MALAD KANDIVALI EDUCATION SOCIETY'S

NAGINDAS KHANDWALA COLLEGE OF COMMERCE, ARTS & MANAGEMENT STUDIES & SHANTABEN NAGINDAS KHANDWALA COLLEGE OF SCIENCE
MALAD [W], MUMBAI – 64
AUTONOMOUS INSTITUTION
(Affiliated To University Of Mumbai)
Reaccredited 'A' Grade by NAAC | ISO 9001:2015 Certified

CERTIFICATE

Name: **Dikshita Shetty**

Roll No: **385** Programme: BSc IT/CS Semester: III

This is certified to be a bonafide record of practical works done by the above student in the college laboratory for the course **Hybrid Application Development (classcode: 2037UCSMD)** for the partial fulfilment of Third Semester of BSc IT/CS during the academic year 2020-21.

The journal work is the original study work that has been duly approved in the year 2020-21 by the undersigned.

External Examiner

Mr. Gangashankar Singh
(Subject-In-Charge)

Date of Examination: (College Stamp)

Class: S.Y. B.Sc. IT Sem- III **Roll No: 385**

Subject: Hybrid Application Development

INDEX

PRACTICAL 1

AIM: AngularJS Data Binding.

THEORY:

AngularJS Data Binding

Data binding is a very useful and powerful feature used in software development technologies. It acts as a bridge between the view and business logic of the application. The data binding is the data synchronization processes that work between the model and view components. In Angular, model treat as source of application and view is the projection of angular model.

In angularjs when model data got changed that time the view data will change automatically and vice versa.
We have two types of data bindings available in angularjs those are

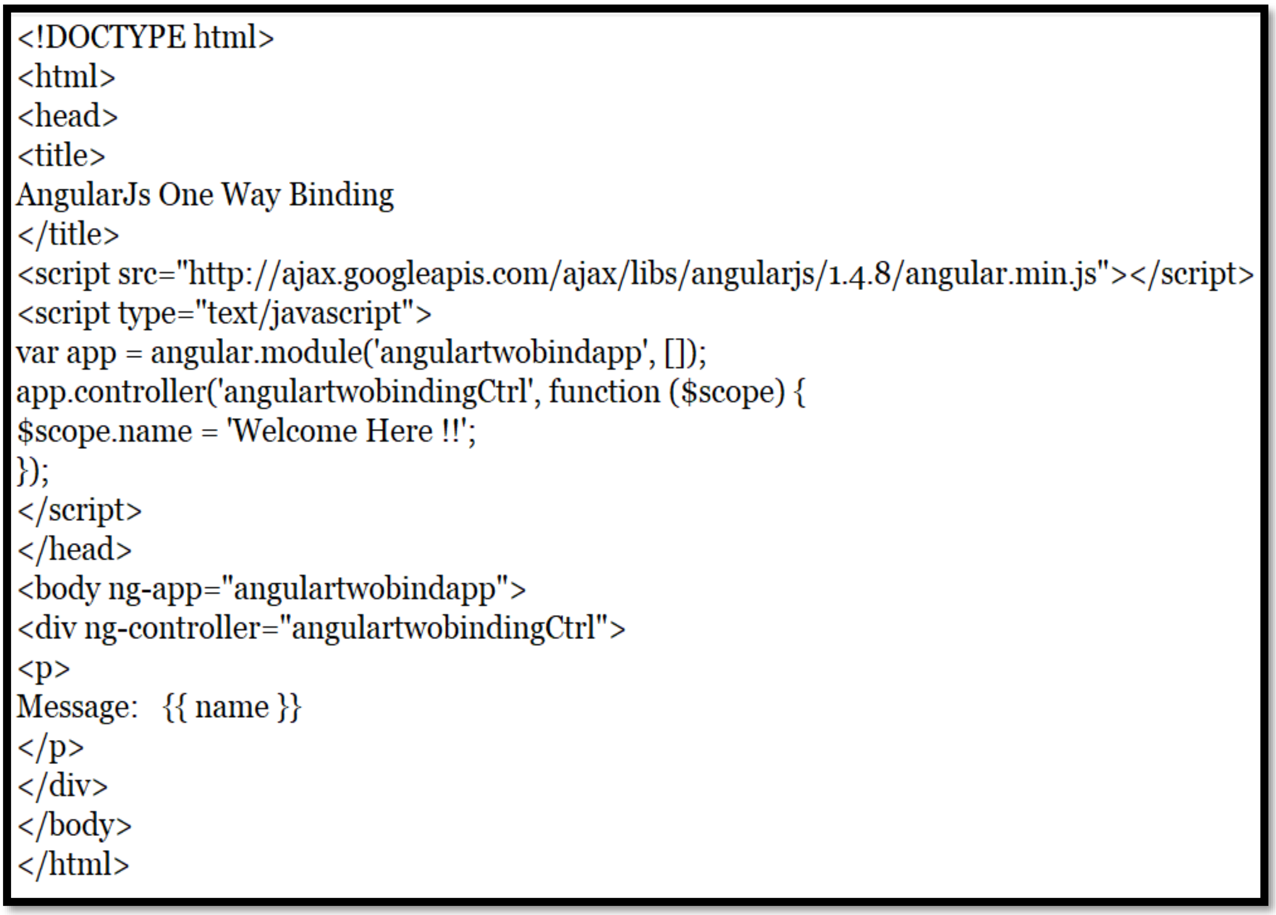
- One-Way data binding
- Two-Way data binding

One-Way Data Binding

The one-way data binding is an approach where a value is taken from the data model and inserted into an HTML element. There is no way to update model from view. It is used in classical template systems. These systems bind data in only one direction.

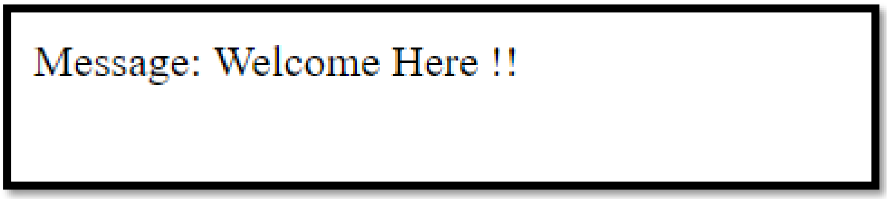
In the below program we have already given a message to be displayed which is Welcome Here!!! Here we are binding model values to html elements using data bindings, but html elements, it won't change the values in model we need to write custom code to make it updated every time. This is known as One-Way data binding.

CODE FOR ONE WAY BINDING:



```
<!DOCTYPE html>
<html>
<head>
<title>
AngularJs One Way Binding
</title>
<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>
<script type="text/javascript">
var app = angular.module('angulartwobindapp', []);
app.controller('angulartwobindingCtrl', function ($scope) {
$scope.name = 'Welcome Here !!';
});
</script>
</head>
<body ng-app="angulartwobindapp">
<div ng-controller="angulartwobindingCtrl">
<p>
Message:   {{ name }}
</p>
</div>
</body>
</html>
```

OUTPUT:



```
Message: Welcome Here !!
```

Two-Way Data Binding

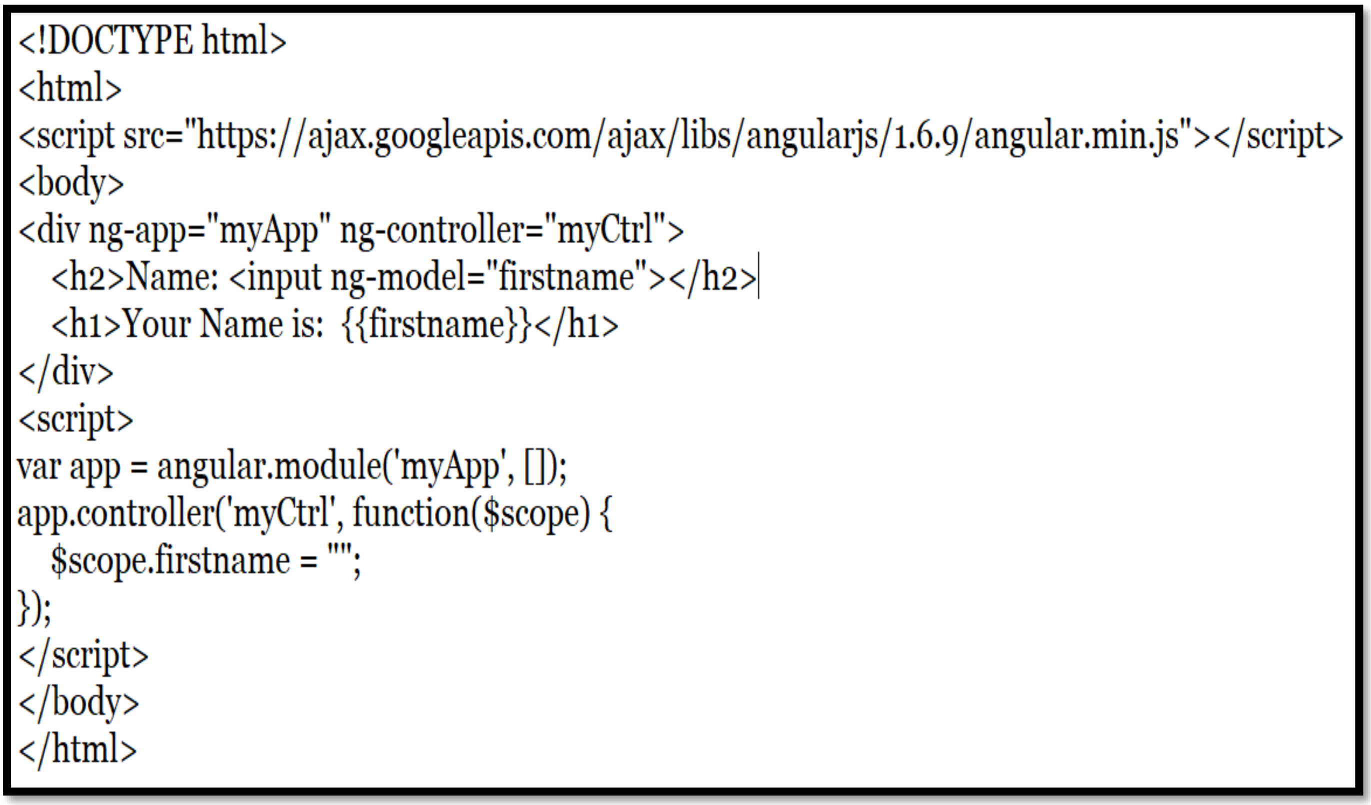
Data-binding in Angular apps is the automatic synchronization of data between the model and view components.

Data binding lets you treat the model as the single-source-of-truth in your application. The view is a projection of the model at all times. If the model is changed, the view reflects the change and vice versa.

In the below program we defined ng-model objective to html control and used same ng-model value to show input control value. Here whenever we change input control value, automatically the appearance value also will get changed.

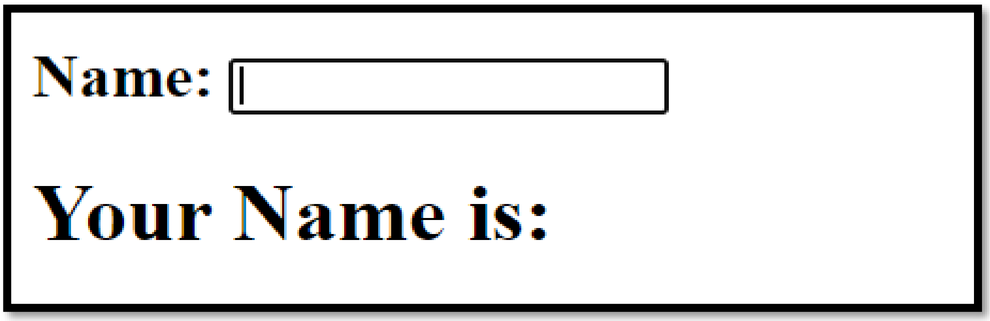
Which means as soon as you write or give input it will automatically display down as we have used expression there. This is known as Two-Way binding.

CODE FOR TWO WAY BINDING:

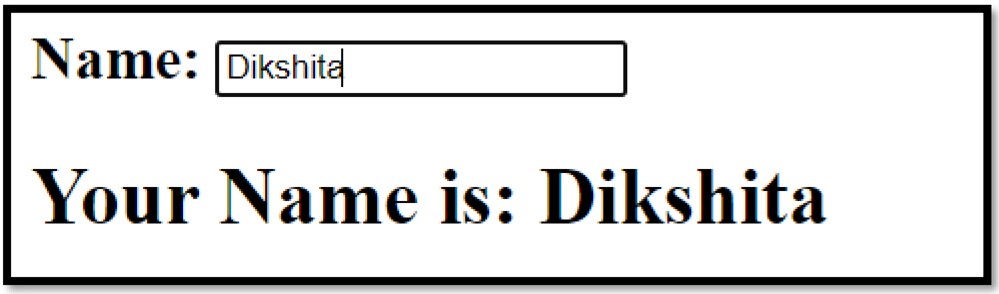


```
<!DOCTYPE html>
<html>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
<body>
<div ng-app="myApp" ng-controller="myCtrl">
   <h2>Name: <input ng-model="firstname"></h2>
   <h1>Your Name is:  {{firstname}}</h1>
</div>
<script>
var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope) {
   $scope.firstname = "";
});
</script>
</body>
</html>
```

OUTPUT:



```
Name:

Your Name is:
```



```
Name: Dikshita

Your Name is: Dikshita
```

PRACTICAL 2

AIM: AngularJS Directives.

THEORY:

AngularJS Directives

AngularJs directives are an extension over the HTML elements and that allow you to extend the HTML elements behavior by adding special attribute prefix 'ng-'.

AngularJS is having set of built-in directives that allow us to implement required functionality by using HTML elements in application.

AngularJS Built-in Directives

The following are the few built-in directives available in angularjs.

- **AngularJS ng-app Directive**

Generally ng-app directive in angularjs is used to define the starting point of application and it will act as the root element for application. The ng-app in angularjs will identify which part of HTML needs to use for angularjs app. Generally once angularjs framework initialized it will check if HTML contains ng-app directive or not in case if it found it will compile that respective HTML template. If we define ng-app in root element either <html> or <body> of html document then it will control complete html dom elements.

- **AngularJS ng-init Directive**

The ng-init directive in angularjs is used to initialize application data during the starting of application. Generally, in angularjs application ng-init directive is used to define a local variable with value and we can use that variable value in application wherever we required.

- **AngularJS ng-model Directive**

In angularjs ng-model directive is used to get value of input controls like textbox, label, etc and bind that value to application data.

