Title: ”Develop and deploy web application using AngularJS”.

1 Problem Statement

Design,develop an application performing arithmetic operations using Angular JS.

2 Learning Objectives

1.To understand in depth working of AngularJS.
2.To use AngularJS to develop any web application.

3 Requirements

Hardware : 64-bit 2.8 GHz processor, 4 GB RAM

Software : 64-bit OS, Web Browser

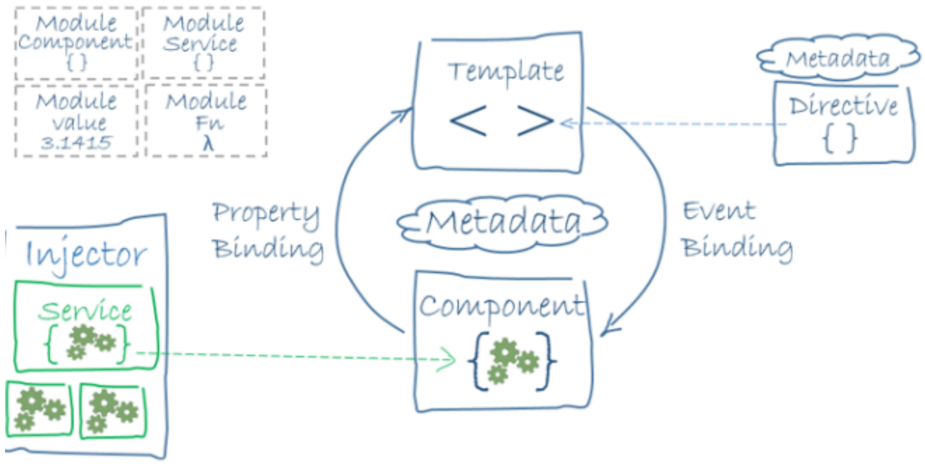
4 Theory

AngularJS AngularJS (commonly referred to as "Angular.js" or "AngularJS 1.X") is a JavaScript-based open-source frontend web application framework mainly maintained by Google and by a community of individuals and corporations to address many of the challenges encountered in developing single-page applications. The JavaScript components complement Apache Cordova, the framework used for developing cross-platform mobile apps. It aims to simplify both the development and the testing of such applications by providing a framework for client-side model–view–controller (MVC) and model–view–viewmodel (MVVM) architectures, along with components commonly used in rich Internet applications.

Angular (commonly referred to as "Angular 4" or "Angular 2") is a TypeScript-based open-source front-end web application platform led by the Angular Team at Google and by a community of individuals and corporations. Angular is a complete rewrite from the same team that built AngularJS.

4.1 Architecture Overview

Angular is a framework for building client applications in HTML and either JavaScript or a language like TypeScript that compiles to JavaScript. The framework consists of several libraries, some of them core and some optional. You write Angular applications by composing HTML templates with Angularized markup, writing component classes to manage those templates, adding application logic in services, and boxing components and services in modules. Then you launch the app by bootstrapping the root module. Angular takes over, presenting your application content in a browser and responding to user interactions according to the instructions you've provided.



4.2 Modules

Angular apps are modular and Angular has its own modularity system called NgModules. NgModules are a big deal. Every Angular app has at least one NgModule class, the root

module, conventionally named AppModule. While the root module may be the only module in a small application, most apps have many more feature modules, each a cohesive block of code dedicated to an application domain, a workflow, or a closely related set of capabilities. An NgModule, whether a root or feature, is a class with an @NgModule decorator.
NgModule is a decorator function that takes a single metadata object whose properties describe the module.
The most important properties are:
1.declarations - the view classes that belong to this module. Angular has three kinds of view classes: components, directives, and pipes.
2.exports - the subset of declarations that should be visible and usable in the component templates of other modules.
3.imports - other modules whose exported classes are needed by component templates declared in this module.
4. providers - creators of services that this module contributes to the global collection of services; they become accessible in all parts of the app.
5.bootstrap - the main application view, called the root component, that hosts all other app views. Only the root module should set this bootstrap property.

4.3 Components

A component controls a patch of screen called a view.
For example, the following views are controlled by components:
1.The app root with the navigation links.
2.The list of heroes.
3.The hero editor.
You define a component's application logic—what it does to support the view—inside a class. The class interacts with the view through an API of properties and methods.

