Angular 6.x For JEE

Lab Book

Document Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Revision No. | Author | **Summary of Changes** |
| March 2018 | 1 | Rahul Vikash | Created new lab book as per course contents |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

*Getting Started..……..…………………………………………………………………………… 4*

[Overview 4](#_Toc452123394)

[Setup Checklist for AngularJS 4](#_Toc452123395)

[Instructions 4](#_Toc452123396)

[Learning More 4](#_Toc452123397)

[*Lab 1. TypeScript Angular Component & Data Binding 5*](#_Toc452123398)

[*Lab 2. Angular Directives & Data Binding 7*](#_Toc452123400)

[*Lab 3. Angular with Form 7*](#_Toc452123400)

[*Lab 4. Angular Services & Pipes 7*](#_Toc452123400)

[*Lab 5: Angular CRUD with Java middle ware application (Spring REST) with Router 11*](#_Toc452123410)

Lab6: Using RxJS operators

*Appendices………………………………………………………………………………………… 11*

## Appendix A: Table of Figures ………………………………………………………… 11

Getting Started

## Overview

This lab book is a guided tour for learning Angular 2.0 For JEE version and above. It comprises ‘To Do’ assignments. Follow the steps provided to work out the ‘To Do’ assignments given.

## Setup Checklist for Angular 2.0

Here’s what is expected on your machine for the lab in order to work.

Minimum System Requirements

* Intel Pentium 90 or higher (P166 recommended)
* Microsoft Windows XP, Windows 7 or Windows 8
* Memory: 2GB of RAM (1GB or more recommended)
* Google Chrome 36.0 or Mozilla Firefox 31.0 or Internet Explorer 10 or above

Please ensure that the following is done:

* A text editor like Notepad or Notepad++ or Eclipse Luna is installed.
* Node.js install with node module

## Instructions

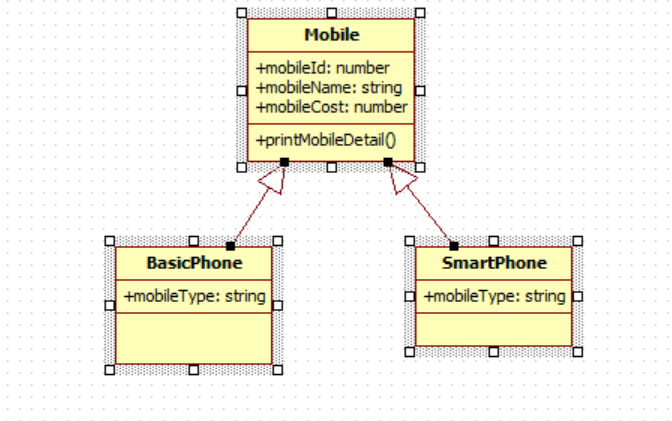
* Create a directory by your name in drive <drive>. In this directory, create a subdirectory angular\_assgn. For each lab exercise create a directory as lab <lab number>.
* You may also look up the on-line help provided in https://v2.angular.io/docs/ts/latest/guide/setup.html

## Learning More

1. Typescript Angular Components & Data Binding

|  |  |
| --- | --- |
| Goals | * Understand the typescript & Angular data binding components |
| Time | 60 minutes |
|  |  |

* 1. Create Typescript class ,as per below mentioned diagram



All 3 classes are in separate file. We have to add mobile details such as id, name, cost & type from different typescript file & print data in console

1. Get Data from array

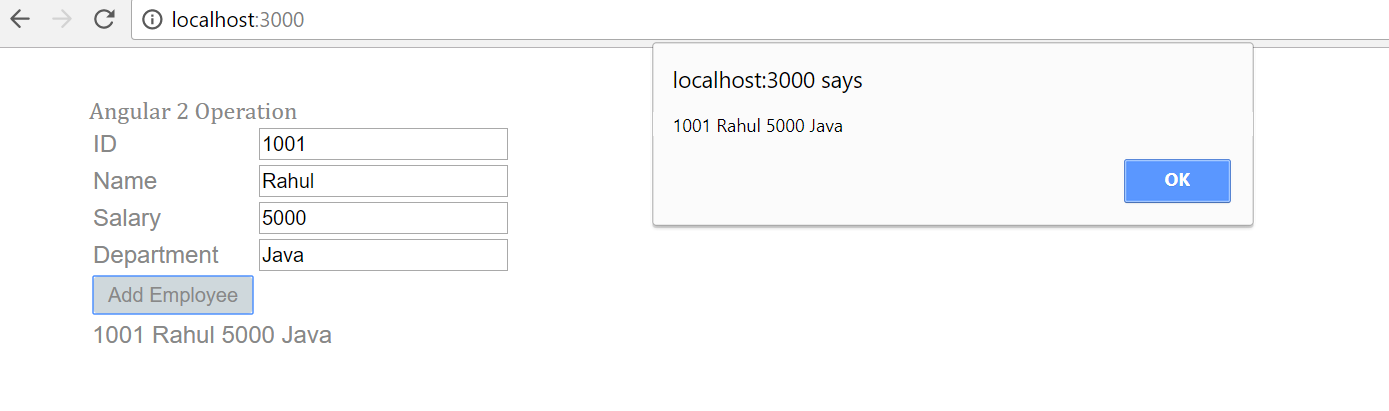
## 1.2: Create a Angular Application & display the result shown below

## 

## When we put some data in ID, name, salary, department it will come automatically come below the submit button shown below

## 

When we click on Add employee alert box will open & display all data inserted as shown in given image