- **AngulartJS ng-repeat Directive**

The ng-repeat directive in angularjs is used to loop through items in collection element and it will act as for loop.

- **AngularJS ng-bind Directive**

In angularjs, the ng-bind directive is used to bind/replace the text content of any particular HTML element with the value of a given expression that is used against the ng-bind in angularjs application. The value of the specified HTML element will be updated whenever the value of the given expression will change in angularjs ng-bind directive.

- **AngularJS ng-show and ng-hide Directives**

In angularjs, ng-show and ng-hide directives are used to take control of displaying HTML elements in our applications. By using ng-show or ng-hide directives, we can show or hide HTML elements depending on the value of their data model values.

- **AngularJS ng-switch Directive**

In angularjs, ng-switch directive will behave same as a switch-case statement in general programming scenario. In angularjs, ng-switch is useful to conditionally swap between the HTML DOM elements based on the data model written in the expression.

- **AngularJS ng-if Directive**

In angularjs, the ng-if directive is used to remove and reinsert a portion of the HTML DOM element into an HTML template tree. This is executed based on the value of expression assigned to the ng-if attribute.

- **AngularJS ng-include Directive**

In angularjs, the ng-include directive will compile and include one HTML page in another page. Since we know that embedding one HTML page in another HTML page is not supported in HTML to achieve this we need to follow some custom approaches like Ajax and including HTML via server-side programming.

- **AngularJS ng-cloak Directive**

The ng-cloak directive in angularjs is a special type of directive which is used to prevent showing an un-compiled form of elements while page is loading. We can say that ng-cloak directive will hold and wait till operations complete to show the elements on page.

By using ng-cloak directive in angularjs, we can prevent or avoid undesirable flicker effect to display elements on page. We can use ng-cloak directive with <body> element or we can use it in a required portion of HTML elements.

- **AngularJS ng-view Directive**

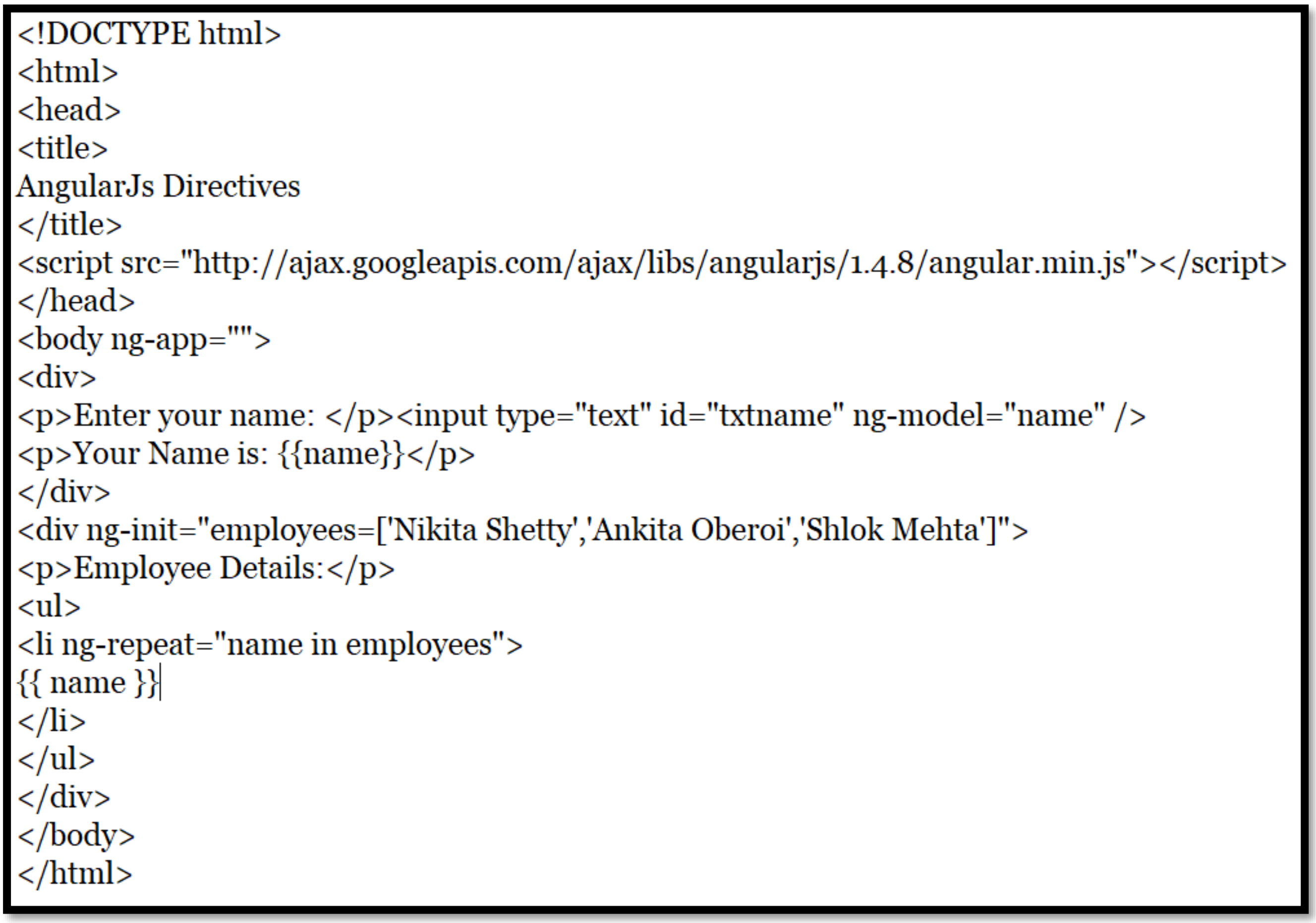
In angularjs, ng-view directive is used to switch between the views in angularJs application. Generally, we will use ng-view directive with route service to change the views based on the defined conditions in angular applications.

- **AngularJS ng-template Directive**

In angularjs, the ng-template directive is used to load the content of a <script> element into $templateCache, and this is done further by using ng-include, ng-view, template, etc. There is one rule which we need to follow when we use ng-template directive in angularjs application. In ng-template the type of the <script> element must be specified as text/ng-template, and a cache name for the template must be assigned through the element's id, which can then be used as a directive's templateUrl.

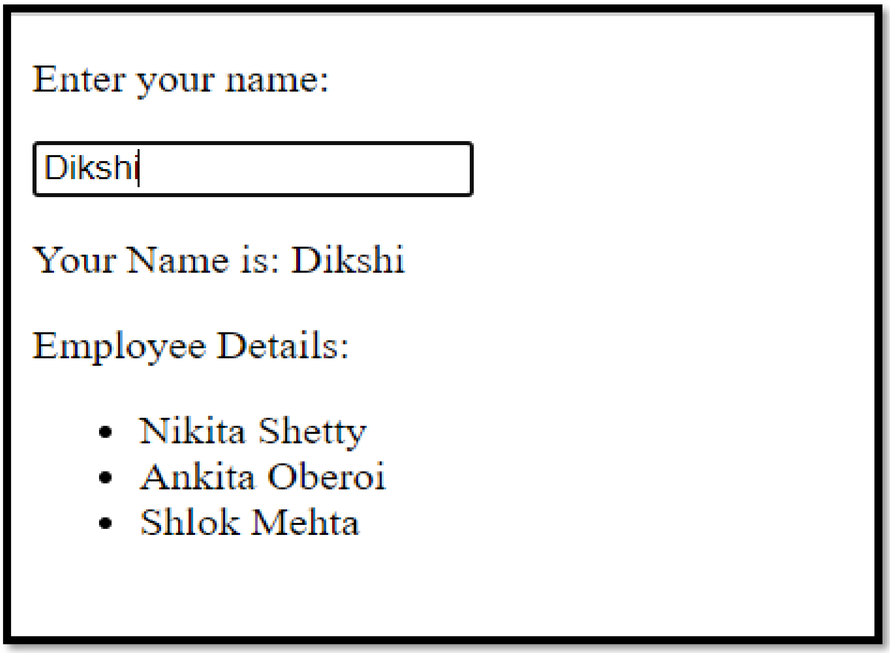
In the below program we have used some of the in-built AngularJS directives such as ng-model, ng-init, ng-repeat. We used ng-repeat which will loop in the employees list to display employee details. And ng-model is used to bind the data entered by the user in the text box. This is how we use directives.

CODE:



```
<!DOCTYPE html>
<html>
<head>
<title>
AngularJs Directives
</title>
<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>
</head>
<body ng-app="">
<div>
<p>Enter your name: </p><input type="text" id="txtname" ng-model="name" />
<p>Your Name is: {{name}}</p>
</div>
<div ng-init="employees=['Nikita Shetty','Ankita Oberoi','Shlok Mehta']">
<p>Employee Details:</p>
<ul>
<li ng-repeat="name in employees">
{{ name }}
</li>
</ul>
</div>
</body>
</html>
```

OUTPUT:



Enter your name:

Dikshi

Your Name is: Dikshi

Employee Details:

- Nikita Shetty
- Ankita Oberoi
- Shlok Mehta

PRACTICAL 3

AIM: AngularJS Controllers.

THEORY:

AngularJS Controller

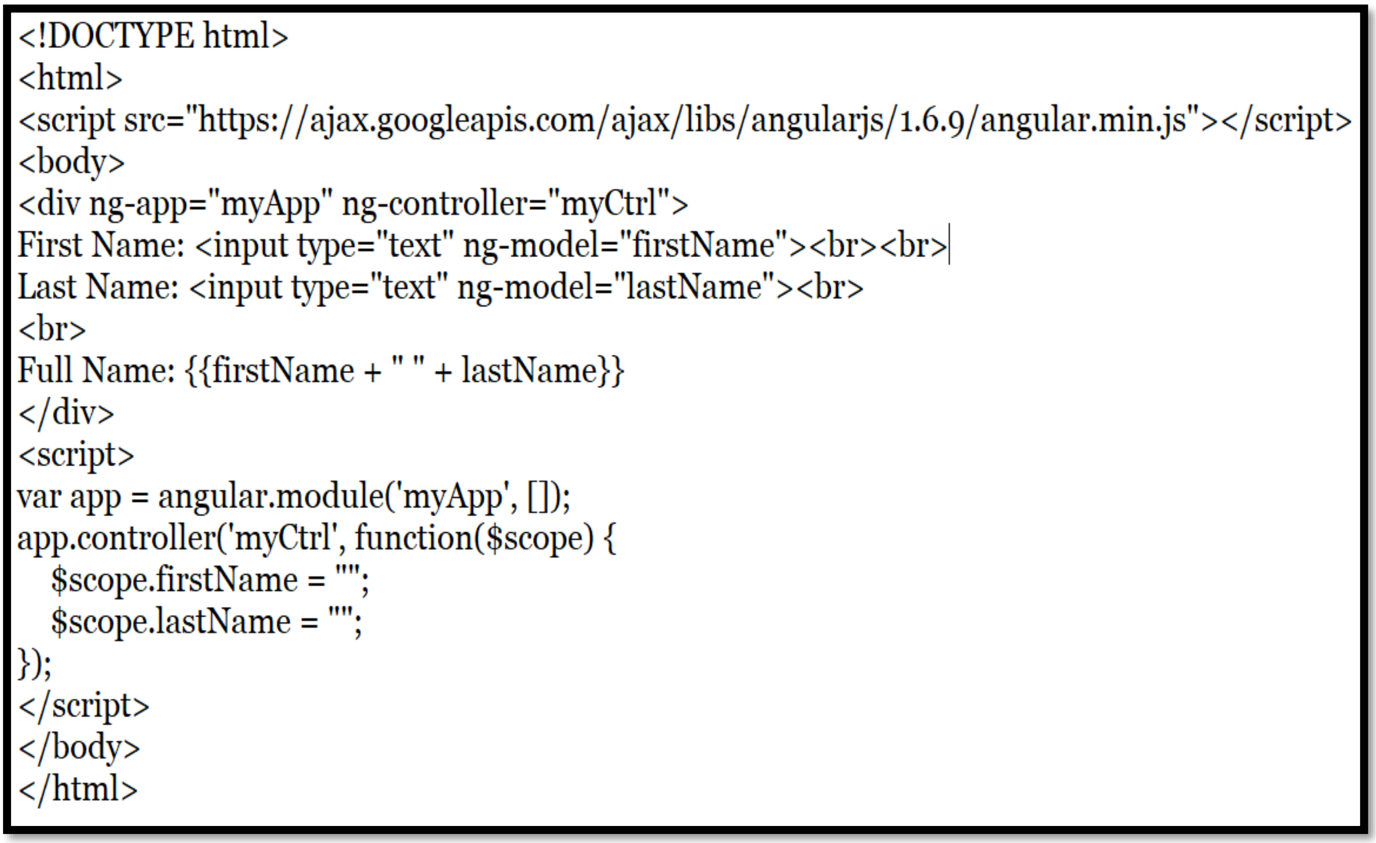
The controller in AngularJS is a JavaScript function that maintains the application data and behavior using $scope object.

You can attach properties and methods to the $scope object inside a controller function, which in turn will add/update the data and attach behaviours to HTML elements. The $scope object is a glue between the controller and HTML.

The ng-controller directive is used to specify a controller in HTML element, which will add behavior or maintain the data in that HTML element and its child elements.

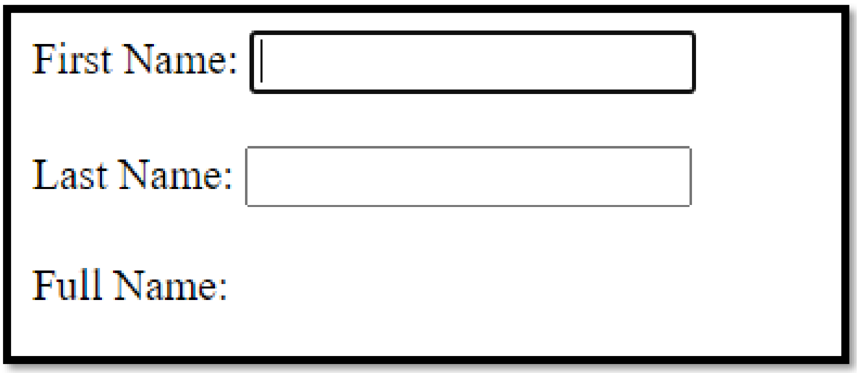
In the below program the AngularJS application is defined by ng-app="myApp"**.**
The application runs inside the <div>.
The ng-controller="myCtrl" attribute is an AngularJS directive. It defines a controller.
The myCtrl function is a JavaScript function.
AngularJS will invoke the controller with a $scope object.
In AngularJS, $scope is the application object (the owner of application variables and functions).
The controller creates two properties (variables) in the scope (firstName and lastName).
The ng-model directives bind the input fields to the controller properties (firstName and lastName).

