Angular 15

- Day - 1
    - Building a walking skeleton Part Two - Angular - **1hr 15min**
        - Creating the Angular application
        - Running the angular project and reviewing the bootstrap of the app
        - Adding VS Code extensions to work with Angular.
        - Making HTTP requests in Angular
        - Adding CORS support in the API
        - Displaying the fetched users in the browser
        - Adding bootstrap and font-awesome - Optional
        - Adding HTTPS to Angular using mkcert
        - Saving into source control.
    - Client login and register - **2hr**
        - Creating a nav bar
        - Introduction to Angular template forms
        - Introduction to Angular services
        - Injecting services into components
        - Using conditionals to show and remove content.
        - Using the angular bootstrap components - dropdown
        - Introduction to observables
        - Persisting the login
        - Adding a home page
        - Adding a register form
        - Parent to child communication
        - Child to parent communication
        - Hooking up the register method to the service
    - Using Pipes to Transform Output – **45 minutes.**
        - Introduction & Why Pipes are Useful.
        - Using Pipes
        - Parametrizing Pipes
        - Where to learn more about Pipes
        - Chaining Multiple Pipes
        - Creating a Custom Pipe
        - Parametrizing a Custom Pipe
        - Example: Creating a Filter Pipe
        - Pure and Impure Pipes (or: How to "fix" the Filter Pipe)
        - Understanding the "async" Pipe

- Day - 2
  - Routing in Angular – **1hr**
    - Creating some more components
    - Adding the nav links
    - Routing in code
    - Adding a toast service for notifications
    - Adding an Angular route guard
    - Adding a dummy route
    - Adding a new theme
    - Tidying up the app module by using a shared module
  - Authentication & Route Protection - **2hr**
    - Authentication & Route Protection
    - How Authentication Works
    - Adding the Auth Page
    - Switching Between Auth Modes
    - Handling Form Input
    - Preparing the Backend
    - Make sure you got Recipes in your backend!
    - Preparing the Signup Request
    - Sending the Signup Request
    - Adding a Loading Spinner & Error Handling Logic
    - Improving Error Handling
    - Sending Login Requests
    - Login Error Handling
    - Creating & Storing the User Data
    - Reflecting the Auth State in the UI
    - Adding the Token to Outgoing Requests
    - Attaching the Token with an Interceptor
    - Adding Logout
    - Adding Auto-Login
    - Adding Auto-Logout
    - Adding an Auth Guard

  - Dynamic Components – **1hr**
    - Adding an Alert Modal Component
    - Understanding the Different Approaches
    - Using ngIf
    - Preparing Programmatic Creation

        - Creating a Component Programmatically
        - About entryComponents
        - Understanding entryComponents
        - Data Binding & Event Binding

- Day - 3
    - Building the user Interface – **1hr**
        - Using TypeScript
        - Creating the member interface
        - Adding a member service
        - Retrieving the list of members
        - Creating member cards
        - Adding animated buttons
        - Using an interceptor to send the token.
        - Routing to the detailed page
        - Styling the member detailed page
        - Styling the member detailed page part two
        - Adding a photo gallery (Angular 16+)
    - Updating resources – **1hr**
        - Creating a member edit component.
        - Creating the edit template form
        - Adding the update functionality
        - Adding a Can Deactivate route guard.
        - Persisting the changes in the API
        - Updating the user in the client app
        - Adding loading indicators
        - Using the service to store state
    - Adding photo upload functionality – **1hr**
        - Configuring cloud in the API
        - Adding a photo service
        - Testing the photo upload
        - Using the Created at Route method
        - Adding a photo editor component
        - Adding a photo uploader
        - Adding the main photo image to the nav bar
        - Setting the main photo in the client
        - Deleting photos

- Day - 4
  - Reactive forms – **1hr 30 minutes**
    - Reactive forms introduction
    - Client side validation
    - Adding custom validators
    - Validation feedback
    - Creating a reusable text input
    - Using the form builder service
    - Expanding the register form
    - Adding a reusable date input
    - Updating the API register method
    - Client side registration
  - Paging Sorting and Filtering - **2hr 50 minutes**
    - Adding a paged list class
    - Adding helper classes for pagination
    - Using the pagination classes
    - Setting up client pagination
    - Using the angular bootstrap pagination component
    - Adding additional filters
    - Cleaning up the member service
    - Adding filter buttons to the client
    - Adding sorting functionality
    - Adding an action filter
    - Making the Last Active action filter more optimal
    - Sorting on the client
    - Formatting dates
    - Restoring the caching for members
    - Restoring caching for member detailed
    - Remembering the filters for a user in the service
- Day -5
  - Adding the Like user feature – **30 minutes**
    - Setting up the likes functions
    - Adding the likes component
    - Adding pagination for the likes
    - Paginating the likes on the client
  - Deployment - **15 minutes**
    - Deployment Preparation & Steps
    - Using Environment Variables

    - Deploying Angular Applications
    - Deployment Example: Firebase Hosting
    - Server Routing vs Browser Routing
  - Using NgRx for State Management – **4hr (Optional)**
    - What Is NgRx?
    - Understanding NgRx & Its Building Blocks
    - Project Setup & Installing NgRx
    - Adding a First Reducer & Store Setup
    - An Alternative Way Of Creating Reducers
    - Reading Data From The Store
    - Introducing Actions & State Changing Reducers
    - Dispatching Actions
    - Attaching Data To Actions
    - Handling Actions Without createReducer
    - An Alternative Way Of Defining Actions
    - Time To Practice: A Second Action
    - Exploring Selectors
    - Introducing Effects
    - Installing the Effects Package
    - Defining a First Effect
    - The Old @Effect Decorator & Registering Effects
    - Using Store Data In Effects
    - Adding a Second Effect
    - About The Remaining Section
    - Recipe Project Starting Code
    - Getting Started with Reducers
    - Adding Logic to the Reducer
    - Understanding & Adding Actions
    - Setting Up the NgRx Store
    - Selecting State
    - Dispatching Actions
    - Multiple Actions
    - Preparing Update & Delete Actions
    - Updating & Deleting Ingredients
    - Expanding the State
    - Managing More State via NgRx
    - Removing Redundant Component State Management
    - One Root State

        - Setting Up Auth Reducer & Actions
        - Dispatching Auth Actions
        - Auth Finished
        - And Important Note on Actions
        - Exploring NgRx Effects
        - Defining the First Effect
        - Important: Avoid @Effect
        - Effects & Error Handling
        - Login via NgRx Effects
        - Managing UI State in NgRx
        - Finishing the Login Effect
        - Preparing Other Auth Actions
        - Adding Signup
        - Further Auth Effects
        - Adding Auto-Login with NgRx
        - Adding Auto-Logout
        - Finishing the Auth Effects
        - Using the Store Devtools
        - The Router Store
        - Getting Started with NgRx for Recipes
        - Fetching Recipe Detail Data
        - Fetching Recipes & Using the Resolver
        - Fixing the Auth Redirect
        - Update, Delete and Add Recipes
        - Storing Recipes via Effects
        - Alternative NgRx Syntax

- Day – 6
    - Basic Introduction to Unit Testing (Karma and Jasmine)
        - Unit Testing
        - Creating our first unit test
        - Running unit tests using Karma
        - Creating a service to test
        - Unit testing the service.
        - Adding a Component and some fitting Tests.
        - Creating a component to test.
        - Component fixture for unit testing.
        - Handling dependencies for component tests.
        - Testing Dependencies: Components and Services.

- Testing a component property that loads from ngOnInit.
- Testing a components UI functionality.
- Using "fakeAsync" and "tick"
- Isolated vs Non-Isolated Tests