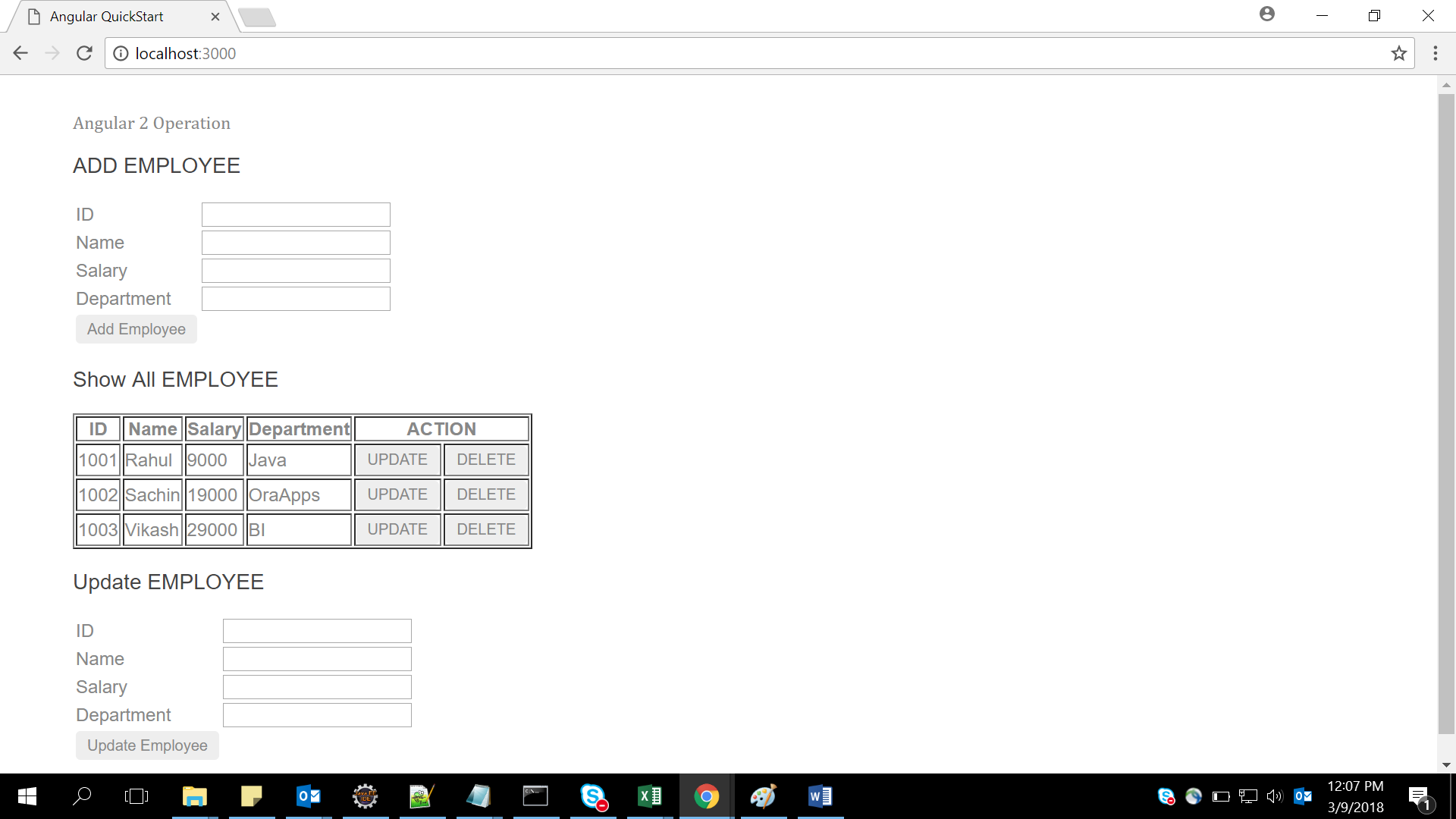


1. Angular Directives & Data Binding

|  |  |
| --- | --- |
| Goals | * Understand the Directives & Data Binding |
| Time | 180 minutes |

## 2.1. Creating a webpage to work with angular

## Create a web page in Angular & display the results shown below.



Show all data comes from array as below :

