Mobile Application Development

LECTURE 2

Introduction to Angular 

- Angular is a full featured JavaScript framework created and maintained by Google and is used for building front-end applications.

- Angular is a Javascript framework to build more interactive web apps. It is designed for both Web, Desktop and Mobile platforms. While, we create apps using HTML, CSS and Javascript, Angular requires us to know **Typescript** (a typed superset of Javascript that scales), kind of stricter version of Javascript provided with OOPS features.

Introduction to Angular 

- Angular is very popular in large enterprise.
- AngularJS **!==** Angular 2/4/5/6/7/8
- AngularJS was released in 2010. Its architecture is very different from Angular 2 onwards.
- Angular 2/4/5/6/7/8+ are not much different. They mostly have under the hood changes to make angular faster and smaller in size.

Frontend and Backend options

Frontend Frameworks / Libraries

- jQuery, AngularJS, **Angular**, React.js, Vuejs, Aurelia, Ember, Knockout, Svelte etc

Backend Options / Frameworks

- Nodejs, C#, Python, Java, Go, Rust, Elixir, PHP etc

- **Nodejs Frameworks** (Express.js, Nest.js, Loopback.js, Koa, Sails, Socket.IO, Hapi.js etc)

Why Angular 

- Angular is an all-in-one solution (Routing/HTTP/Forms/Services/RxJS). Angular really provides so much out of the box (Directives, Pipes etc)

  React.js is just a library. In order to to build complete front-end applications, you need to use *react-router* (router), *formik* (forms), *redux/mobx* (state management).

  Vue.js would also require *vue-router* and *vuex*.

- Provides modular concept which helps for code reuse and easier scaling of architecture.

Why Angular 

- Easy to build enterprise level SPA applications (Single Page Applications). A **single-page** application (SPA) is a web application or web site that interacts with the user by dynamically rewriting the current **page** rather than loading entire new pages from a server.
- Angular Reactive Forms are a breeze to work with.
- Follows an MVC architecture. You HTML is (view), and your components/directives are controllers.

“Care about what other people think and you will always be their prisoner.” ~ Lao Tzu

Introduction to **TypeScript**

- **TypeScript** is an open-sourced programming language developed and maintained by Microsoft.
- **Typescript** is a super-set of JavaScript to compiles to plain JavaScript. Superset (anything u can do in JS, u can also do in typescript) and adds some extra features.
- TypeScript is modern JavaScript (ES6,ES7+) + types. It's about catching bugs early and making you a more efficient developer.

Introduction to TypeScript

- JavaScript is dynamically typed / untyped. This means JavaScript does not know what type a variable is until it is actually instantiated at run-time. This means we donot have to assign types to variables like in Java.
- Designed for large codebases.
- Angular uses it by default but now TypeScript is being adopted by many projects including Node.js frameworks such as (Nest.js) and even React.js.

Introduction to TypeScript

- TypeScript syntax is closer to Java/C#.
- Static type checking is completely optional.
- TypeScript helps us find bugs at compile time.
- TypeScript provides OOP in JavaScript. Class based objects in JS.
- It has many features such as Enums, Interfaces, Generics, Tuple.

Explanation of Typed in **TypeScript**

Assigning a string to a variable but TS compiler is telling us it cant assign it.

```
marks: number;

changeTitle() {
  this.marks = 'a';
}
```

```
expo
  ma
    (property) AppComponent.marks: number
    Type '"a"' is not assignable to type 'number'. ts(2322)
  ch  Peek Problem   No quick fixes available
    this.marks = 'a';
  }

}
```

Lets start by setting up an Angular project

Step 1. Make sure you have angular CLI installed. (also Node.js)

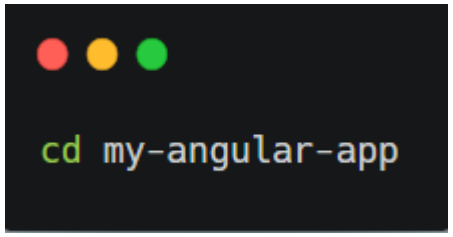
```
npm install -g @angular/cli
```

Step 2. Install an angular app using Angular CLI

```
ng new my-angular-app
```

Lets start by setting up an Angular project

Step 3. Once Installed, lets head over the directory



Step 4. Last but not the least