CODE:



```
<!DOCTYPE html>
<html>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
<body>
<div ng-app="myApp" ng-controller="myCtrl">
First Name: <input type="text" ng-model="firstName"><br><br>
Last Name: <input type="text" ng-model="lastName"><br>
<br>
Full Name: {{firstName + " " + lastName}}
</div>
<script>
var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope) {
   $scope.firstName = "";
   $scope.lastName = "";
});
</script>
</body>
</html>
```

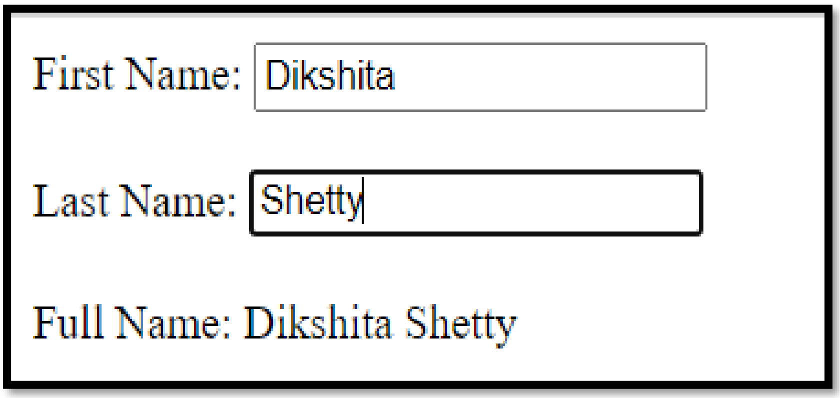
OUTPUT:



First Name:

Last Name:

Full Name:



First Name: Dikshita

Last Name: Shetty

Full Name: Dikshita Shetty

PRACTICAL 4

AIM: AngularJS Events.

THEORY:

AngularJS Events

Generally while developing applications we use different type of html DOM events like mouse clicks, key press, change events, etc likewise angularjs is having its own event directives for DOM interactions.

AngularJS Event Directives

In angularjs we have different type of DOM event listener directives are available and we can attach those events to html elements. Following are the different type of event listeners in angularjs.

- **AngularJS ng-click Event Directive**

In angularjs ng-click directive event / function is used to define an angular click event for html elements. Suppose if you want to add and fire a click event on HTML button click that time we need to use this event.

Syntax of AngularJS ng-click Event Directive:

```
<element ng-click="expression">
...your code...
</element>
```

- **AngularJS ng-dblclick Event**

In angularjs ng-dblclick event directive is used to define double click event for html elements. In case if you want to fire a function or other event on double click of HTML element then we need to use this event.

Syntax of AngularJS ng-dblclick Event / Function:

```
<element ng-dblclick="expression">
...your code...
</element>
```

- **AngularJS ng-blur Event Directive**

In angularjs ng-blur event is used to fire validation or call function when input element lost its focus or comes outside of element. In case if you want to call a function or fire other event whenever we come out of input element then this ng-blur event is very useful for us.

Syntax of AngularJS ng-blur Event Directive:

```
<element ng-blur="expression">

...your code...

</element>
```

- **AngularJS ng-change Event Directive**

In angularjs ng-change event is used to raise or call function whenever the value of input element changes. We can use this angularjs ng-change event with input elements like textboxes, checkboxes, dropdowns and textarea control elements. Generally this angularjs ng-change event will trigger for every change in the value of input controls and it’s very useful for us to call a function or raise event whenever input text changes.

Syntax of AngularJS ng-change Event:

```
<element ng-change="expression">

...your code...

</element>
```

- **AngularJS ng-cut, ng-copy, ng-paste Events**

In angularjs ng-cut, ng-copy and ng-paste events are used to define custom behaviour functions during cut or copy or paste text in input text fields. Suppose if you want to raise or call some custom functions while performing cut or copy or paste events on input text field in angularjs applications it’s better to use ng-cut, ng-copy and ng-paste events.

Syntax of AngularJS ng-cut, ng-copy, ng-paste Events:

```
<element ng-cut="expression" ng-copy="expression" ng-paste="expression">
--Your Code--
</element>
```

- **AngularJS ng-focus Event Directive**

In angularjs ng-focus event directive is used to define or execute custom behaviour functions whenever input field gets focus. Suppose in angularjs if you want to change textbox colour on focus or call some custom functions on input field focus it’s better to use ng-focus event.

Syntax of AngularJS ng-focus Events:

```
<element ng-focus="expression">
--Your Code--
</element>
```

- **AngularJS ng-keydown Event Directive**

In angularjs ng-keydown event directive is used to define events or execute custom behaviour functions immediately on key press in keyboard. The ng-keydown event will raise immediately whenever we press key on keyboard before release itself. Suppose in angularjs if you want to raise an event on key press or call some custom functions immediately on key press it’s better to use ng-keydown event.

Syntax of AngularJS ng-keydown Event:

```
<element ng-keydown="expression">
--Your Code--
</element>
```

- **AngularJS ng-keyup Event Directive**

In angularjs ng-keyup event directive is used to raise or call events / custom functions on keyup event. The ng-keyup event will call custom events / functions on keyup (once we press and release key in keyboard).

Suppose in angularjs application if you want to raise an event on keyup or call some custom functions immediately on key up it's better to use ng-keyup event.

Syntax of AngularJS ng-keyup Event:

```
<element ng-keyup="expression">
--Your Code--
</element>
```

- **AngularJS ng-mousedown Event Directive**

In angularjs ng-mousedown event directive is used to raise or call events / custom functions on mouse click. The ng-mousedown event in angularjs will fire on mouse click and execute custom events / functions based on our requirements. Suppose in angularjs application if you want to raise an event on mouse click or call some custom functions immediately on mouse click it's better to use ng-mousedown event.

Syntax of AngularJS ng-mousedown Event:

```
<element ng-mousedown="expression">
--Your Code--
</element>
```

- **AngularJS ng-mouseenter Event Directive**

In angularjs ng-mouseenter event directive is used to raise or call events / functions whenever mouse cursor or pointer enters on html elements. The ng-mouseenter event in angularjs will fire whenever our mouse cursor enters an element and execute custom events / functions based on our requirements. Suppose in angularjs application if you want to execute custom events / functions whenever mouse cursor moved on html elements then it's better to use ng-mouseenter event.

Syntax of AngularJS ng-mouseenter Event:

```
<element ng-mouseenter="expression">
--Your Code--
</element>
```

- **AngularJS ng-mouseleave Event Directive**

In angularjs ng-mouseleave event directive is used to execute or raise events / functions when the mouse cursor leaves from an elements. The ng-mouseleave event in angularjs will fire an events whenever mouse cursor move away from an element. Suppose in angularjs application if you want to raise an event / function whenever mouse cursor leaves an elements then it’s better to use ng-mouseleave event.

Syntax of AngularJS ng-mouseleave Event:

```
<element ng-mouseleave="expression">
--Your Code--
</element>
```

- **AngularJS ng-mousemove Event Directive**

In angularjs ng-mousemove event directive is used to execute or raise custom events / functions when mouse cursor moving on html elements. The ng-mousemove event in angularjs will fire an events while moving mouse cursor over an element. Suppose in angularjs application if you want to raise an event / function while moving mouse cursor over html elements then it’s better to use ng-mousemove event.

Syntax of AngularJS ng-mousemove Event:

```
<element ng-mousemove="expression">
--Your Code--
</element>
```

- **AngularJS ng-mouseover Event Directive**

In angularjs ng-mouseover event directive is used to execute or raise custom events / functions whenever mouse cursor hover on html elements. The ng-mouseover event in angularjs will fire an events whenever mouse cursor hover on element. Suppose in angularjs application if you want to raise an event / function whenever mouse cursor hover an elements then it’s better to use ng-mouseover event.

Syntax of AngularJS ng-mouseover Event:

```
<element ng-mouseover="expression">
--Your Code--
</element>
```

- **AngularJS ng-mouseup Event Directive**

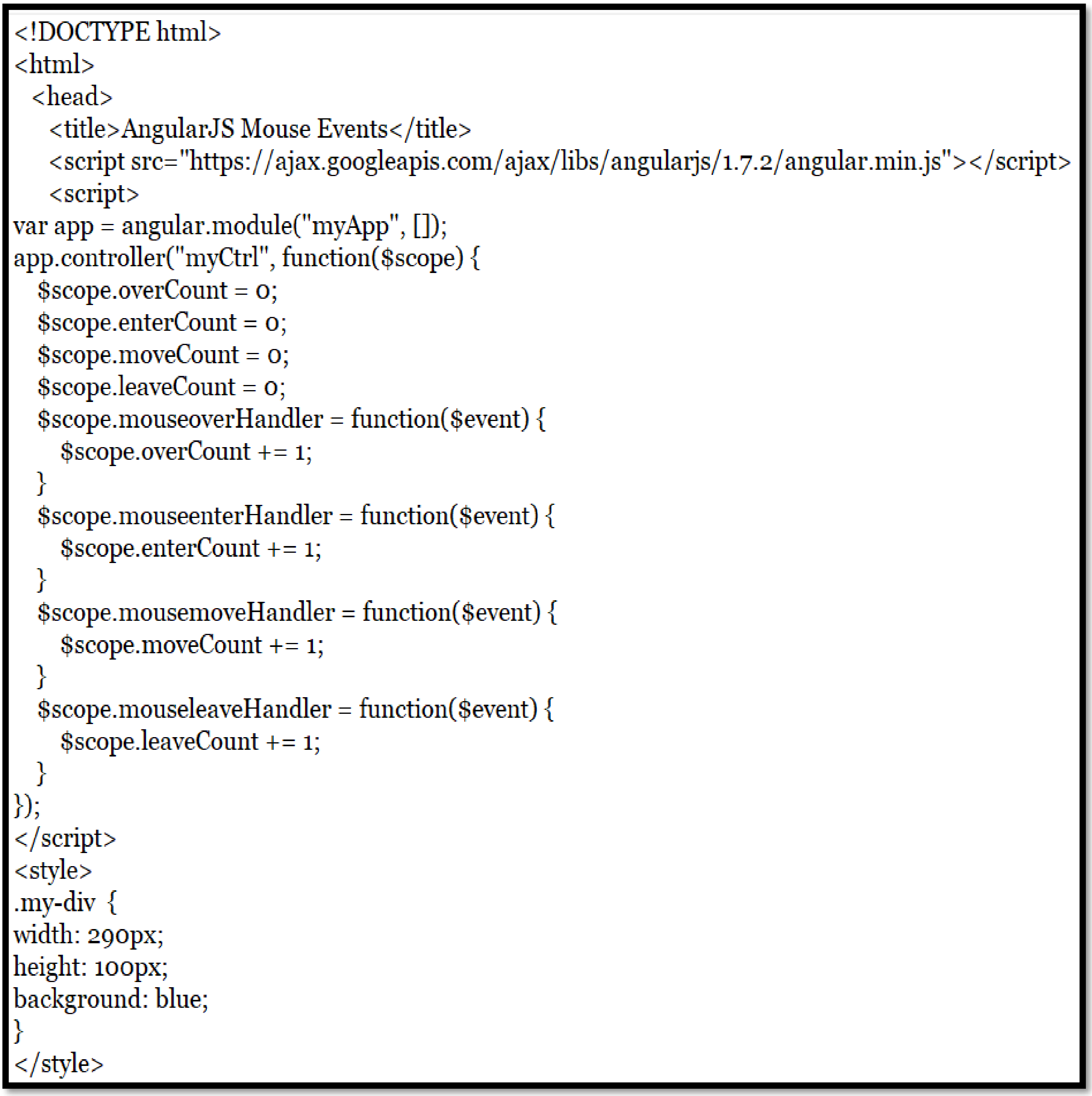
In angularjs ng-mouseup event directive is used to raise an events / functions whenever mouse clicked and released from html elements. The ng-mouseup event in angularjs will fire an events immediately whenever mouse click and released from elements. Suppose in angularjs if we want to raise an event / function whenever we did mouse click and released from html elements then it’s better to use ng-mouseup event.

Syntax of AngularJS ng-mouseup Event:

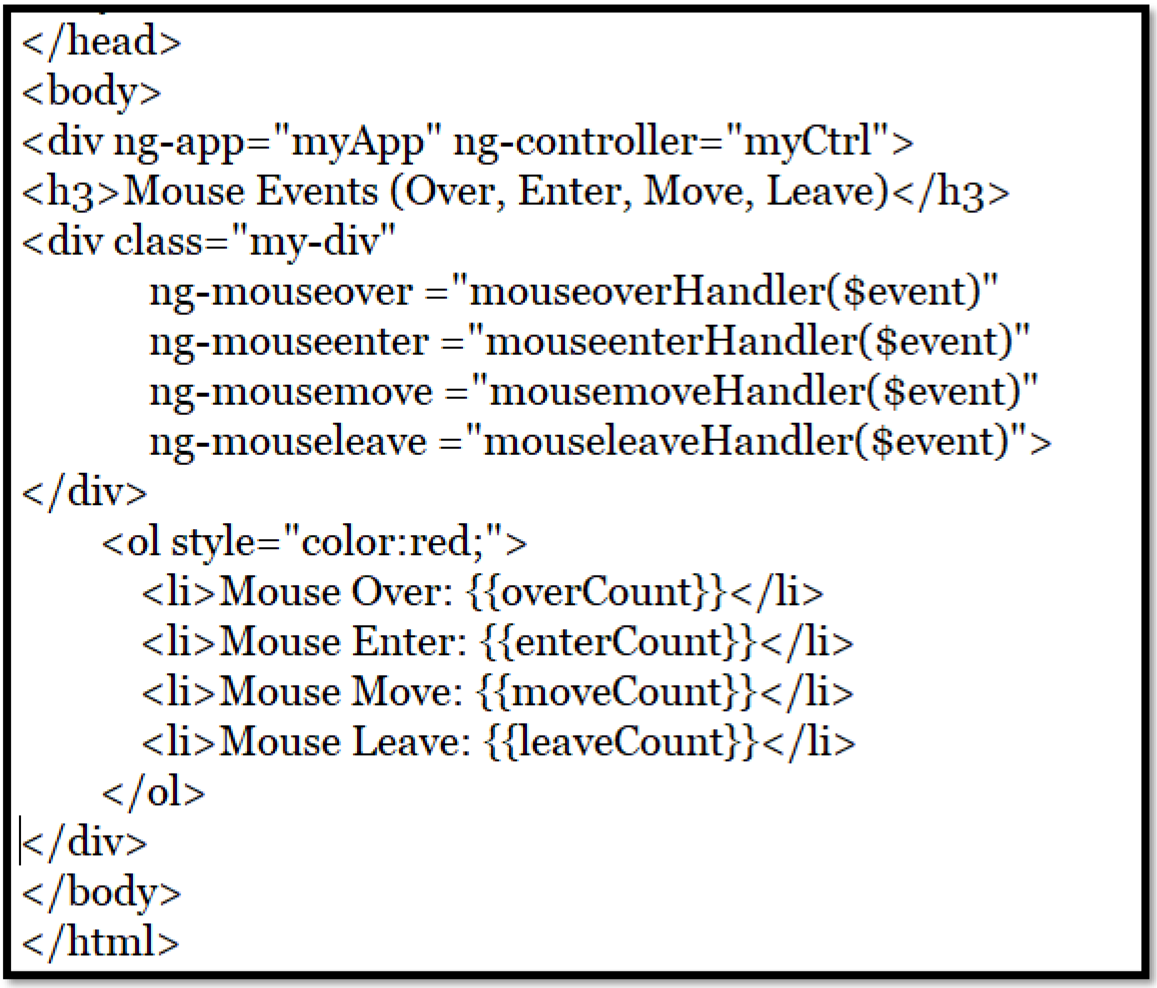
```
<element ng-mouseup="expression">
--Your Code--
</element>
```

In the above program we have used some of the AngularJS mouse events like mouseover, mouseenter, mouseleave and mousemove. As we move the cursor over the blue screen mouseover and mouseenter counts how many times the cursor was over and inside the blue screen. Mouseleave will be counting how many times the mouse left the blue screen. And mousemove will count how many times the mouse was moving on the screen. And the count is displayed. This is how the AngularJS events work.