[

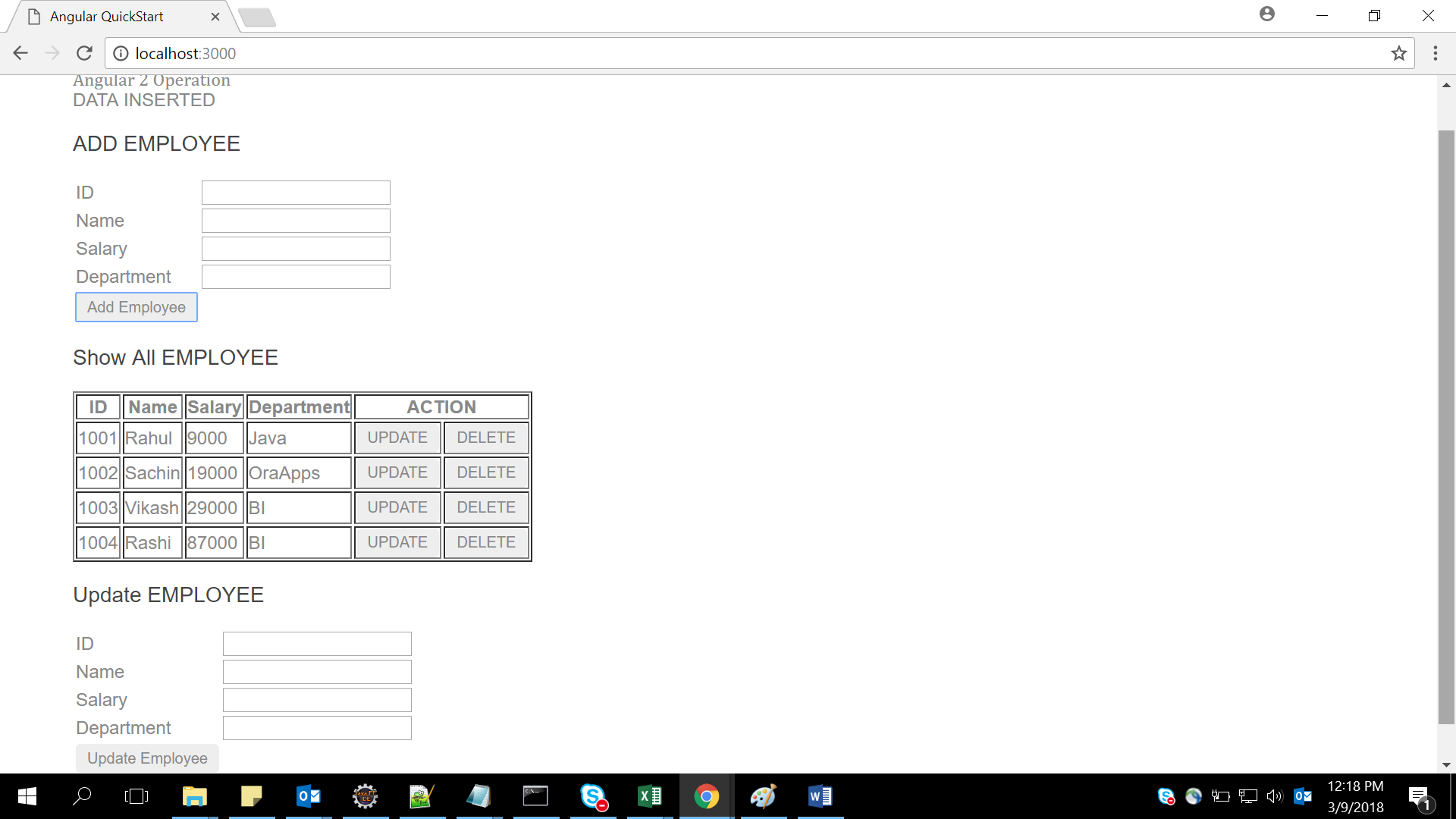
{empId:1001,empName:"Rahul",empSal:9000,empDep:"Java"},

{empId:1002,empName:"Sachin",empSal:19000,empDep:"OraApps"},

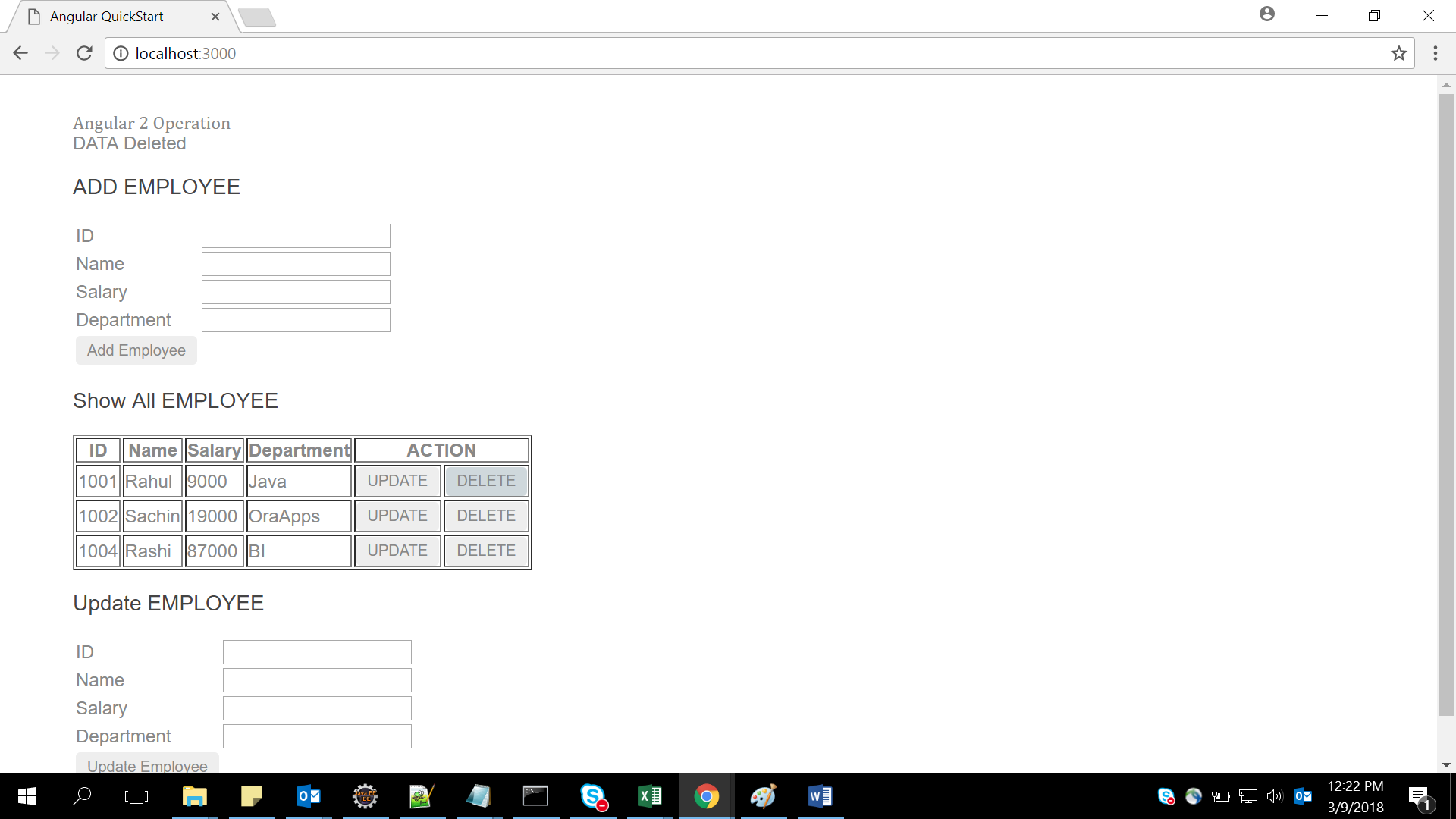
{empId:1003,empName:"Vikash",empSal:29000,empDep:"BI"},

