
**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: **NIDHEE DEEPAK PANCHAL.**

Roll No: **378** Programme: BSc IT 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 (Course Code: 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: 378**

## Subject: Hybrid Application Development

## INDEX

| Sr. No | Date | Topic | Sign |
|---|---|---|---|
| 1 | 27/07/2020 | AngularJS Data Binding | |
| 2 | 7/08/2020 | AngularJS Directives | |
| 3 | 14/08/2020 | AngularJS Controllers | |
| 4 | 25/08/2020 | AngularJS Events | |
| 5 | 08/09/2020 | Ionic Create and Build First Project | |
| 6 | 12/09/2020 | Ionic Adding Cordova Android Platform | |
| 7 | 19/09/2020 | Ionic Create, Generate and Add Pages | |
| 8 | 29/09/2020 | Ionic Use Tabs Starter Template | |

# PRACTICAL 1

## AngularJS Data Binding

Data binding in AngularJS is the synchronization between the model and the view.

The way that AngularJS implements 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. When the model changes the view reflects the change, and vice versa.

When data in the model changes, the view reflects the change, and when data in the view changes, the model is updated as well.

The HTML container where the AngularJS application is displayed called the view.

The view has access to the model, and there are several ways of displaying model data in the view. You can use the **ng-bind** directive, which will bind the innerHTML of the element to the specified model property.

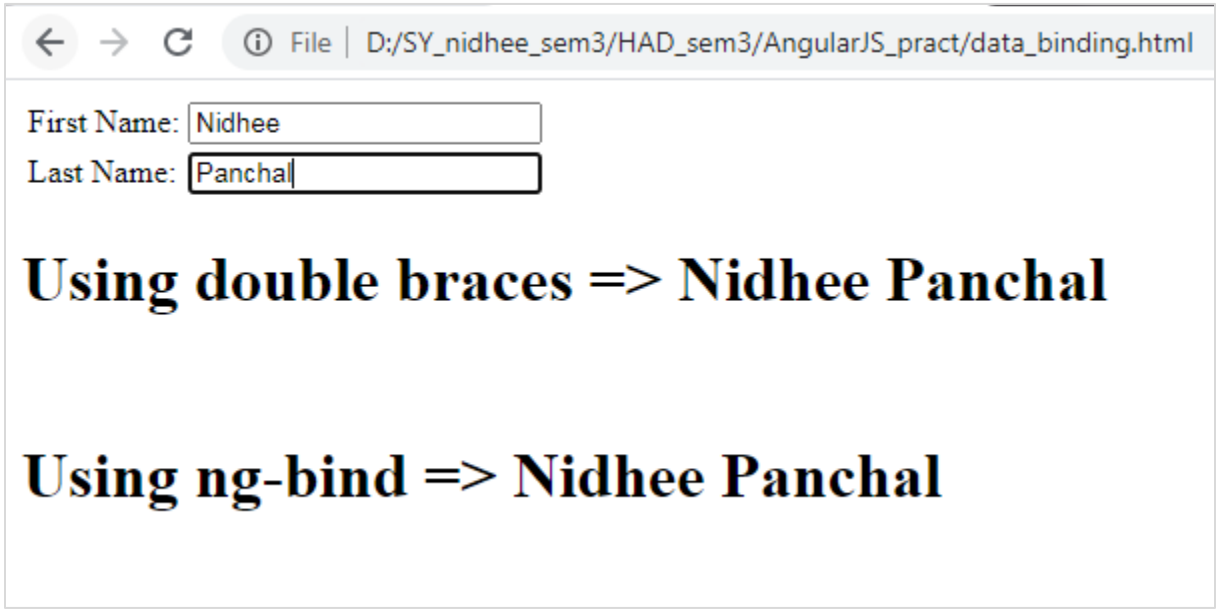
You can also use double braces **{{ }}** to display content from the model.

The **ng-model** directive provides a two-way binding between the model and the view.

data_binding.html > html > body > div > h1

```
<!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">
    <table>
        <tr>
            <td>First Name: </td>
            <td><input ng-model="firstname"></td>
        </tr>
        <tr>
            <td>Last Name: </td>
            <td><input ng-model="lastname"></td>
        </tr>
    </table>
    <h1>Using double braces => {{firstname +" "+ lastname}}</h1>
    <br>
    <h1>Using ng-bind => <b ng-bind="firstname"></b>  <b ng-bind="lastname"></b></h1>
</div>
<script>
var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope) {
    $scope.firstname = "";
    $scope.lastname = "";
});
</script>
</body>
</html>

```



# PRACTICAL 2

## AngularJS Directives

Directives are markers in the Document Object Model (DOM). Directives can be used with any of controller or HTML tag which will tell the compiler what exact operation or behavior is expected. There are some directives present which is predefined but if a developer wants he can create new directives (custom-directive).

**ng-app:**

The **ng-app Directive** in AngularJS is used to define the root element of an AngularJS application. This directive automatically initializes the AngularJS application on page load. It can be used to load various modules in AngularJS Application.

