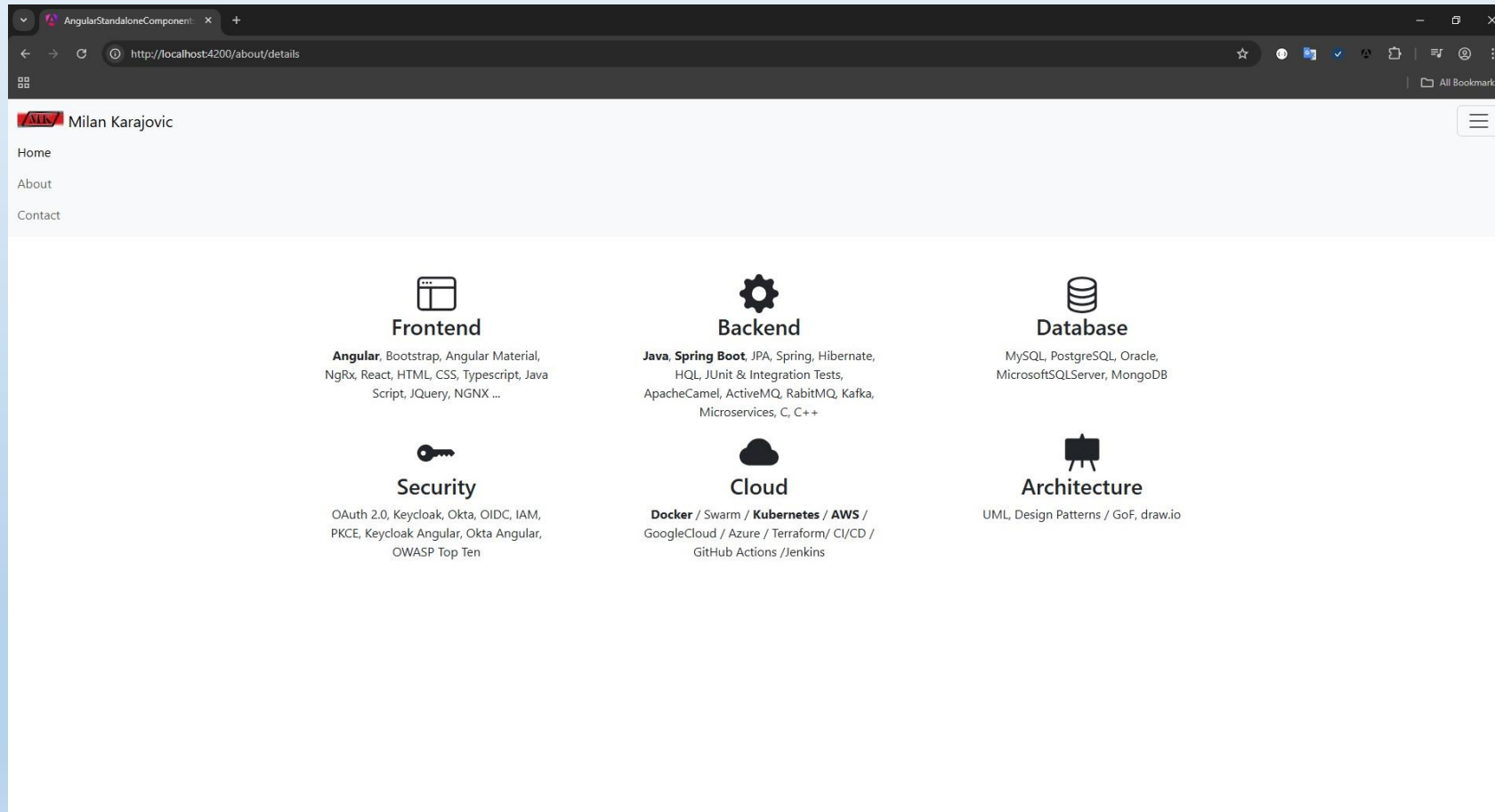
Home

Navigation

About