];

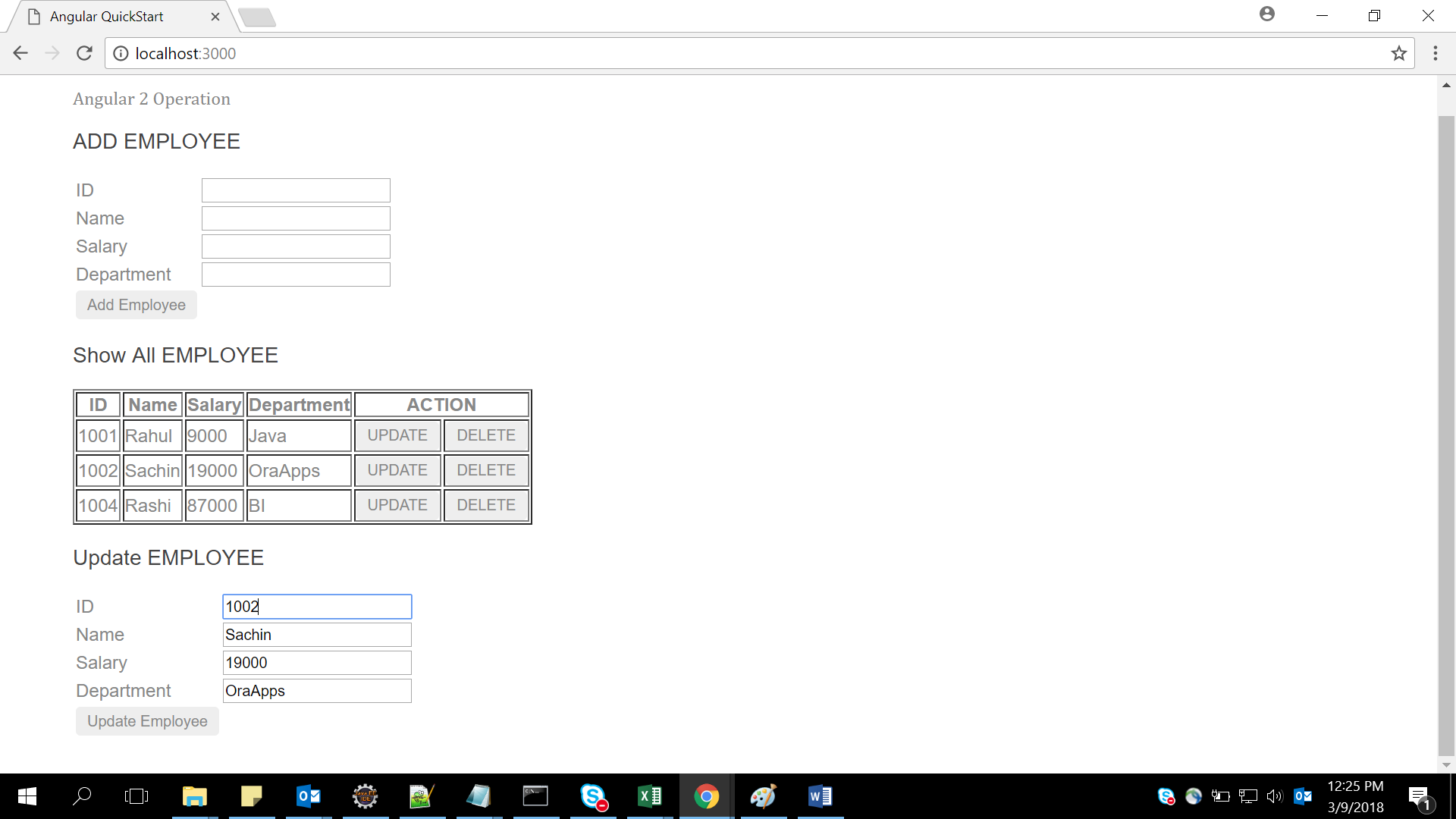
When the user enter Id, name, salary & department in Add Employee & click button Add Employee data will come in Show All Employee & message will display as shown in figure below –example inserted id 1004,name rashi,salary 87000,department BI