**ng-init:**

The ng-init directive is used to initialize an AngularJS Application data. It defines the initial value for an AngularJS application and assigns values to the variables. The ng-init directive defines initial values and variables for an AngularJS application.

**ng-model:**

ng-model is a directive which binds input, select and textarea, and stores the required user value in a variable and we can use that variable whenever we require that value.

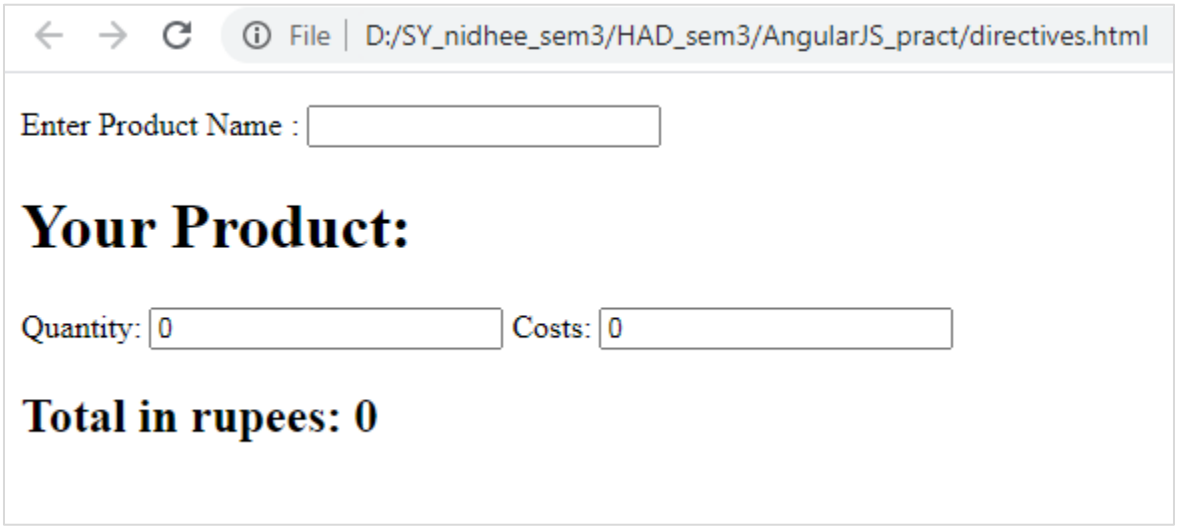
It also is used during validations in a form.

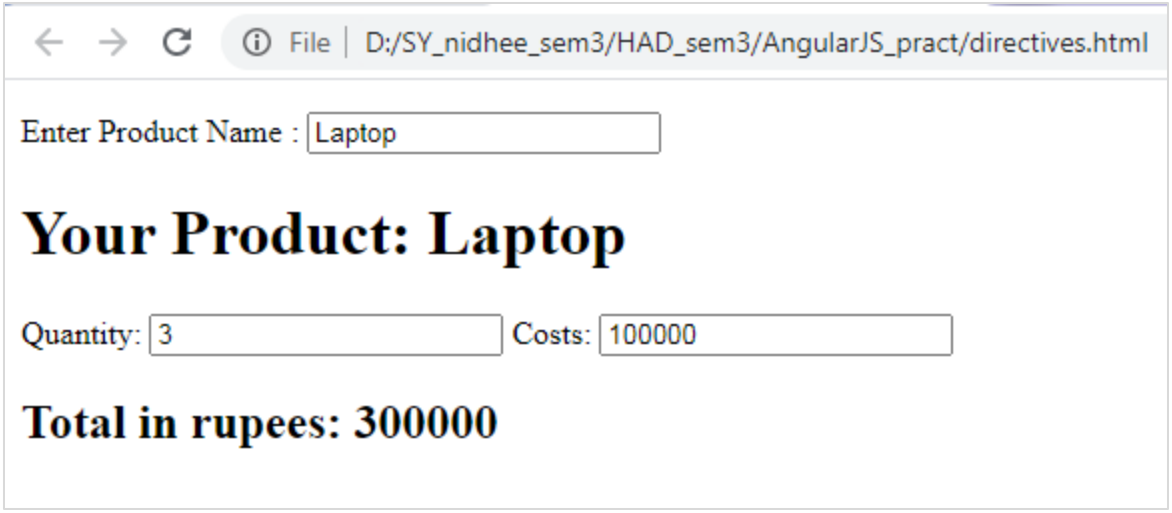
```html
<!DOCTYPE html>
<html lang="en-US">
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js
</script>
<body>

<div ng-app="" ng-init="quantity=0;price=0">
  <p>Enter Product Name : <input type="text" ng-model="name"></p>
  <h1>Your Product: {{name}}</h1>

Quantity: <input type="number" ng-model="quantity">
Costs:    <input type="number" ng-model="price">
<br>
<h2>Total in rupees: {{ quantity * price }}</h2>

</div>
</body>
</html>
```




File | D:/SY_nidhee_sem3/HAD_sem3/AngularJS_pract/directives.html
Enter Product Name : Laptop
Your Product: Laptop
Quantity: 3
Costs: 100000
Total in rupees: 300000

# PRACTICAL 3

## AngularJS Controllers

AngularJS controllers control the data of AngularJS applications.