CODE:

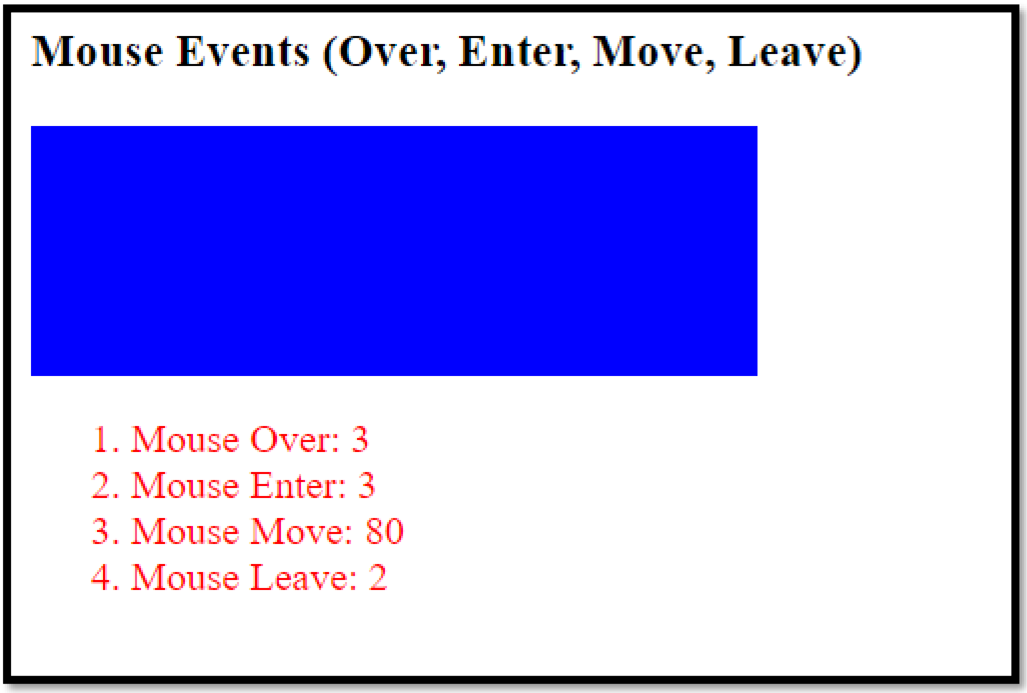


```
<!DOCTYPE html>
<html>
  <head>
    <title>AngularJS Mouse Events</title>
    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.7.2/angular.min.js"></script>
    <script>
var app = angular.module("myApp", []);
app.controller("myCtrl", function($scope) {
   $scope.overCount = 0;
   $scope.enterCount = 0;
   $scope.moveCount = 0;
   $scope.leaveCount = 0;
   $scope.mouseoverHandler = function($event) {
      $scope.overCount += 1;
   }
   $scope.mouseenterHandler = function($event) {
      $scope.enterCount += 1;
   }
   $scope.mousemoveHandler = function($event) {
      $scope.moveCount += 1;
   }
   $scope.mouseleaveHandler = function($event) {
      $scope.leaveCount += 1;
   }
});
</script>
<style>
.my-div {
width: 290px;
height: 100px;
background: blue;
}
</style>
```



```
</head>
<body>
<div ng-app="myApp" ng-controller="myCtrl">
<h3>Mouse Events (Over, Enter, Move, Leave)</h3>
<div class="my-div"
        ng-mouseover ="mouseoverHandler($event)"
        ng-mouseenter ="mouseenterHandler($event)"
        ng-mousemove ="mousemoveHandler($event)"
        ng-mouseleave ="mouseleaveHandler($event)">
</div>
    <ol style="color:red;">
      <li>Mouse Over: {{overCount}}</li>
      <li>Mouse Enter: {{enterCount}}</li>
      <li>Mouse Move: {{moveCount}}</li>
      <li>Mouse Leave: {{leaveCount}}</li>
    </ol>
</div>
</body>
</html>
```

OUTPUT:



Mouse Events (Over, Enter, Move, Leave)

1. Mouse Over: 3
2. Mouse Enter: 3
3. Mouse Move: 80
4. Mouse Leave: 2

PRACTICAL 5

AIM: Ionic 4-Create and build first project or application(Android and iOS)

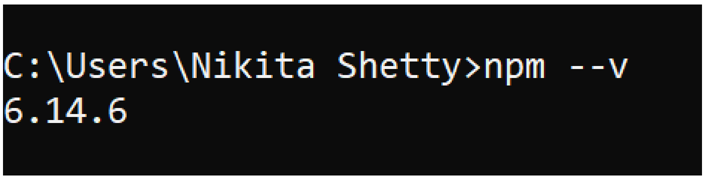
THEORY:

Introduction:

Ionic is open source framework used for developing mobile applications. It provides tools and services for building Mobile UI with native look and feel.

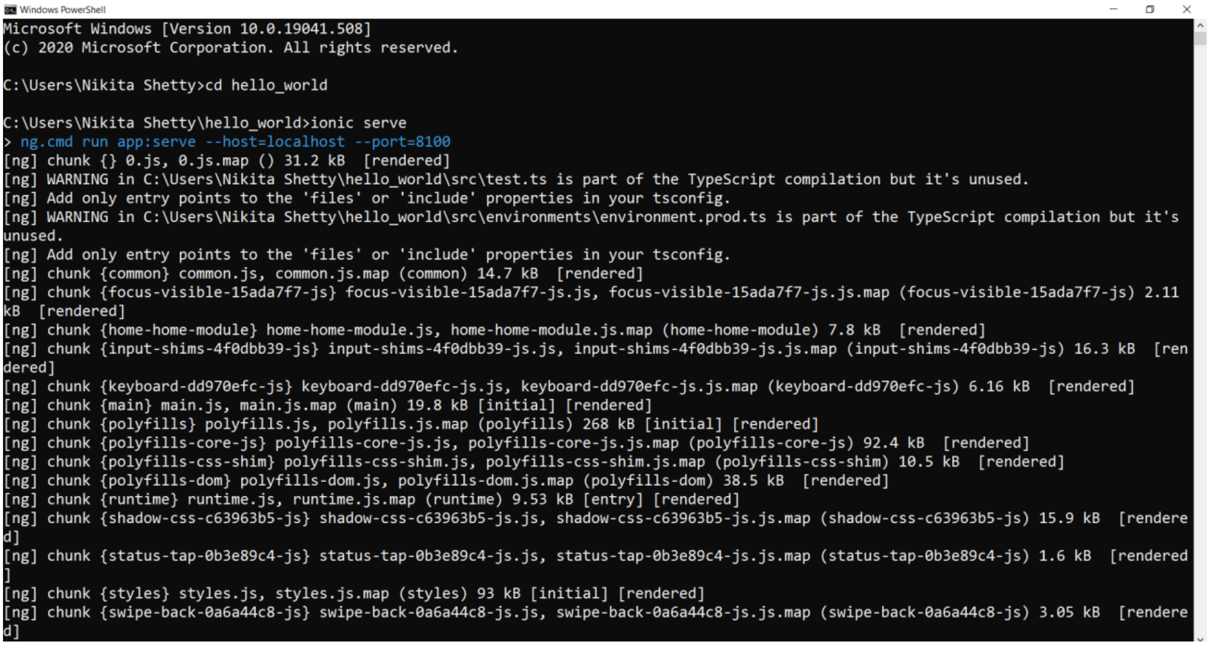
Steps to install Ionic and run the first Ionic Web Application:

1. Go to command prompt.
2. Check if npm is installed or not, using the following command:
   npm --v
   This will show the version of the npm.

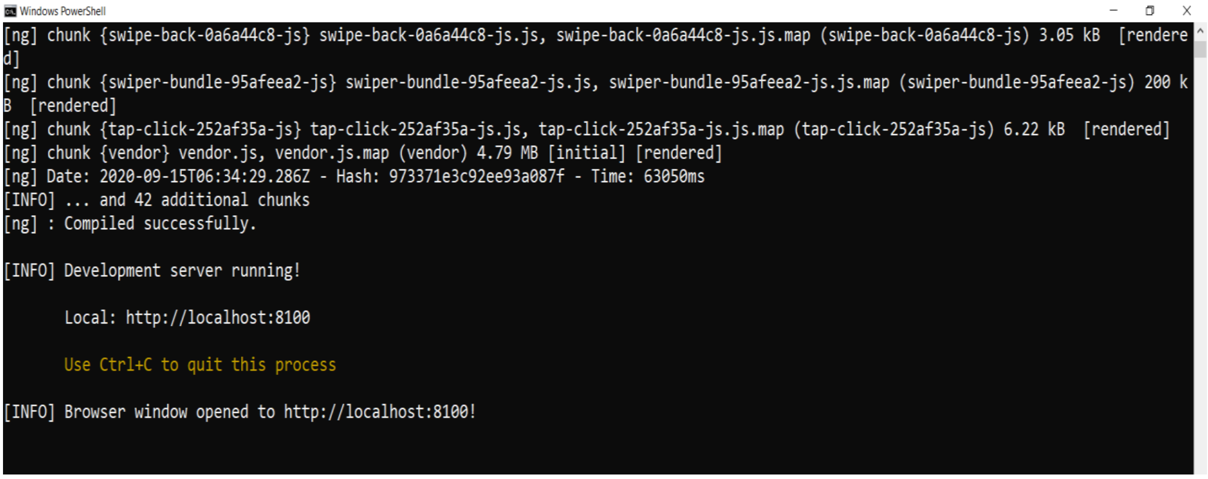


```
C:\Users\Nikita Shetty>npm --v
6.14.6
```

   If not then we have to install npm.
3. Then type the following commands to install ionic cordova and create a ionic project with the template:
   - **npm install -g ionic cordova**
   - **ionic start hello_world blank**
     (Here, we have chosen blank as the template)
     After this step hello_world folder will be downloaded .
4. Then in command prompt go to the hello_world folder using this command:
   cd hello_world
5. Then type: **ionic serve**
   This will give the following output:

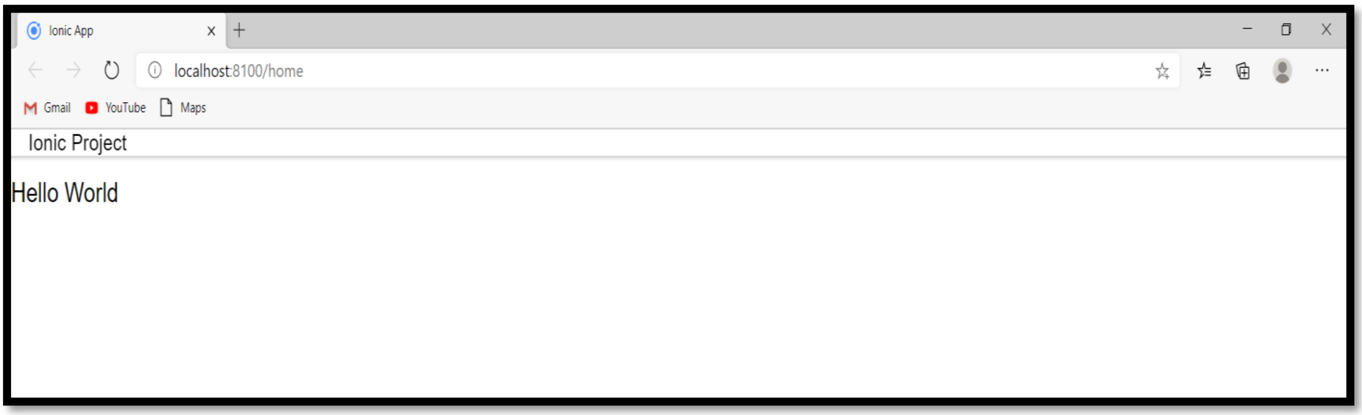


```
Microsoft Windows [Version 10.0.19041.508]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\Nikita Shetty>cd hello_world

C:\Users\Nikita Shetty\hello_world>ionic serve
> ng.cmd run app:serve --host=localhost --port=8100
[ng] chunk {} 0.js, 0.js.map () 31.2 kB  [rendered]
[ng] WARNING in C:\Users\Nikita Shetty\hello_world\src\test.ts is part of the TypeScript compilation but it's unused.
[ng] Add only entry points to the 'files' or 'include' properties in your tsconfig.
[ng] WARNING in C:\Users\Nikita Shetty\hello_world\src\environments\environment.prod.ts is part of the TypeScript compilation but it's
unused.
[ng] Add only entry points to the 'files' or 'include' properties in your tsconfig.
[ng] chunk {common} common.js, common.js.map (common) 14.7 kB  [rendered]
[ng] chunk {focus-visible-15ada7f7-js} focus-visible-15ada7f7-js.js, focus-visible-15ada7f7-js.js.map (focus-visible-15ada7f7-js) 2.11
kB  [rendered]
[ng] chunk {home-home-module} home-home-module.js, home-home-module.js.map (home-home-module) 7.8 kB  [rendered]
[ng] chunk {input-shims-4f0dbb39-js} input-shims-4f0dbb39-js.js, input-shims-4f0dbb39-js.js.map (input-shims-4f0dbb39-js) 16.3 kB  [ren
dered]
[ng] chunk {keyboard-dd970efc-js} keyboard-dd970efc-js.js, keyboard-dd970efc-js.js.map (keyboard-dd970efc-js) 6.16 kB  [rendered]
[ng] chunk {main} main.js, main.js.map (main) 19.8 kB [initial] [rendered]
[ng] chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 268 kB [initial] [rendered]
[ng] chunk {polyfills-core-js} polyfills-core-js.js, polyfills-core-js.js.map (polyfills-core-js) 92.4 kB  [rendered]
[ng] chunk {polyfills-css-shim} polyfills-css-shim.js, polyfills-css-shim.js.map (polyfills-css-shim) 10.5 kB  [rendered]
[ng] chunk {polyfills-dom} polyfills-dom.js, polyfills-dom.js.map (polyfills-dom) 38.5 kB  [rendered]
[ng] chunk {runtime} runtime.js, runtime.js.map (runtime) 9.53 kB [entry] [rendered]
[ng] chunk {shadow-css-c63963b5-js} shadow-css-c63963b5-js.js, shadow-css-c63963b5-js.js.map (shadow-css-c63963b5-js) 15.9 kB  [rendere
d]
[ng] chunk {status-tap-0b3e89c4-js} status-tap-0b3e89c4-js.js, status-tap-0b3e89c4-js.js.map (status-tap-0b3e89c4-js) 1.6 kB  [rendered
]
[ng] chunk {styles} styles.js, styles.js.map (styles) 93 kB [initial] [rendered]
[ng] chunk {swipe-back-0a6a44c8-js} swipe-back-0a6a44c8-js.js, swipe-back-0a6a44c8-js.js.map (swipe-back-0a6a44c8-js) 3.05 kB  [rendere
d]
```



```
[ng] chunk {swipe-back-0a6a44c8-js} swipe-back-0a6a44c8-js.js, swipe-back-0a6a44c8-js.js.map (swipe-back-0a6a44c8-js) 3.05 kB  [rendere
d]
[ng] chunk {swiper-bundle-95afeea2-js} swiper-bundle-95afeea2-js.js, swiper-bundle-95afeea2-js.js.map (swiper-bundle-95afeea2-js) 200 k
B  [rendered]
[ng] chunk {tap-click-252af35a-js} tap-click-252af35a-js.js, tap-click-252af35a-js.js.map (tap-click-252af35a-js) 6.22 kB  [rendered]
[ng] chunk {vendor} vendor.js, vendor.js.map (vendor) 4.79 MB [initial] [rendered]
[ng] Date: 2020-09-15T06:34:29.286Z - Hash: 973371e3c92ee93a087f - Time: 63050ms
[INFO] ... and 42 additional chunks
[ng] : Compiled successfully.

[INFO] Development server running!

       Local: http://localhost:8100

       Use Ctrl+C to quit this process

[INFO] Browser window opened to http://localhost:8100!
```

