# Department of Computing

**CS-344: Web Engineering**

**Class: BESE-9AB**

# Lab 010: Angular 9 Basics

**Date: December 26, 2020**

**Time: 10:00-13:00 & 14:00-17:00**

**14:00-17:00**

# Instructor: Dr. Qaiser Riaz

# 

# Ahmed Hassan Ismail – BESE-9B - 237897

# Lab 09: Angular 9 Basics

### Introduction:

Angular is an MVC (or MVV) based JavaScript framework which is widely used to develop single page web applications. Students have learned basic concepts of Angular 9 in the lectures. This lab will help them to further understand these concepts by developing example applications.

### Lab Objectives:

The objective of this lab is to help the students to setup the environment for Angular 9 and to understand the directory structure of angular 9. The students will familiarize themselves with basic usage of Angular 9. Particularly in this lab the students will learn adding and displaying items in a component using Angular9 as per the given task.

### Tools:

Node JS, npm, Angular cli, Visual Studio Code/or any other editor, browser.

### Helping Material:

Lecture slides.

W3Schools: <https://www.w3schools.io/learn/angular9-introduction/>

### Lab Tasks

**Task 1**

**Installing Node and Angular**

Before getting started with Angular, follow the instructions to install Node.js on your system. Go to Official page of Node.js <https://nodejs.org/en/download/>. Choose LTS installer for your respective systems. If you are a window user, download LTS Windows Installer.

A picture containing website, timeline

Description automatically generated

Click on windows installer.

A picture containing graphical user interface

Description automatically generated

Graphical user interface, text, application

Description automatically generated

**Click on next, accept terms and again click next**

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Node.js is installed on your system. Node.js comes with **npm,** a package manager to install node packages. To check current node and npm versions, open command prompt and type **node -v** and **npm -v.**

Text

Description automatically generated

Once node and npm is installed on your system, it is time to install Angular. To install angular globally on your system, type **npm install global @angular/cli** or **npm i -g @angular/cli.**

Text

Description automatically generated

Text

Description automatically generated

Angular is successfully installed on your system.

**Creating your first angular 9 app**

Open terminal and type **np new [project-name]**

Text

Description automatically generated

**Now it will add the stylesheet you will use & it will create demo app**

Text

Description automatically generated

Graphical user interface

Description automatically generated  
Angular project has been created. Open the folder in your favorite editor.

A screenshot of a computer screen

Description automatically generated

Text

Description automatically generated

Open terminal and type **ng serve –open.** The **–open** command will open this project in your default browser.

Text

Description automatically generated

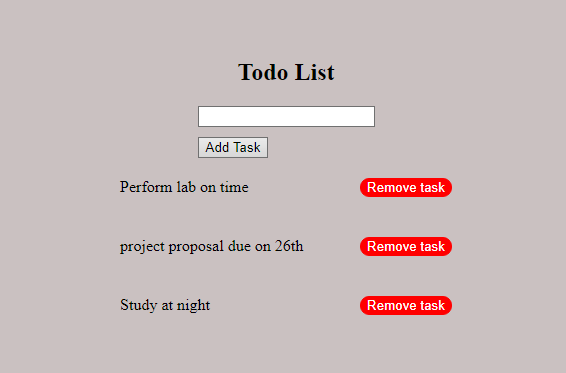
You will see your project running in your default browser.

Graphical user interface

Description automatically generated

### Task 2

Your task is to create a simple to-do-list app in angular.



You are required to implement only add and remove functionality in the app. You should create an input field to take input from the user. When the user clicks on **Add Task** button, the entered text should be added to the list. You are also required to display the contents of the list i.e. to-do tasks, below the input filed. When the user clicks on **Remove Task** button, the task should be removed from the list which contain all the tasks and update the dom.

|  |
| --- |
| Solution |
| Task 1:  Only Screenshot of installed node & npm versions    Task 2:  import { Injectable } from '@angular/core';  import { environment } from 'environments/environment';  import { Http, Response } from '@angular/http';  import { Todo } from './todo';  import { Observable } from 'rxjs/Observable';  import 'rxjs/add/operator/map';  import 'rxjs/add/operator/catch';  import 'rxjs/add/observable/throw';  const API\_URL = environment.apiUrl;  @Injectable()  export class ApiService {  constructor(  private http: Http  ) {  }  public getAllTodos(): Observable<Todo[]> {  return this.http  .get(API\_URL + '/todos')  .map(response => {  const todos = response.json();  return todos.map((todo) => new Todo(todo));  })  .catch(this.handleError);  }  public createTodo(todo: Todo): Observable<Todo> {  return this.http  .post(API\_URL + '/todos', todo)  .map(response => {  return new Todo(response.json());  })  .catch(this.handleError);  }  public getTodoById(todoId: number): Observable<Todo> {  return this.http  .get(API\_URL + '/todos/' + todoId)  .map(response => {  return new Todo(response.json());  })  .catch(this.handleError);  }  public updateTodo(todo: Todo): Observable<Todo> {  return this.http  .put(API\_URL + '/todos/' + todo.id, todo)  .map(response => {  return new Todo(response.json());  })  .catch(this.handleError);  }  public deleteTodoById(todoId: number): Observable<null> {  return this.http  .delete(API\_URL + '/todos/' + todoId)  .map(response => null)  .catch(this.handleError);  }  private handleError (error: Response | any) {  console.error('ApiService::handleError', error);  return Observable.throw(error);  }  }  Task 2 screenshot: |

### Deliverables

Compile a single word document by filling in the solution part and submit this Word file on LMS. You must include your name, ID, and class on first page. The lab grading policy is as follows: The lab is graded between 0 to 10 marks. For some of the labs, students have to present their solutions in a viva session. In case of any problems with submissions on LMS, you should contact your lab engineer Mr. Aftab Hussain , email [aftab.hussain1@seecs.edu.pk](mailto:aftab.hussain1@seecs.edu.pk).