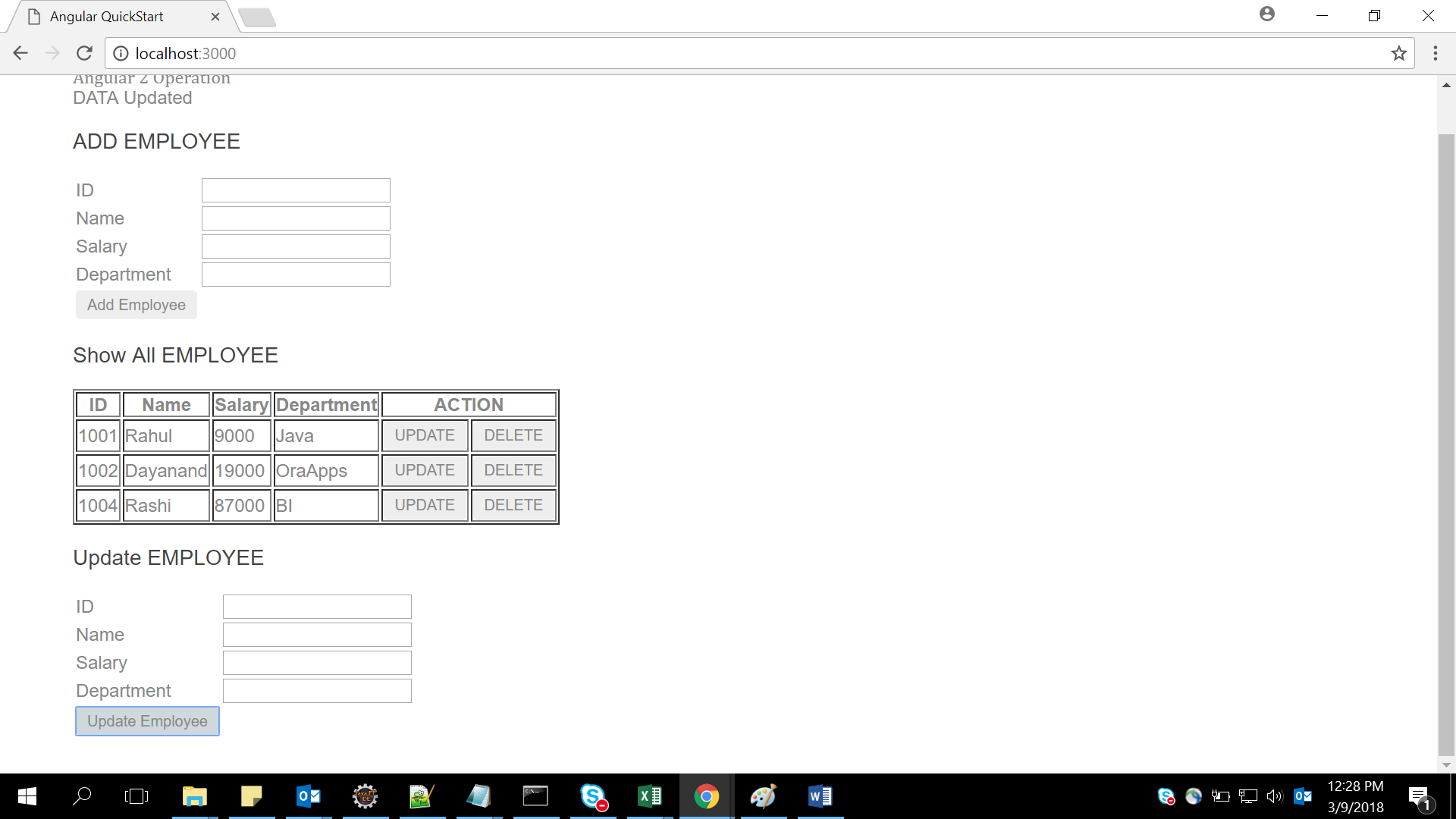
When we click button “Delete” that particular row will be deleted from Show All Employee & message will come ”DATA Deleted” as shown in below figure –Example delete rows 1003 by pressing delete button



When we click button “Update” that particular row will be come in Update employee from Show All Employee & as shown in figure & when updated employee button click the data will update in show All employee & message will display

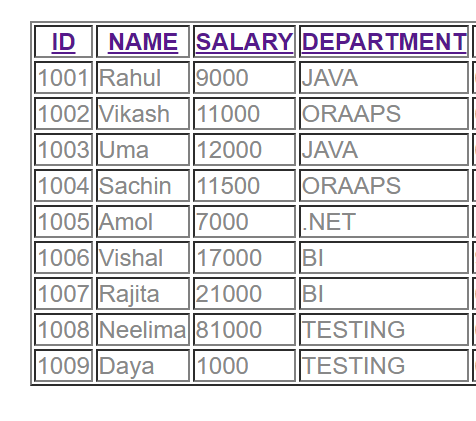


After Update employee button clicked data updated in show all employee & message came



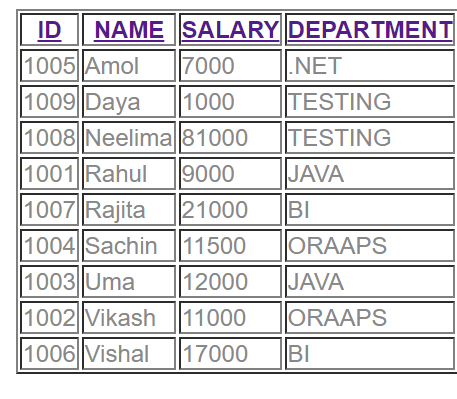
## 2.2. Creating a webpage to work with angular

Create a web page and display the results shown below :



Click on any header --Id, name, salary, department it will display the list in sorted order of that header .

Suppose Clicking on **Name** header so we will get desired output



So by clicking any header data will come in sorted order.