4.4 Templates

You define a component's view with its companion template. A template is a form of HTML that tells Angular how to render the component.

4.5 Metadata

Metadata tells Angular how to process a class.

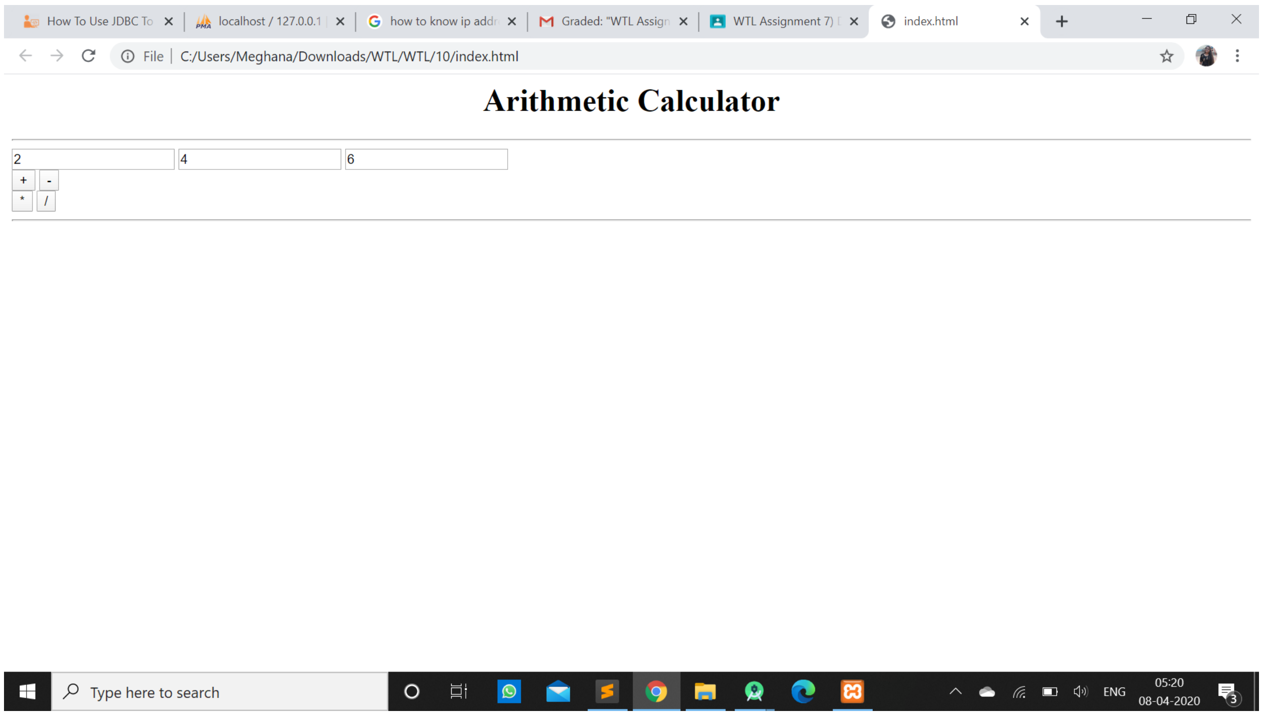
4.6 Data binding

Without a framework, you would be responsible for pushing data values into the HTML controls and turning user responses into actions and value updates. Writing such push/pull logic by hand is tedious, error-prone, and a nightmare to read as any experienced jQuery programmer can attest. Angular supports data binding, a mechanism for coordinating parts of a template with parts of a component. Add binding markup to the template HTML to tell Angular how to connect both sides.

4.7 Directives

Angular templates are dynamic. When Angular renders them, it transforms the DOM according to the instructions given by directives. A directive is a class with a @Directive decorator. A component is a directivewith-a-template; a @Component decorator is actually a @Directive decorator extended with template-oriented features. Two other kinds of directives exist: structural and attribute directives. They tend to appear within an element tag as attributes do, sometimes by name but more often as the target of an assignment or a binding. Structural directives alter layout by adding, removing, and replacing elements in DOM.

5 Output



Arithmetic Calculator

```
2    4    6
+ -
* /
```

6 Conclusion

Here, we learnt Architecture, modules, components and so many concept of Angular JS. and we have studied how to develop web application using Angular JS.