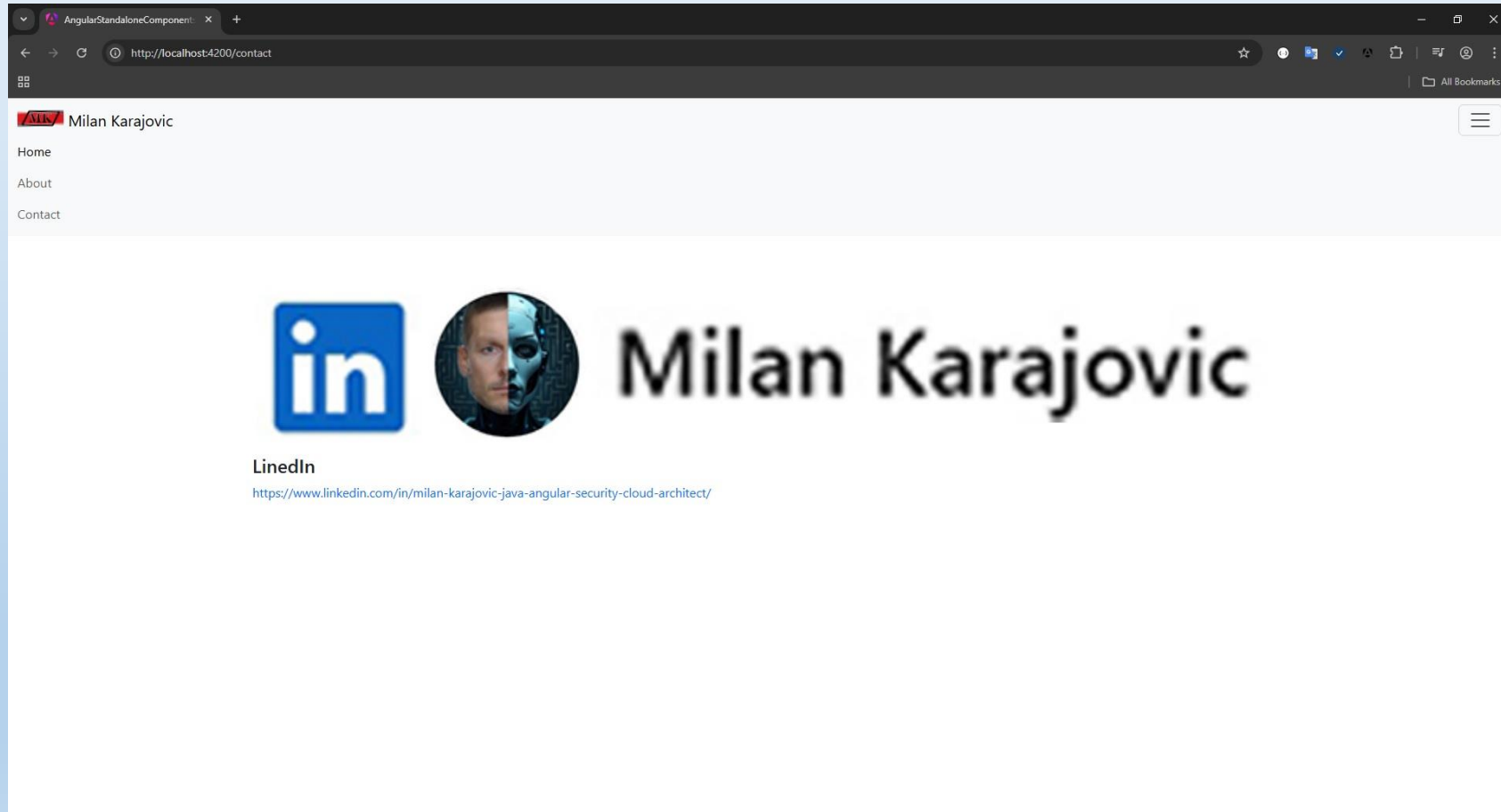
About

About -> Details

Contact

Thanks

- See you again!