Internal Array: [

{empId:1001,empName:'Rahul',empSal:9000,empDep:'JAVA',empjoiningdate:'6/12/2014'},

{empId:1002,empName:'Vikash',empSal:11000,empDep:'ORAAPS',empjoiningdate:'6/12/2017'},

{empId:1003,empName:'Uma',empSal:12000,empDep:'JAVA',empjoiningdate:'6/12/2010'},

{empId:1004,empName:'Sachin',empSal:11500,empDep:'ORAAPS',empjoiningdate:'11/12/2017'},

{empId:1005,empName:'Amol',empSal:7000,empDep:'.NET',empjoiningdate:'1/1/2018'},

{empId:1006,empName:'Vishal',empSal:17000,empDep:'BI',empjoiningdate:'9/12/2012'},

{empId:1007,empName:'Rajita',empSal:21000,empDep:'BI',empjoiningdate:'6/7/2014'},

{empId:1008,empName:'Neelima',empSal:81000,empDep:'TESTING',empjoiningdate:'6/17/2015'},

{empId:1009,empName:'Daya',empSal:1000,empDep:'TESTING',empjoiningdate:'6/17/2016'} ];

Lab 3: Angular with Form

|  |  |
| --- | --- |
| Goals | * Understand the Angular form |
| Time | 40 minutes |

3. Create a registration form and perform validations on the data entered in the fields. [***Can make use of Angular Template Driven Forms or Model driven Forms***]



**Note:**

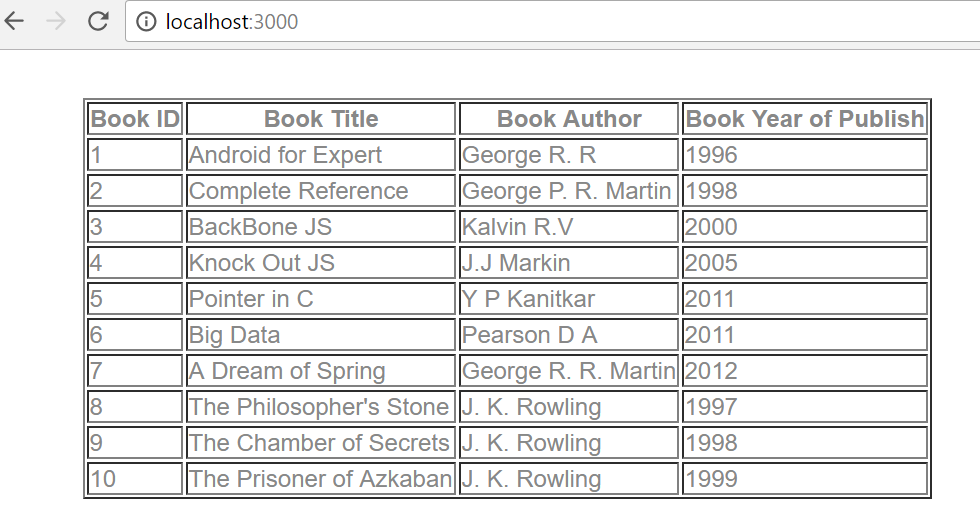
1. Make use of Angular 2 Forms-Model Driven or template driven.
2. All fields are mandatory
3. Product category will be –Grocery, Mobile, Electronics & Cloths
4. When click on Add Product if all fields are validated the output will come on console
5. We can select multiple store in Available list

Lab 4: Angular with Service & pipes

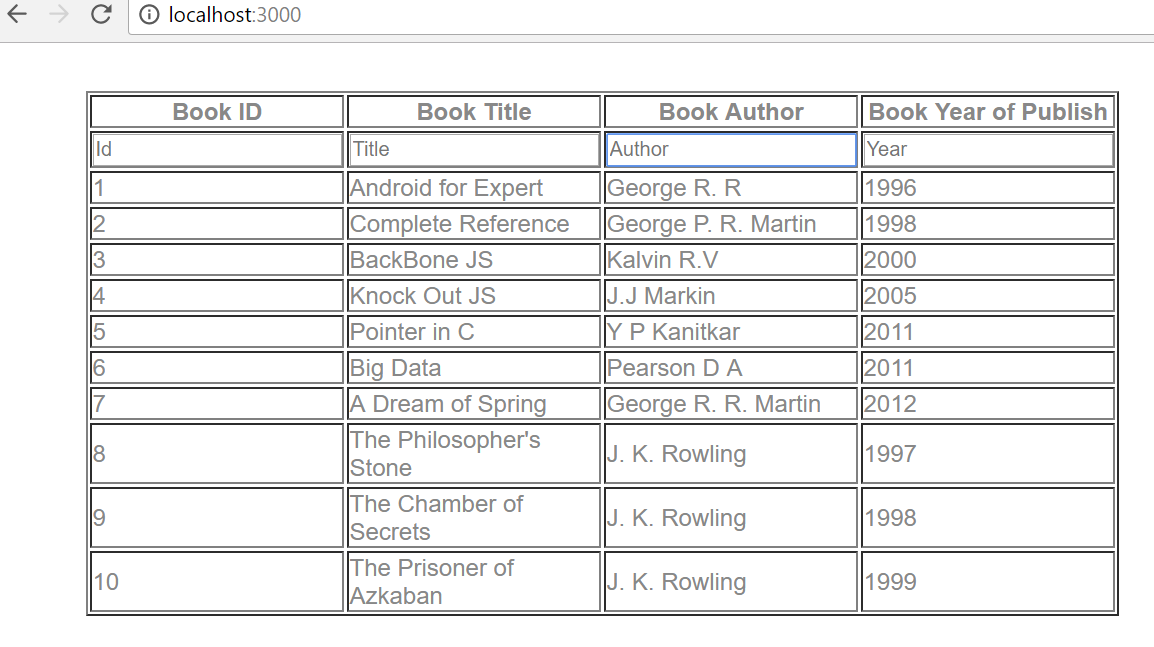
|  |  |
| --- | --- |
| Goals | Working with angular http services & pipes |
| Time | 120 minutes |

