Angular has evolved since it’s inception in 2010. It has changed substantially in the last couple years with version 7 just release in October 2018. Ideally used to create Single Page Applications (SPAs). Angular uses TypeScript—object oriented JavaScript.

**Technology Stack:**

Server:

* NodeJS v10
* ASP.NET Core 2.2 Web API
* Entity Framework Core 2.2
* SQL Server

Client:

* Angular 7 (or 8)
* Angular CLI (Command Line Interface)

Visual Studio Code

Links:

<https://angular.io/>

<https://nodejs.org/en/>

<https://code.visualstudio.com/>

**Environment Setup:**

Install NodeJS if not installed already. Run **node –v** or **node –version** from command line to check version. You require minimum version 8 or above (preferably v10)

Install Angular CLI from command line: **npm install –g @angular/cli**

Use a projects folder and make a new directory for the new application (e.g. SpaDemo). Navigate to this new folder from command line and create a new Angular application: **ng new SpaDemo**

You are asked two questions: Do you want to add routing? Answer yes. What stylesheet format would you like to use? Select SCSS. App creation takes a few minutes.

Navigate into the new application folder: **cd SpaDemo**

Launch Visual Studio Code from the command line: **code .** (that is type code, a space, and a period)

Start a development server using Angular CLI: **ng serve –o**  (-o opens a browser at localhost:4200)

Tutorial Link: <https://coursetro.com/posts/code/171/Angular-7-Tutorial---Learn-Angular-7-by-Example>

**Components – Angular Building Blocks:**

Components are made up of three main elements:

* The HTML template
* The main logic file (TypeScript)
* The style sheet

When a new project is created using Angular CLI, a single component is created called app and the files are called:

* app.component.html
* app.component.scss
* app.component.ts

The app.component.ts file looks like this:

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

@Component({

selector: 'app-root',

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

styleUrls: ['./app.component.scss']

})

export class AppComponent {

title = 'SpaDemo';

}

Component is imported from @angular/core

The @Component decorator provides configuration options

The logic is contained in a class called AppComponent which contains a single property called title

Add new components using this in the SpaDemo folder: ng generate component <component name>

We want to add the following components:

* Home
* About
* Contacts
* Instructors

**Nav Css Code:**

header {

background: rgb(16, 65, 110);

.logo {

font-weight: bold;

}

nav {

justify-self: right;

ul {

list-style-type: none;

margin: 0; padding: 0;

li {

float: left;

a {

padding: 1.5em;

text-transform: uppercase;

font-size: .8em;

&:hover {

background: #8E2BFF;

}

}

}

}

}

}

**Home Css Code:**

ul {

list-style-type: none;

margin: 0;padding: 0;

li {

background: rgb(238, 238, 238);

padding: 2em;

border-radius: 4px;

margin-bottom: 7px;

display: grid;

grid-template-columns: 60px auto;

p {

font-weight: bold;

margin-left: 20px;

}

img {

border-radius: 50%;

width: 100%;

}

}

}

**Contacts Css Code:**

label {

display: block;

input, textarea {

display: block;

width: 50%;

margin-bottom: 20px;

padding: 1em;

}

.error {

margin-top: -20px;

background: yellow;

padding: .5em;

display: inline-block;

font-size: .9em;

margin-bottom: 20px;

}

}

.cta {

background: #7700FF;

border: none;

color: white;

text-transform: uppercase;

border-radius: 4px;

padding: 1em;

cursor: pointer;

font-family: 'Montserrat';

}

.results {

margin-top: 50px;

strong {

display: block;

}

span {

margin-bottom: 20px;

display: block;

}

}