This will open the page in the browser and will display the following output:



Ionic Project

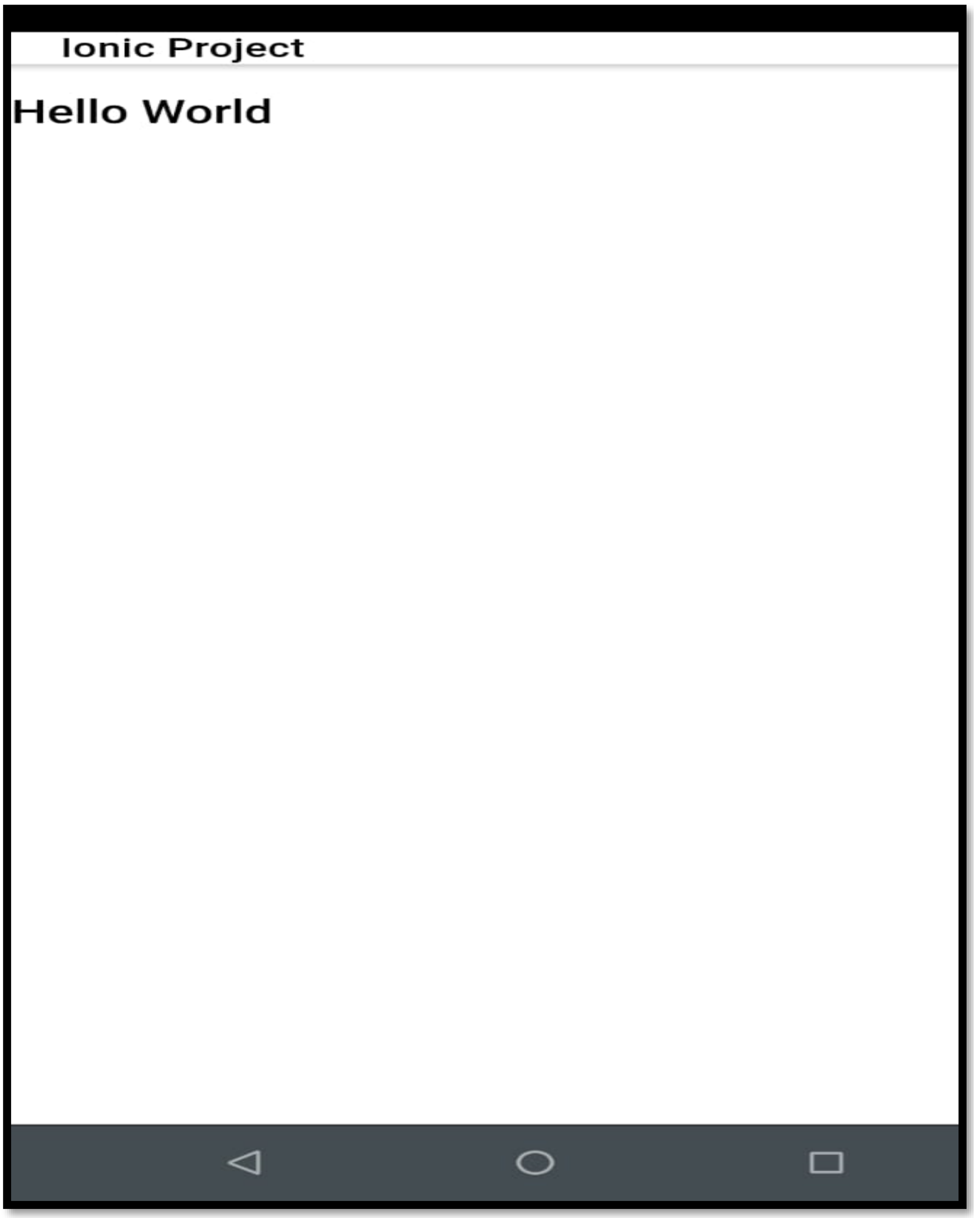
Hello World

This is how the ionic web application looks. Now we have to run this application on android device.

Steps to run the application on Android device:

1. If you want you can run the commands on Android Studio or else, you can use Command prompt.
2. Then type the following command:
   Ionic cordova platform add android
   This will add the android platform to the directory.
3. Then type:
   ionic cordova build android
   This will build the android apk in the directory.
4. Then lastly, type:
   ionic cordova run android
   This will run the Android application.
   If we have our phone connected to the machine, then we need to turn on the USB debugging under the Developer settings. This will export the app and will be viewable in your phone or you can run by setting up an emulator in the Android Studio.

This is the output of Application on Android Device.



Ionic Project

Hello World

CONCLUSION:

So, in this way we create a new ionic project or application and run the ionic application on an Android device.

PRACTICAL 6

AIM: Ionic - Adding Cordova Android Platform

THEORY:

Steps to create ionic project:

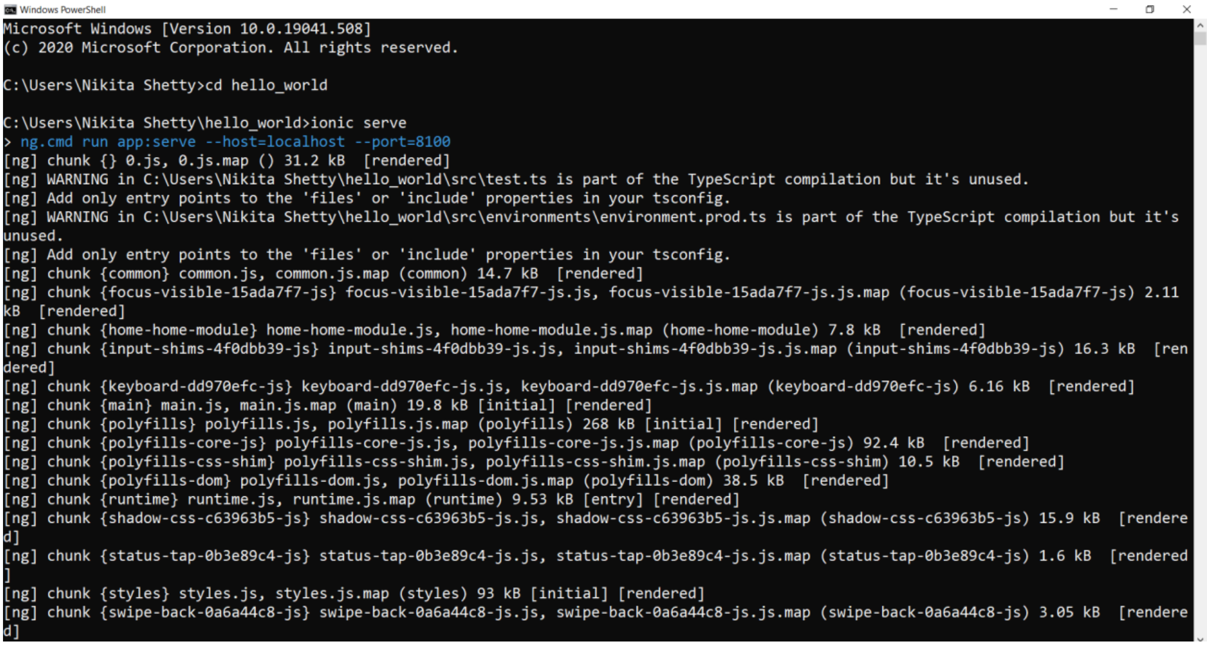
1. Go to command prompt.
2. If you have not installed ionic cordova then use the following command to install ionic cordova:

   npm install -g ionic cordova

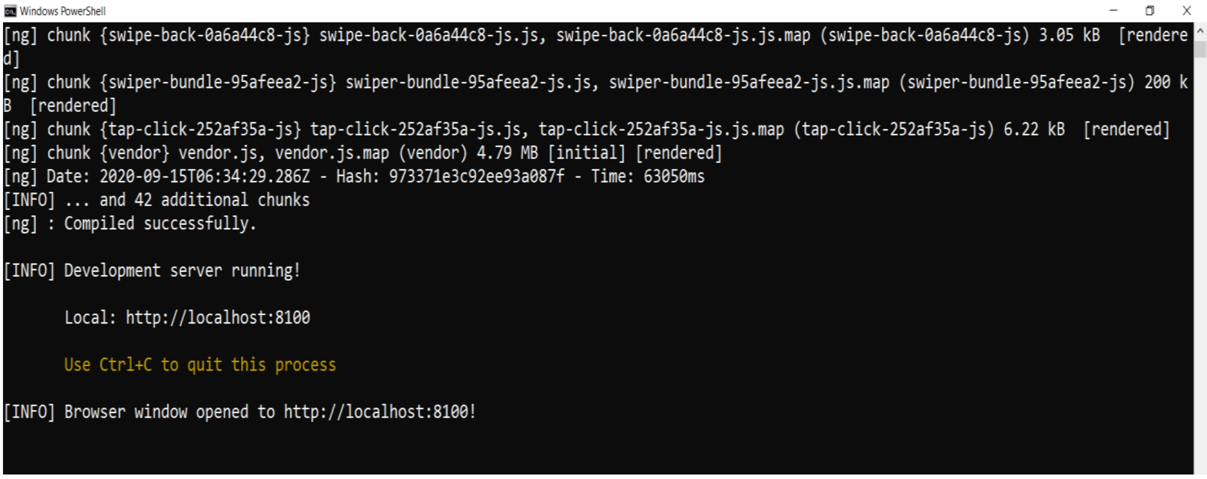
3. Then create a new ionic project with the template of your choice using the following command:

   ionic start hello_world blank

   (Here, we have chosen blank as the template)
   After this step hello_world folder will be downloaded .
4. Then in command prompt go to the hello_world folder using this command:
   cd hello_world
5. Then type: **ionic serve**
   This will give the following output:



```
Microsoft Windows [Version 10.0.19041.508]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\Nikita Shetty>cd hello_world

C:\Users\Nikita Shetty\hello_world>ionic serve
> ng.cmd run app:serve --host=localhost --port=8100
[ng] chunk {} 0.js, 0.js.map () 31.2 kB  [rendered]
[ng] WARNING in C:\Users\Nikita Shetty\hello_world\src\test.ts is part of the TypeScript compilation but it's unused.
[ng] Add only entry points to the 'files' or 'include' properties in your tsconfig.
[ng] WARNING in C:\Users\Nikita Shetty\hello_world\src\environments\environment.prod.ts is part of the TypeScript compilation but it's
unused.
[ng] Add only entry points to the 'files' or 'include' properties in your tsconfig.
[ng] chunk {common} common.js, common.js.map (common) 14.7 kB  [rendered]
[ng] chunk {focus-visible-15ada7f7-js} focus-visible-15ada7f7-js.js, focus-visible-15ada7f7-js.js.map (focus-visible-15ada7f7-js) 2.11
kB  [rendered]
[ng] chunk {home-home-module} home-home-module.js, home-home-module.js.map (home-home-module) 7.8 kB  [rendered]
[ng] chunk {input-shims-4f0dbb39-js} input-shims-4f0dbb39-js.js, input-shims-4f0dbb39-js.js.map (input-shims-4f0dbb39-js) 16.3 kB  [ren
dered]
[ng] chunk {keyboard-dd970efc-js} keyboard-dd970efc-js.js, keyboard-dd970efc-js.js.map (keyboard-dd970efc-js) 6.16 kB  [rendered]
[ng] chunk {main} main.js, main.js.map (main) 19.8 kB [initial] [rendered]
[ng] chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 268 kB [initial] [rendered]
[ng] chunk {polyfills-core-js} polyfills-core-js.js, polyfills-core-js.js.map (polyfills-core-js) 92.4 kB  [rendered]
[ng] chunk {polyfills-css-shim} polyfills-css-shim.js, polyfills-css-shim.js.map (polyfills-css-shim) 10.5 kB  [rendered]
[ng] chunk {polyfills-dom} polyfills-dom.js, polyfills-dom.js.map (polyfills-dom) 38.5 kB  [rendered]
[ng] chunk {runtime} runtime.js, runtime.js.map (runtime) 9.53 kB [entry] [rendered]
[ng] chunk {shadow-css-c63963b5-js} shadow-css-c63963b5-js.js, shadow-css-c63963b5-js.js.map (shadow-css-c63963b5-js) 15.9 kB  [rendere
d]
[ng] chunk {status-tap-0b3e89c4-js} status-tap-0b3e89c4-js.js, status-tap-0b3e89c4-js.js.map (status-tap-0b3e89c4-js) 1.6 kB  [rendered
]
[ng] chunk {styles} styles.js, styles.js.map (styles) 93 kB [initial] [rendered]
[ng] chunk {swipe-back-0a6a44c8-js} swipe-back-0a6a44c8-js.js, swipe-back-0a6a44c8-js.js.map (swipe-back-0a6a44c8-js) 3.05 kB  [rendere
d]
```



```
[ng] chunk {swipe-back-0a6a44c8-js} swipe-back-0a6a44c8-js.js, swipe-back-0a6a44c8-js.js.map (swipe-back-0a6a44c8-js) 3.05 kB  [rendere
d]
[ng] chunk {swiper-bundle-95afeea2-js} swiper-bundle-95afeea2-js.js, swiper-bundle-95afeea2-js.js.map (swiper-bundle-95afeea2-js) 200 k
B  [rendered]
[ng] chunk {tap-click-252af35a-js} tap-click-252af35a-js.js, tap-click-252af35a-js.js.map (tap-click-252af35a-js) 6.22 kB  [rendered]
[ng] chunk {vendor} vendor.js, vendor.js.map (vendor) 4.79 MB [initial] [rendered]
[ng] Date: 2020-09-15T06:34:29.286Z - Hash: 973371e3c92ee93a087f - Time: 63050ms
[INFO] ... and 42 additional chunks
[ng] : Compiled successfully.

[INFO] Development server running!

       Local: http://localhost:8100

       Use Ctrl+C to quit this process

[INFO] Browser window opened to http://localhost:8100!
```