## 4.1 Read the booklist.json file with the help of Angular service & display the images shown below;

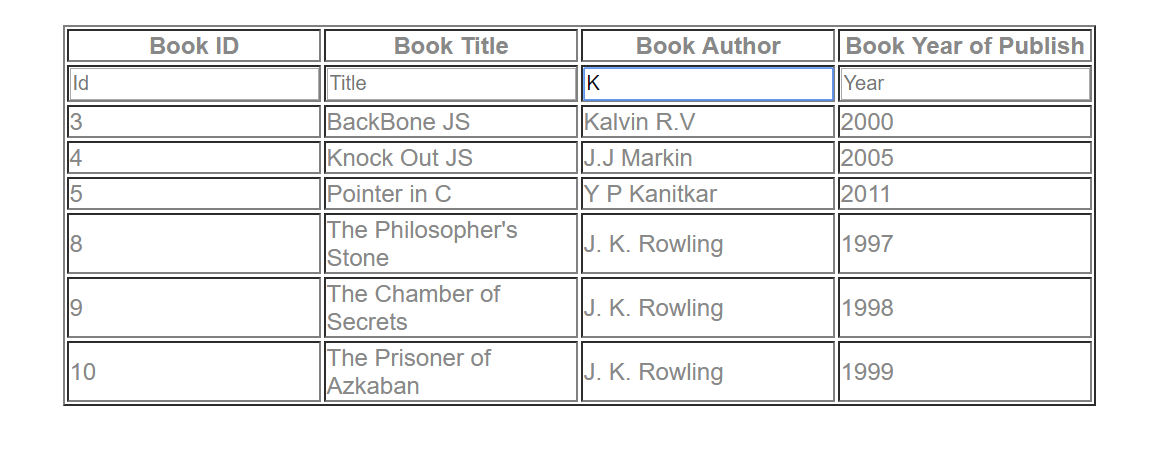
Use <booklist.json>



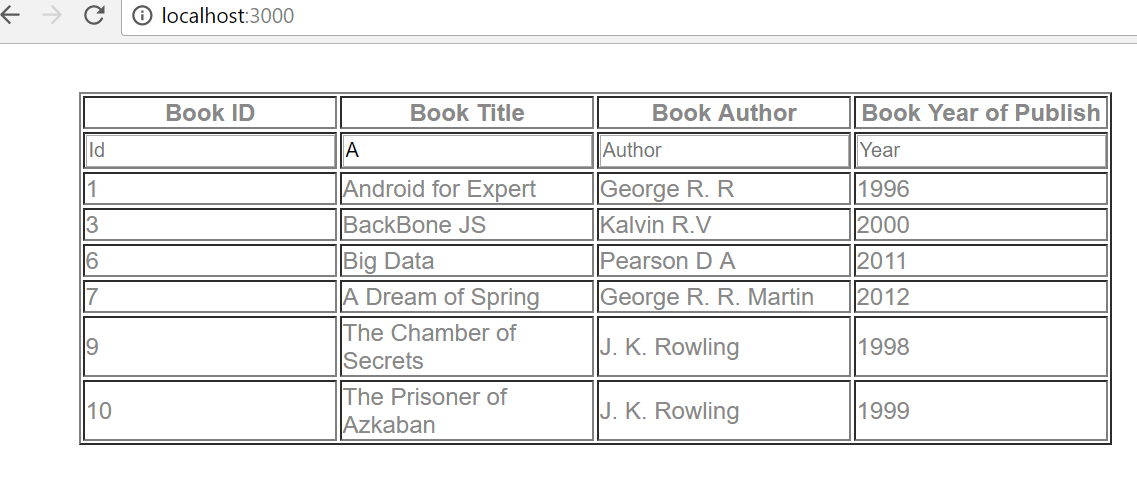
4.2: **Extended assignment 4.1 read booklist.json & apply search filter on id, title, year & author ,as shown in diagram**



If we Search author, id, title, year only searched value will display in output …..searching K in author



Searching Book Title …..A



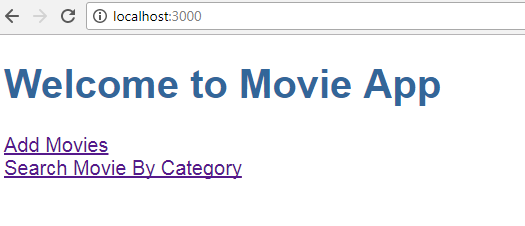
Lab 5: Angular CRUD with Java middle ware application (Spring REST) with Router

|  |  |
| --- | --- |
| Goals | * Angular2 with Spring restful with JPA |
| Time | 240 minutes |

5. Create a “Movie App” wherein movies could be maintained according to genres like Fiction, Drama and Satire. There is also a search option allowing to search for movies.

Use <Movies.java>

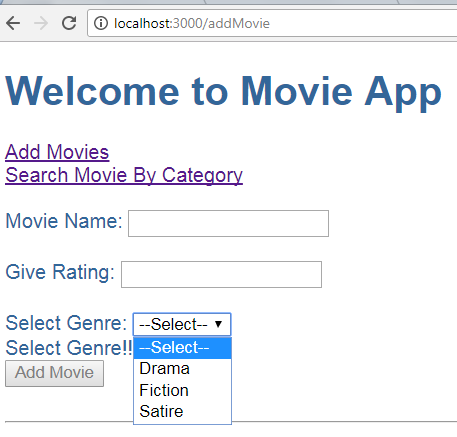
Refer below screen shot:



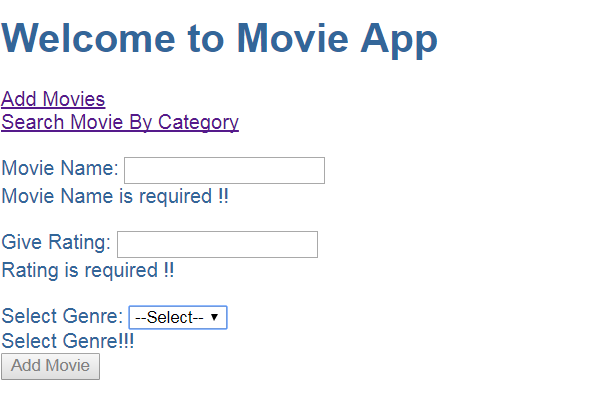
1. When the user clicks on “Add Movies” -> Allows a User to add a movie.

