Angular

Exercise 1

1. Create two new components (manually or with cli): WarningAlert and SuccessAlert
2. Output them beneath each other in AppComponent
3. Output a warning (This is a warning, you are in danger) or success (You are so successful) message in the components
4. Style the components

Exercise 2

1. Add a input field which updates a property (user\_name) via two-way binding
2. Output the property with string interpolation
3. Add a button which may be clicked only when the user\_name is not empty
4. Upon clicking the button reset the user\_name to empty string

Exercise 3

1. Add a button which says “Display details” and a paragraph with any content
2. Toggle the displaying of the paragraph using the button created in first step
3. Log all button clicks in an array and output the array below the paragraph
4. Starting at the 5th log item, give all future log items a blue background(use ngClass) and text color white(use ngStyle)

Exercise 4

1. Create 3 new components, GameControl, Odd and Even
2. GamControl component should have buttons to start and stop the game
3. When starting a game an event should be emitted each second(use setInterval())
4. Event should be listenable from outside of the component
5. When stopping the game no more events should be emitted(clearInterval(ref))
6. Odd component should be displayed for every odd number emitted and Even component for even number
7. Simply output ODD-NUMBER and EVEN-NUMBER texts respectively
8. Style elements with different colors and background

Exercise 5

1. Create a list of Active and Inactive Users separated by a line
2. The list should have the username and the status( Active or Inactive)
3. The status should be a link
4. On clicking the links the status should be toggled (Active to Inactive and vice versa) and by doing so the user should be placed in the respective list
5. Use services for toggling the Data

Exercise 6

1. Create 3 components Destination List, Detailed View and Not Found
2. Destination List Component screen should show list of destinations to travel. When the URL matches destinations, Destination List Component is to be rendered. It should display a list of cities
3. Detailed View screen should show a detailed description of the selected city. When the URL matches destinations/:city, Detailed View screen should be rendered
4. Add a textarea in the Detailed View screen below the city description which should be editable/non editable depending upon query parameter passed from Destination List Component
5. Add a button in Detailed View screen to navigate back to Destination List Component
6. When the paths don’t match the above mentioned path, render the Not Found Component

Exercise 7

1. Create a form with following controls and validators
2. Text Field – Project Name(should not be empty)
3. Email Field – Main(should not be empty and a valid email)
4. Project Status dropdown with three values ‘Stable’, ‘Critical’ and ‘Finished’
5. Button – Submit(enabled only when the validations are met)
6. Add own validator which doesn’t allow ‘Test’ as a project name
7. Upon submitting the form, print values

Exercise 8

1. Create a pipe which takes telephone number as a string and displays in format (123) 456-7891

Exercise 9

Use Firebase for the following

1. Add user login and logout
2. Retrieve user profile information
3. Protect application routes
4. Call an API with protected endpoints
5. Add an interceptor which adds a common Header to all http requests

Exercise 10

For dynamic component, don’t use ngIf, use componentFactoryResolver

1. Create an alert modal as a dynamic component
2. Use this component to display error and warning message in application