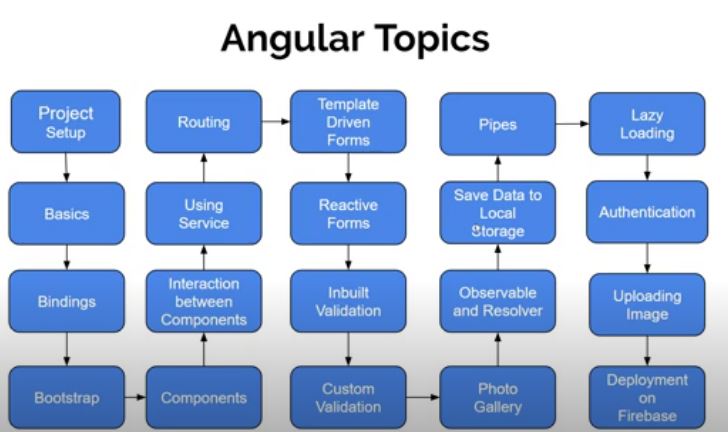
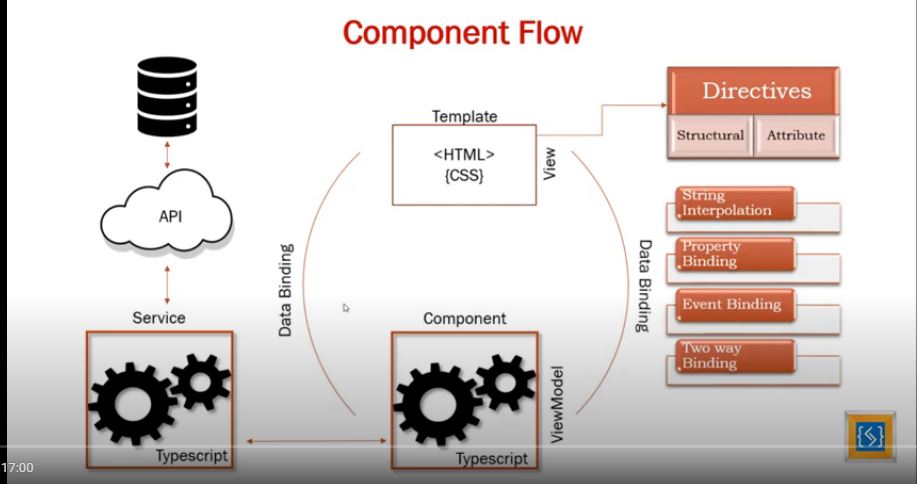
**Real Estate Agency using Angular 11 and Asp.Net core web api and boostrap 4**

These are 2 of the hottest frameworks right now for the ‘back-end’ (Microsoft’s ASP.NET Core) and the ‘front-end’ (Google’s Angular) and are well worth spending the time to learn. This course starts from scratch, you neither need to know Angular 1 nor Angular 2 We will start from nothing and incrementally build this property dealing application front-end using Angular 10. And then we will connect our front-end with the Web-API until we have a fully functional Web Application that we will publish to Firebase and then on IIS. Here are some of the things you will learn about in this course: Creating the Angular App and ASP.NET Core WebAPI using the DotNet CLI and the Angular CLI Adding a Client side login and register function to our Angular application Adding routing to the Angular application and securing routes. Using Automapper in ASP.NET Core Building a great looking UI using Bootstrap Adding Photo Upload functionality as well as a cool looking gallery in Angular Angular Template forms and Reactive forms and validation Paging, Sorting and Filtering Publishing the application to Firebase, IIS and then on Azure.

****

First we install the vs code extension write angular essential (john papa) and install it .



**Components:**

First we create components of our project we make parent and child component relations

Firstly we make a new folder in app name property then we add a component property-card component (child) and then we add a component property-list component(parent) also we create a component navbar they are not include in property folder remember.

**Property-card Component:(Child)**

In property- card component we show the list of houses in column wise where the card show the detail of house name ,number and contact so we created input decorator to tell the parent component about property of house so they use of input using databinding and loop and array to show multiple houses.