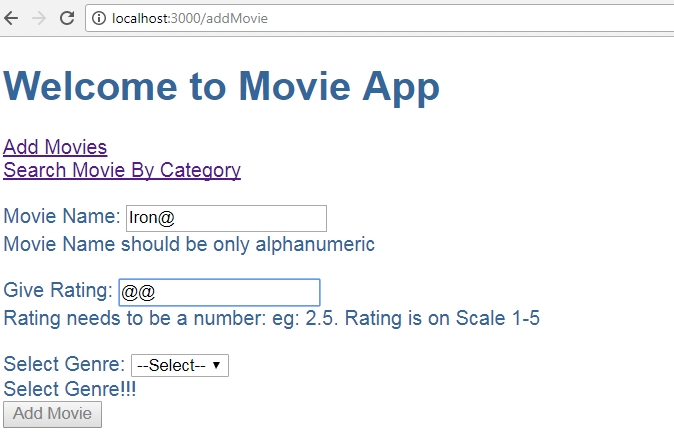
The user can add movie(s) depending on Genre (drop-down list loaded with values [Drama, Fiction, Satire])

The user could add multiple movies. On Adding Movies will be added in Database with the help of Spring Web services (Restful). Use Spring restful with JPA concept Refer below screen shot:



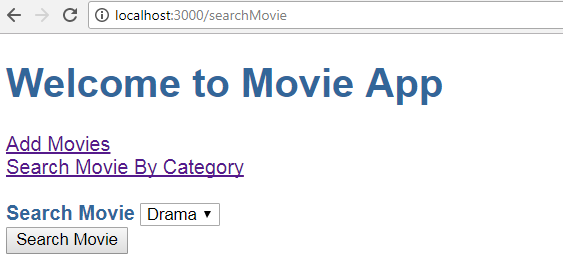
Refer below screens for validations:



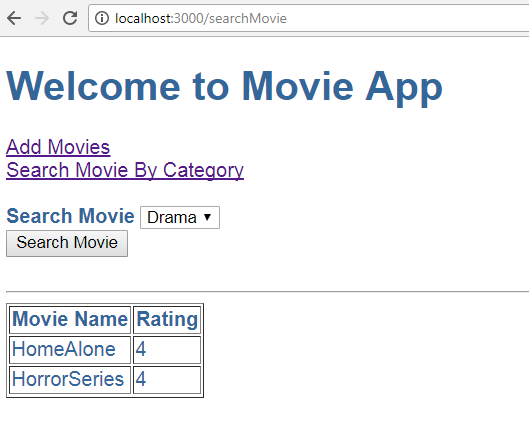


1. When the user clicks on “Search Movie By Category” -> Allows user to search movie(s) by Category. [Depending on Genre (Drama,Fiction,Satire)]

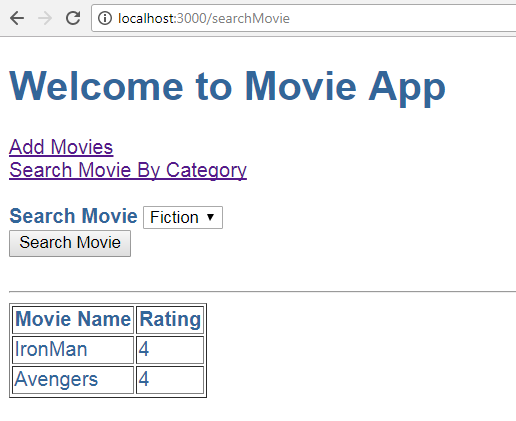
There could be more than one movies depending on same Genre. Refer below screen shots:



Thus if Searching Movie on ‘Drama’ displays below Result:



Thus if Searching Movie on ‘Fiction’ displays below Result:



**Note:**

1. In above assignment movies are added via “Add Movies” and Searched via “Search Movie By Category”. ***These are different components.***
2. Make use of Angular 2 Forms.
3. Make use of services.
4. Make use of Router Module. Which should be created as a separate module and included in the main App Module.
5. Use Spring restful & JPA concept to add movies & search movies.
6. All adding & searching movies done through database

Lab 6: Using RxJS operators convert the above list of user’s data in Uppercase and hire\_date or Joining date format in user friendly manner and subscribe it to view.

Lab 7: Develop the Angular MEAN Application using the Angular 6 as Front End and use Mongo DB as Database and use Express Js and Node JS Server environment.

## Appendix A: Table of Figures

[Figure 1 5](#_Toc452192526)

[Figure 2 5](#_Toc452192527)

[Figure 3 6](#_Toc452192528)

[Figure 4 6](#_Toc452192529)

[Figure 5 7](#_Toc452192530)

[Figure 6 8](#_Toc452192531)

[Figure 7 8](#_Toc452192532)

[Figure 8 8](#_Toc452192533)

[Figure 9 9](#_Toc452192534)

[Figure 10 10](#_Toc452192535)

[Figure 11 10](#_Toc452192536)