Steps to add cordova android platform:

Once you are done creating the ionic project and want to run it on the Android devices we have to add android platform to our project.

We can do that using the following command:

ionic cordova platform add android

This will add the android platform to our project. So, now we can run the project or app on android devices using the following commands.

To create an android apk type:

ionic cordova build android

This will build the android apk in the directory.

Then lastly, type:

ionic cordova run android

This will run the application on Android device.

If we have our phone connected to the machine, then we need to turn on the USB debugging under the Developer settings. This will export the app and will be viewable in your phone or you can run by setting up an emulator in the Android Studio.

CONCLUSION:

So, in this way we can create a project and add Android platform to our project to run the application on Android devices.

PRACTICAL 7

AIM: Ionic - Create, Generate and Add Pages.

THEORY:

We can create an ionic project with a template of our choice and can add or create pages in the project.

We can create pages either manually or generate them using Ionic CLI.

So, I have created an ionic project named ionic_dikshita and now will be adding pages in this project.

Steps to generate pages and edit them:

1. To generate or add page we use following command:
   Ionic g page <PageName>
   (In place of the page name we will be typing the name of the page we want to add in our project).

We have added three pages in our project namely:

1. Home page
2. Items Details page
3. Items List page
4. About Us page

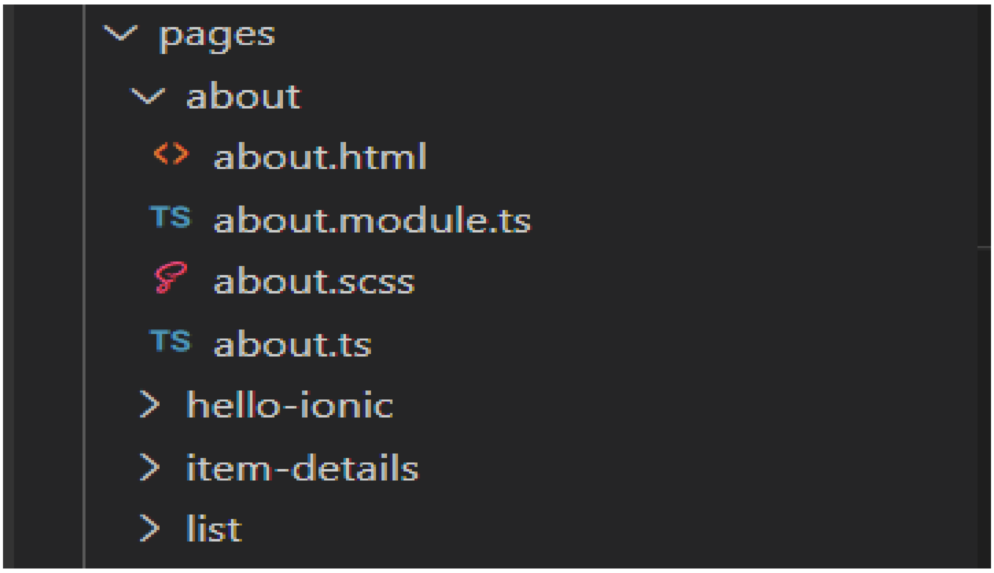
So, for these pages we will type the command as follows:

- **Ionic g page hello-ionic**
  (The main page)
- **Ionic g page item-details**
  (The Items Items details page)
- **Ionic g page list**
  (The Items list page)
- **Ionic g page about**
  (The about us page)

This command will create a folder with name **pages** which will include those three pages.
Each page will include four files:

- **html file**: which contains UI components for the page.
- **module.ts file**: which contains modules for specific page.
- **scss file**: which contains sass styles to style your page.
- **ts file**: which contains the TypeScript code.

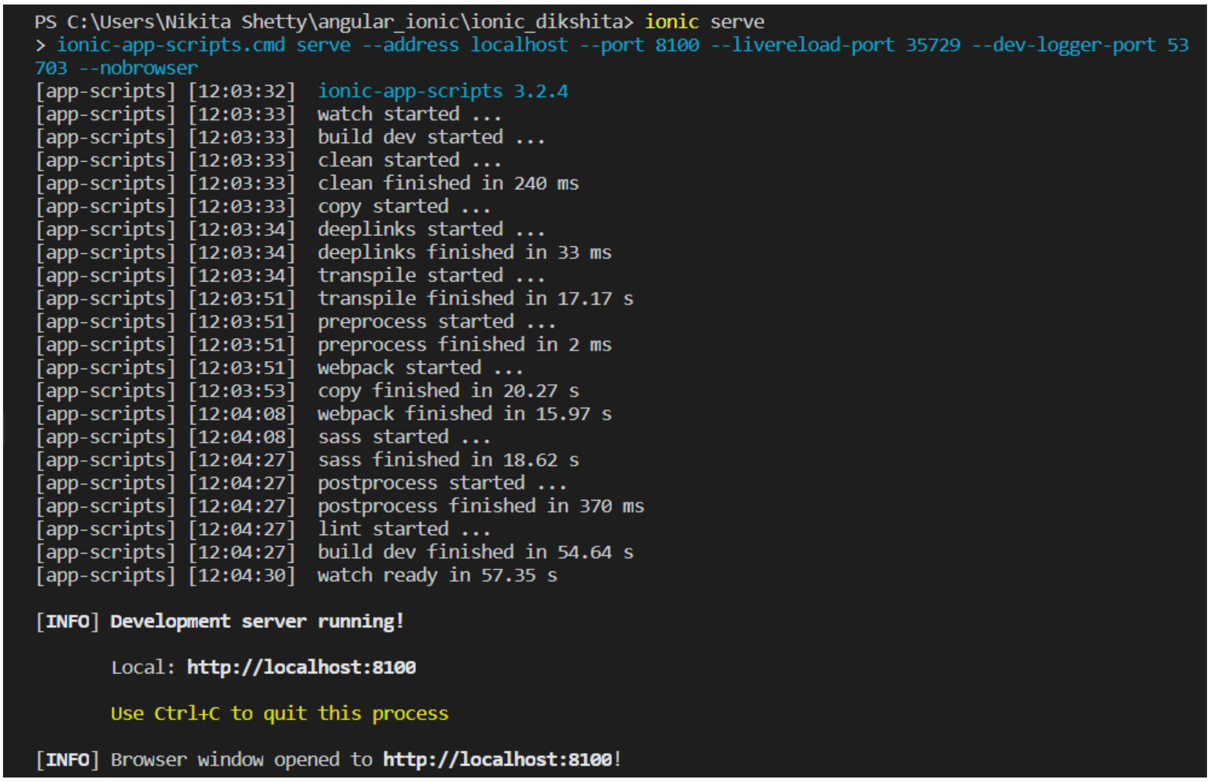
Like this:



```
pages
  about
    about.html
    about.module.ts
    about.scss
    about.ts
  hello-ionic
  item-details
  list
```

Then, we will design each pages and after the project is ready we will type command:

ionic serve

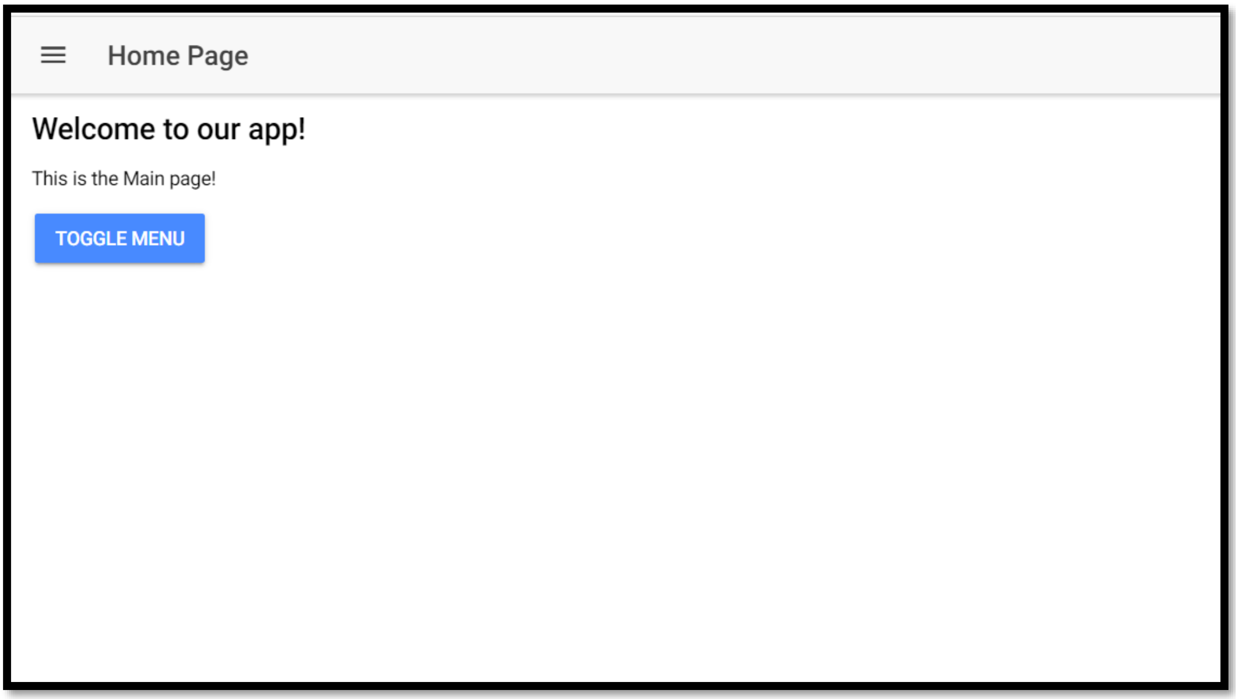


```
PS C:\Users\Nikita Shetty\angular_ionic\ionic_dikshita> ionic serve
> ionic-app-scripts.cmd serve --address localhost --port 8100 --livereload-port 35729 --dev-logger-port 53
703 --nobrowser
[app-scripts] [12:03:32]  ionic-app-scripts 3.2.4
[app-scripts] [12:03:33]  watch started ...
[app-scripts] [12:03:33]  build dev started ...
[app-scripts] [12:03:33]  clean started ...
[app-scripts] [12:03:33]  clean finished in 240 ms
[app-scripts] [12:03:33]  copy started ...
[app-scripts] [12:03:34]  deeplinks started ...
[app-scripts] [12:03:34]  deeplinks finished in 33 ms
[app-scripts] [12:03:34]  transpile started ...
[app-scripts] [12:03:51]  transpile finished in 17.17 s
[app-scripts] [12:03:51]  preprocess started ...
[app-scripts] [12:03:51]  preprocess finished in 2 ms
[app-scripts] [12:03:51]  webpack started ...
[app-scripts] [12:03:53]  copy finished in 20.27 s
[app-scripts] [12:04:08]  webpack finished in 15.97 s
[app-scripts] [12:04:08]  sass started ...
[app-scripts] [12:04:27]  sass finished in 18.62 s
[app-scripts] [12:04:27]  postprocess started ...
[app-scripts] [12:04:27]  postprocess finished in 370 ms
[app-scripts] [12:04:27]  lint started ...
[app-scripts] [12:04:27]  build dev finished in 54.64 s
[app-scripts] [12:04:30]  watch ready in 57.35 s

[INFO] Development server running!

       Local: http://localhost:8100

       Use Ctrl+C to quit this process

[INFO] Browser window opened to http://localhost:8100!
```

This will open the app in the browser.