**Property-card.component.ts**

import { Component, Input, OnInit } from '@angular/core';

@Component({

  selector: 'app-property-card',

  templateUrl: './property-card.component.html',

  styleUrls: ['./property-card.component.css']

})

export class PropertyCardComponent implements OnInit {

  @Input() propertynames:any

  constructor() { }

  ngOnInit(): void {

  }

}

**Property-card.component.html**

  <div class="card">

    <div class="card-img-wrapper">

<img class="card-img-top" src="assets/images/house.jpg">

<ul class="list-inline text-center member-icons animate">

  <li class="list-inline-item">

    <button class="btn btn-primary">

      <i class="far fa-edit"></i> <!--font awesome icon used-->

    </button>

  </li>

    <li class="list-inline-item">

<button class="btn btn-primary">

  <i class="far fa-address-card"></i><!--font awesome icon used-->

    </button>

  </li>

</ul>

</div>

<div class="card-body p-2">

<h3>{{propertynames.Name}}</h3>

<h3 >{{propertynames.Type}}</h3>

<h3>{{propertynames.Price}}</h3>

</div>

</div>

**Property-card.component.css**

.h{

  color:green;

}

.propImg{

  height: 250px;

}

.card:hover img{

  transform: scale(1.2,1.2);

  transition-duration: 500ms;

  transition-timing-function: ease-out;

}

.card img{

  transform: scale(1,1);

  transition-duration: 500ms;

  transition-timing-function: ease-out;

}

.card-img-wrapper{

  overflow: hidden;

  position: relative;

}

 .member-icons{

  position: absolute;

  bottom: -30%;

  right: 0;

  left: 0;

  margin-right: auto;

  margin-left: auto;

}

.card-img-wrapper:hover .member-icons{

  bottom: 0;

}

.animate{

  transition: all 0.3s ease-in-out;

}

**Property-list.component.ts(Parent)**

Where we fetch the data of our child component using @input decorator first we make an array of object wo make 6 houses data then in html file we using structural directives ngFor to loop through the array data and using databinding to communicate our child component.

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

@Component({

  selector: 'app-property-list',

  templateUrl: './property-list.component.html',

  styleUrls: ['./property-list.component.css']

})

export class PropertyListComponent implements OnInit {

  Properties: Array<any> = [

    {

    "Id":1,

    "Name": "Zaryab House",

    "Type":"House",

    "Price": 1200000

  },

  {

    "Id":2,

    "Name": "Shahroz House",

    "Type":"Plot",

    "Price": 2000000

  },

  {

    "Id":3,

    "Name": "Ahsan House",

    "Type":"Flat",

    "Price": 1300000

  },

  {

    "Id":4,

    "Name": "Minaj House",

    "Type":"bugalow",

    "Price": 2400000

  },

  {

    "Id":5,

    "Name": "Minaj House",

    "Type":"bugalow",

    "Price": 2400000

  },

  {

    "Id":6,

    "Name": "Minaj House",

    "Type":"bugalow",

    "Price": 2400000

  },

]

  constructor() { }

  ngOnInit(): void {

  }

}

**Property-list.component.html**

<div class="row">

  <div \*ngFor='let prop of Properties ' class="shadow-lg col-sm-4 p-3">

<app-property-card  [propertynames]= "prop" ></app-property-card>

</div>

</div>

**App.component.html**

<app-nav-bar></app-nav-bar>

<div class="container" style="margin-top: 70px;">

<app-property-list></app-property-list>

</div>

**Navbar.component.html**

<nav class="navbar navbar-expand-lg navbar-dark bg-dark fixed-top">

  <a class="navbar-brand" href="#">Real Estate</a>

  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

    <span class="navbar-toggler-icon"></span>

  </button>

  <div class="collapse navbar-collapse" id="navbarSupportedContent">

    <ul class="navbar-nav mr-auto">

      <li class="nav-item active">

        <a class="nav-link" href="#">Home <span class="sr-only">(current)</span></a>

      </li>

      <li class="nav-item">

        <a class="nav-link" href="#">Link</a>

      </li>

      <li class="nav-item dropdown">

        <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">

          Dropdown

        </a>

        <div class="dropdown-menu" aria-labelledby="navbarDropdown">

          <a class="dropdown-item" href="#">Action</a>

          <a class="dropdown-item" href="#">Another action</a>

          <div class="dropdown-divider"></div>

          <a class="dropdown-item" href="#">Something else here</a>

        </div>

      </li>

      <li class="nav-item">

        <a class="nav-link disabled" href="#" tabindex="-1" aria-disabled="true">Disabled</a>

      </li>

    </ul>

    <form class="form-inline my-2 my-lg-0">

      <input class="form-control mr-sm-2" type="search" placeholder="Search" aria-label="Search">

      <button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search</button>

    </form>

  </div>

</nav>