The pages look as follows:

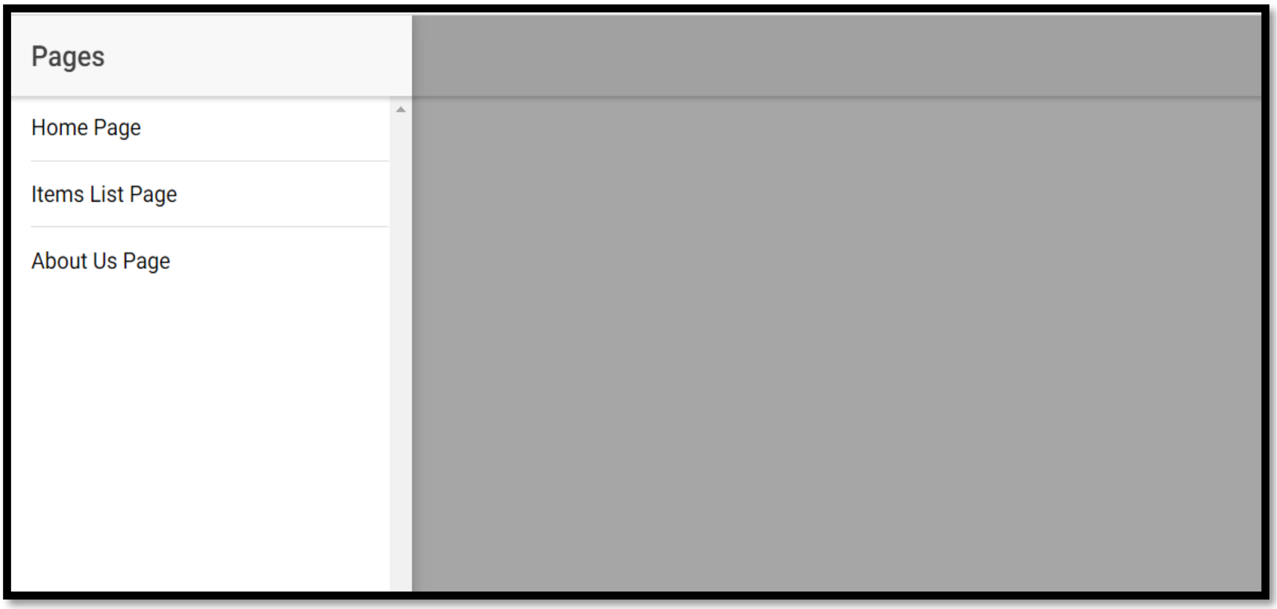
MAIN PAGE:



Welcome to our app!

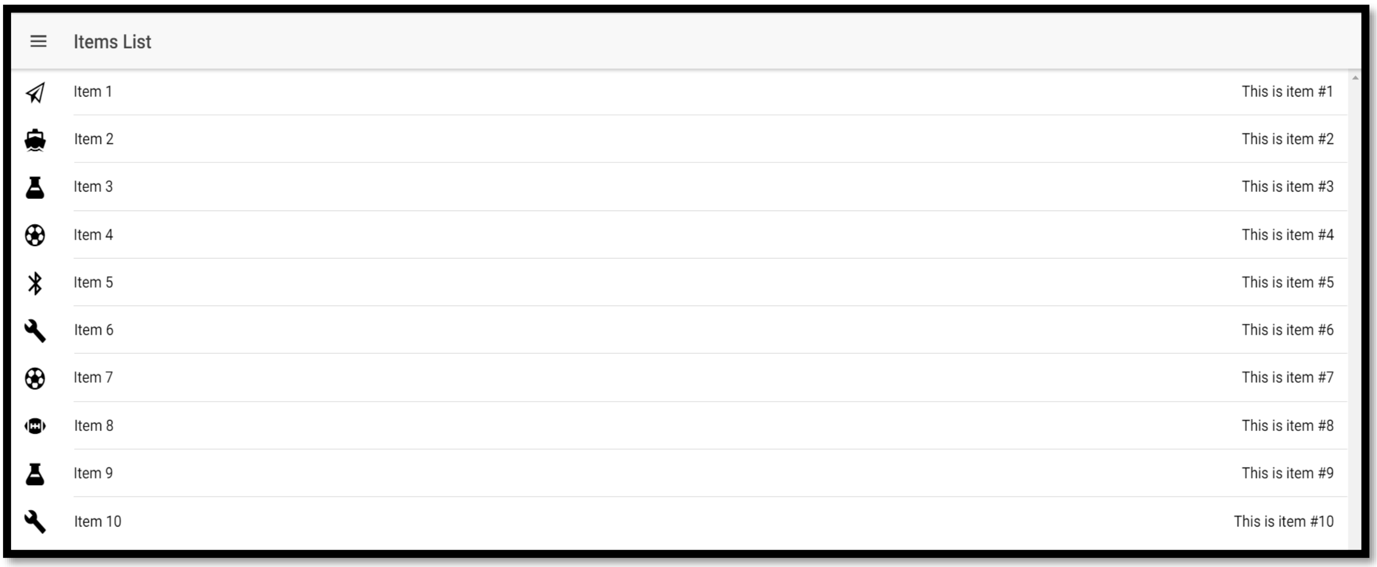
This is the Main page!

When we click on the toggle menu button it will open the toggle menu .



The toggle menu includes the following pages list. As we click on the pages for example if we click on about us page it will navigate to that About Us page.

ITEMS LIST PAGE:

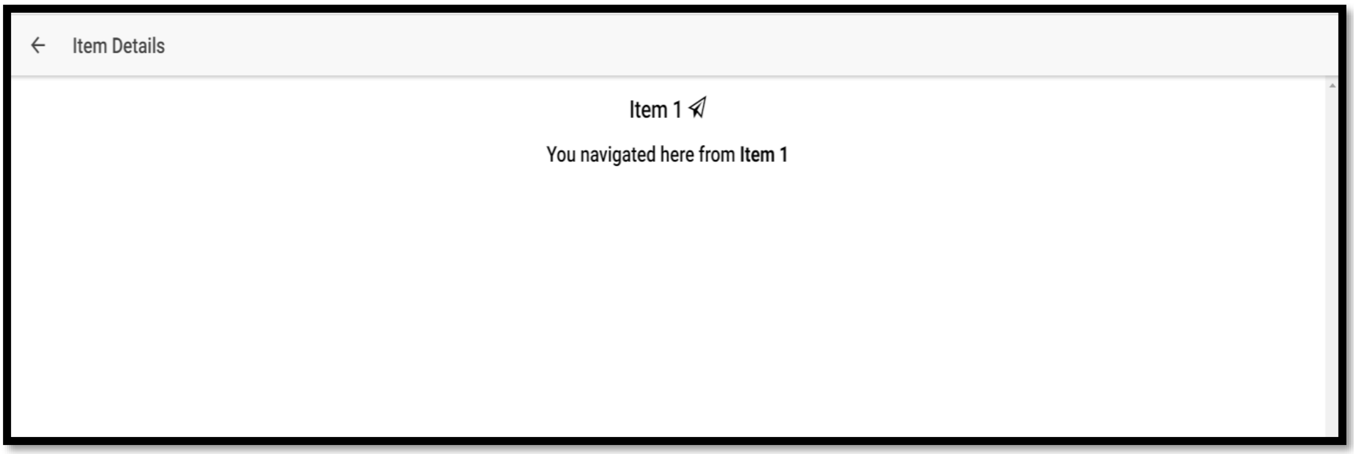


Items List

Item 1	This is item #1
Item 2	This is item #2
Item 3	This is item #3
Item 4	This is item #4
Item 5	This is item #5
Item 6	This is item #6
Item 7	This is item #7
Item 8	This is item #8
Item 9	This is item #9
Item 10	This is item #10

As we click on any of the items it will navigate to that Item’s detail page.

ITEMS DETAILS PAGE:

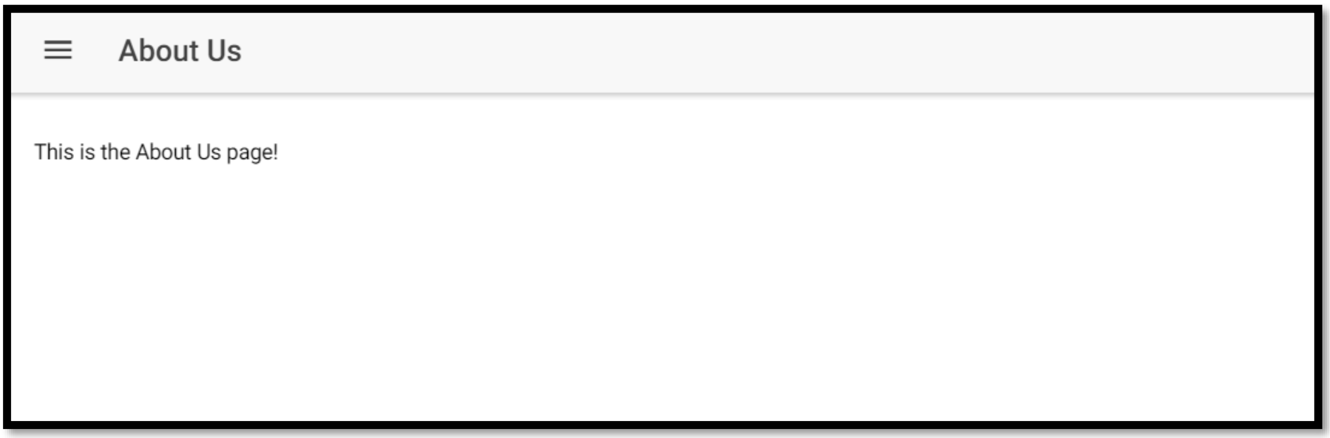


Item Details

Item 1

You navigated here from **Item 1**

ABOUT US PAGE:



About Us

This is the About Us page!

CONCLUSION:

So, in this way we generated or added pages in the project.

PRACTICAL 8

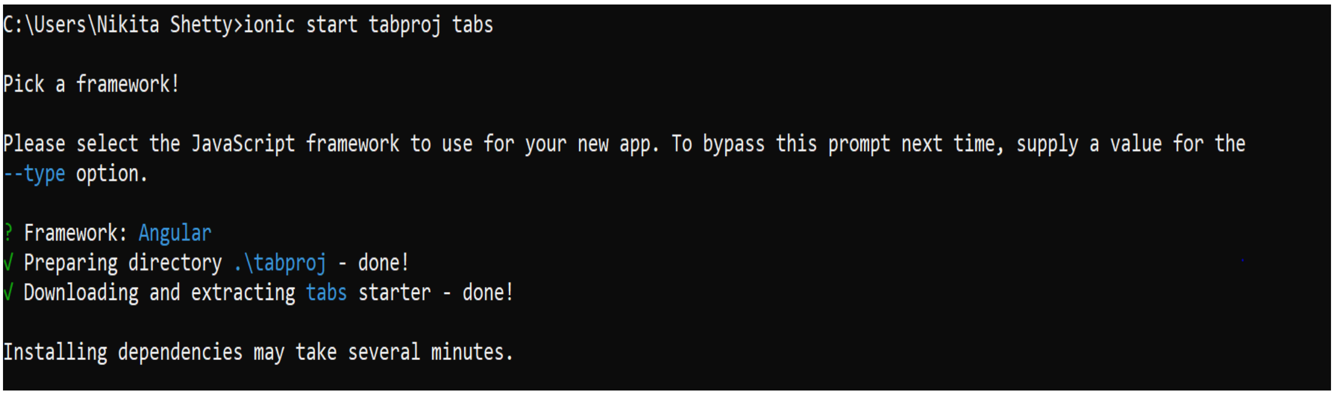
AIM: Ionic - Use Tabs Starter Template.

THEORY:

We will create an ionic project with **tabs template**.

For this we have to type the following command:

ionic start tabproj tabs



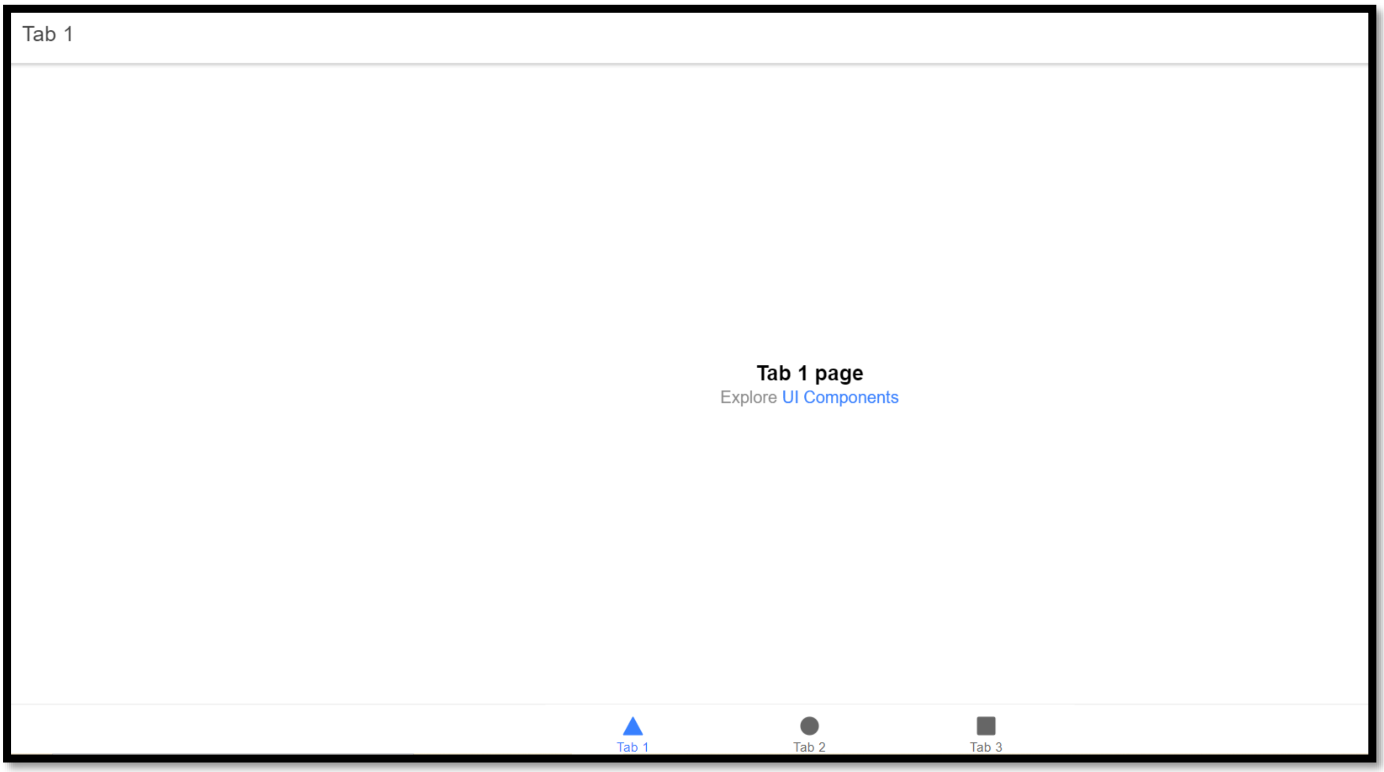
```
C:\Users\Nikita Shetty>ionic start tabproj tabs

Pick a framework!

Please select the JavaScript framework to use for your new app. To bypass this prompt next time, supply a value for the
--type option.

? Framework: Angular
√ Preparing directory .\tabproj - done!
√ Downloading and extracting tabs starter - done!

Installing dependencies may take several minutes.
```

This will create project named with tabproj with tabs template.

Then type the following command after navigating to the project folder:

Ionic serve

This will open the project in the browser.



Tab 1

Tab 1 page

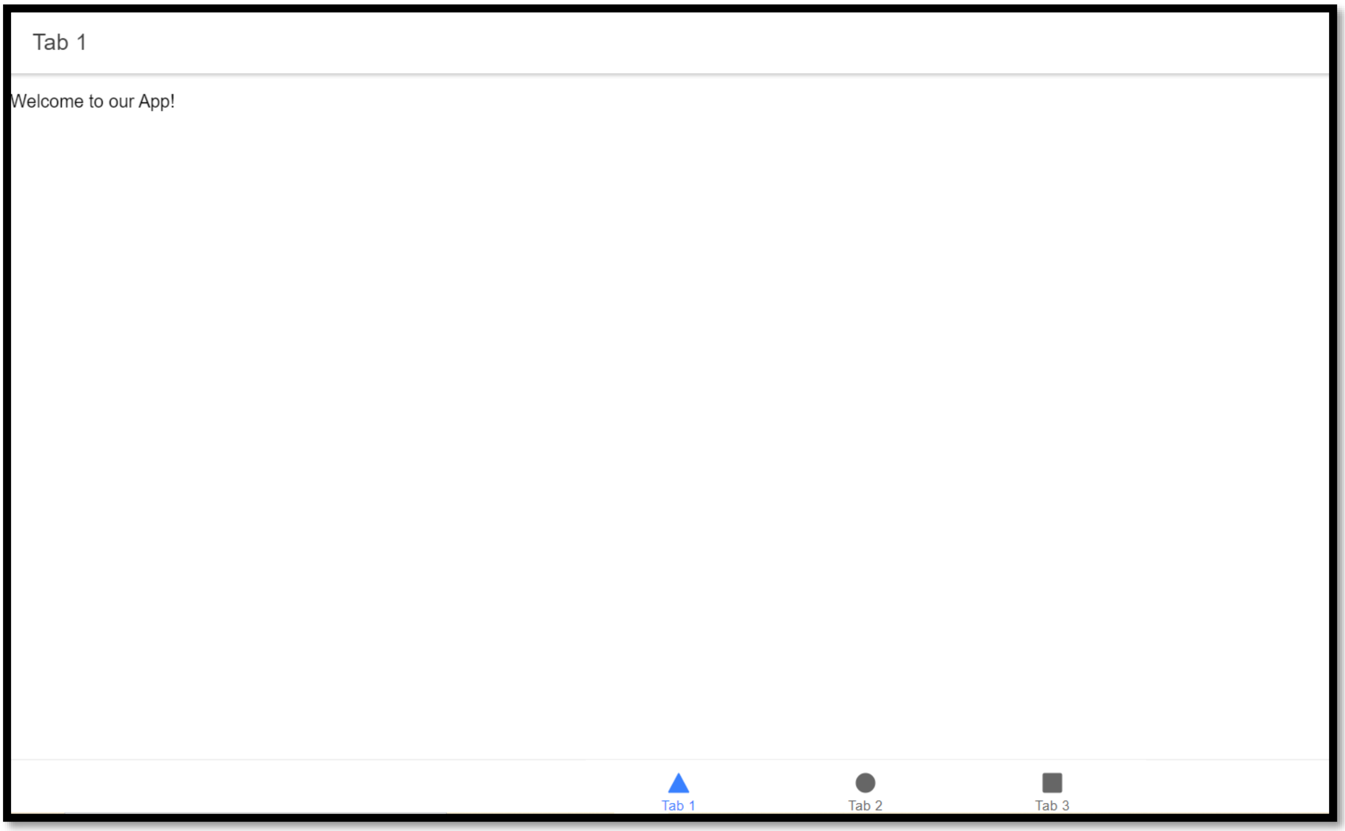
Explore UI Components

The project with tabs template looks like this. It has by default three tabs which is fixed in the footer of the page. Clicking on the tabs will navigate to the different pages.

We can now add any content on these tab pages.

Page on clicking TAB 1:

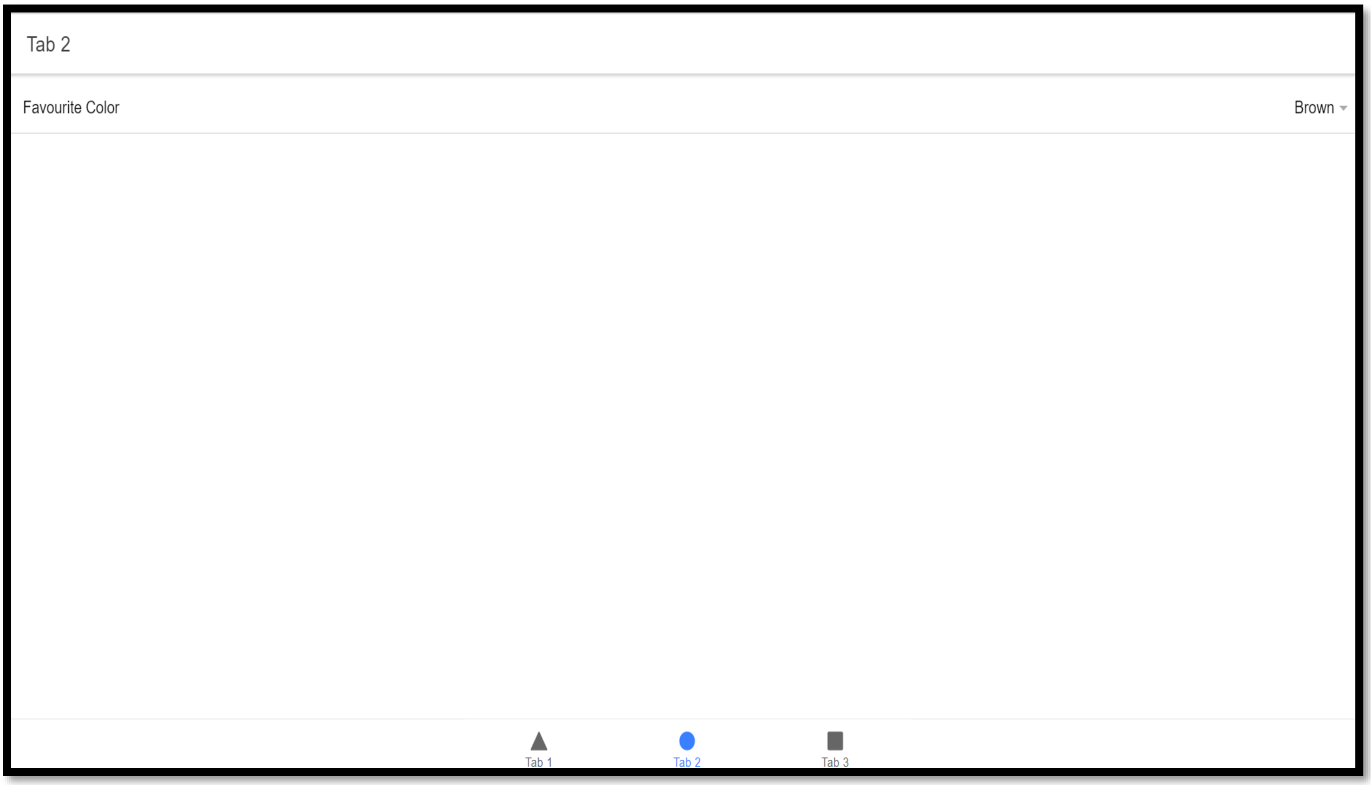
In this page we have just displayed a single statement.



Welcome to our App!

Page on clicking TAB 2:

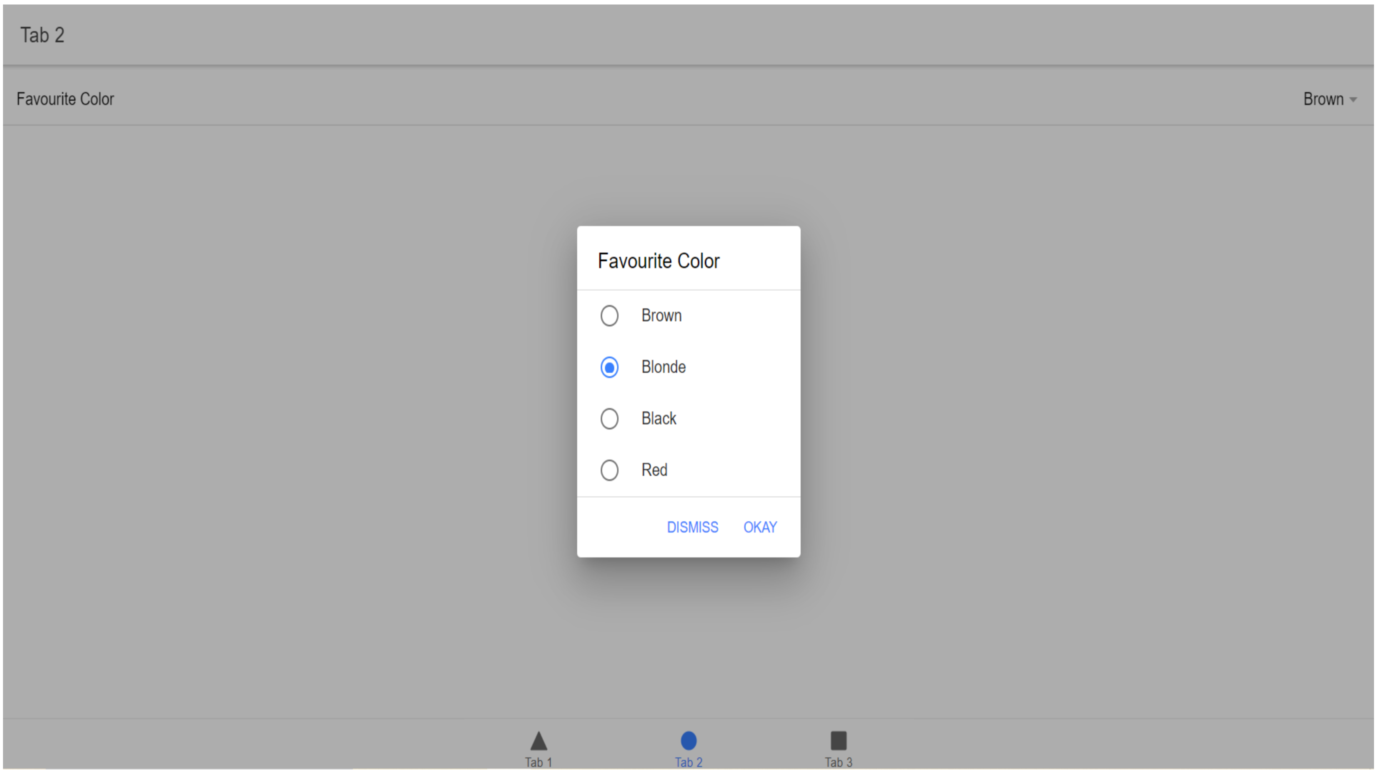
In this page we have used select component of ionic i.e. ion-select. It allows user to select an item from the select menu. Like as you can see we have given options for selecting favourite color from the list of options in this page.



Tab 2

Favourite Color Brown

On clicking the arrow icon beside the color brown it opens the menu which has many other options to select from.



Tab 2

Favourite Color Brown

Favourite Color

○ Brown

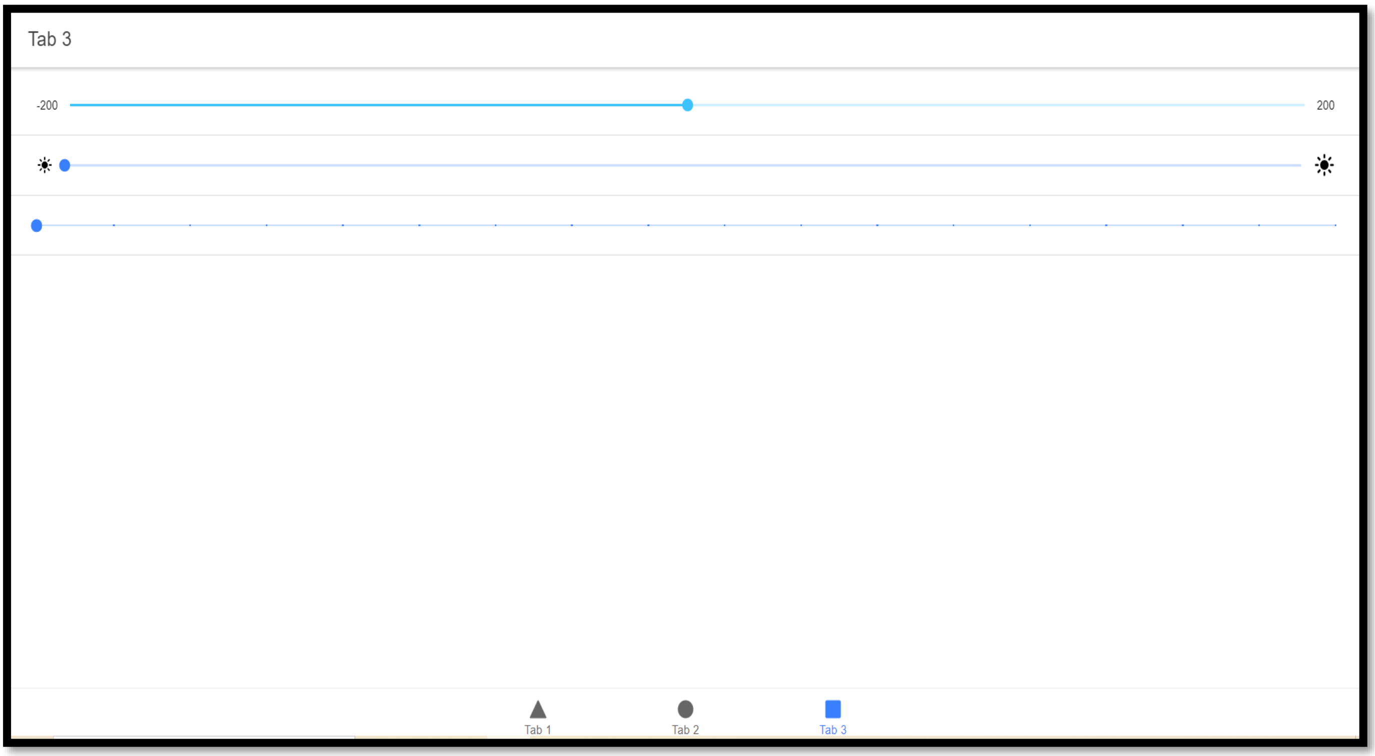
● Blonde

○ Black

○ Red

Page on clicking TAB 3:

In this page we have used range slider which lets user to select from the range of values by moving the slider knob.



CONCLUSION:

So, in this way we used the tabs starter template for creating an ionic project. Which comes with default tabs. Where these different tabs can navigate to different pages and we added different elements in those pages.