AngularJS controllers are regular JavaScript Objects.

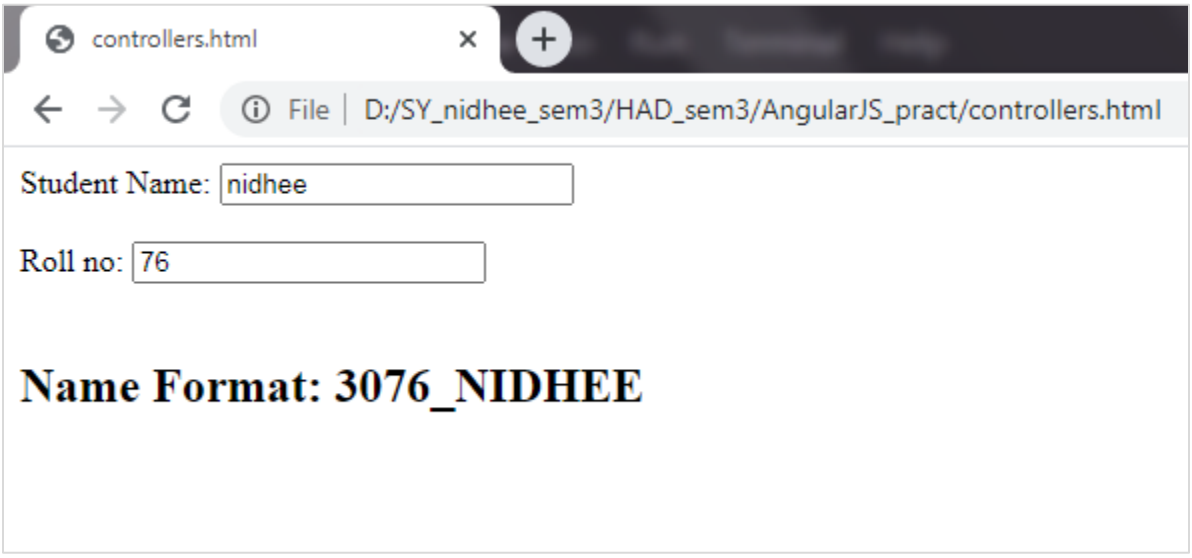
AngularJS controllers are used to control the flow of data of AngularJS application. A controller is defined using **ng-controller directive**. A controller is a JavaScript object containing attributes/properties and functions. Each controller accepts $scope as a parameter which refers to the application/module that controller is to control.

In AngularJS, if you want to do some manipulation in the view you must provide or add behavior in the scope object. AngularJS allow you to add behavior in the form of function or method in **$scope** object. You can then use these methods in your view by calling the method from controller.

In this program, we have named our controller as personCtrl and we have used method inside controller. We have accepted two values that is name and roll number of a student and the with the help of function returned the name in a particular format.

controllers.html > html > script

```
<!DOCTYPE html>
<html>
<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js">
</script>
<body>
<div ng-app="myApp" ng-controller="personCtrl">
Student Name: <input type="text" ng-model="studentName"><br><br>
Roll no: <input type="text" ng-model="rno"><br>
<br>
<h2>Name Format: {{ formattedName() | uppercase}}  </h2>
</div>
<!--Using methods inside Controller-->
<script>
var app = angular.module('myApp', []);
app.controller('personCtrl', function($scope) {
    $scope.studentName = "nidhee";
    $scope.rno = "76";
    $scope.formattedName = function() {
        return "30" + $scope.rno + "_" + $scope.studentName ;
    };
});
</script>
</body>
</html>
```



# PRACTICAL 4

## AngularJS Events:

When creating more advanced AngularJS applications, sooner or later your application will need to handle DOM events like mouse clicks, moves, keyboard presses, change events etc. AngularJS has a simple model for how to add event listeners to the HTML you generate from your views.

Events in AngularJS can be added using the Directives mentioned below:

**ng-mousemove:** Movement of mouse leads to the execution of event.

**ng-mouseup:** Movement of mouse upwards leads to the execution of event.

**ng-mousedown:** Movement of mouse downwards leads to the execution of event.

**ng-mouseenter:** Click of the mouse button leads to the execution of event.

**ng-mouseover:** Hovering of the mouse leads to the execution of event.

**ng-cut:** Cut operation leads to the execution of the event.

**ng-copy:** Copy operation leads to the execution of the event.

**ng-keypress:** Press of key leads to the execution of the event.

**ng-keyup:** Press of upward arrow key leads to the execution of the event.

**ng-keydown:** Press of downward arrow key leads to the execution of the event.

**ng-dblclick:** Double click leads to the execution of the event.

And many more are there.

In this code, we have used 2 events:

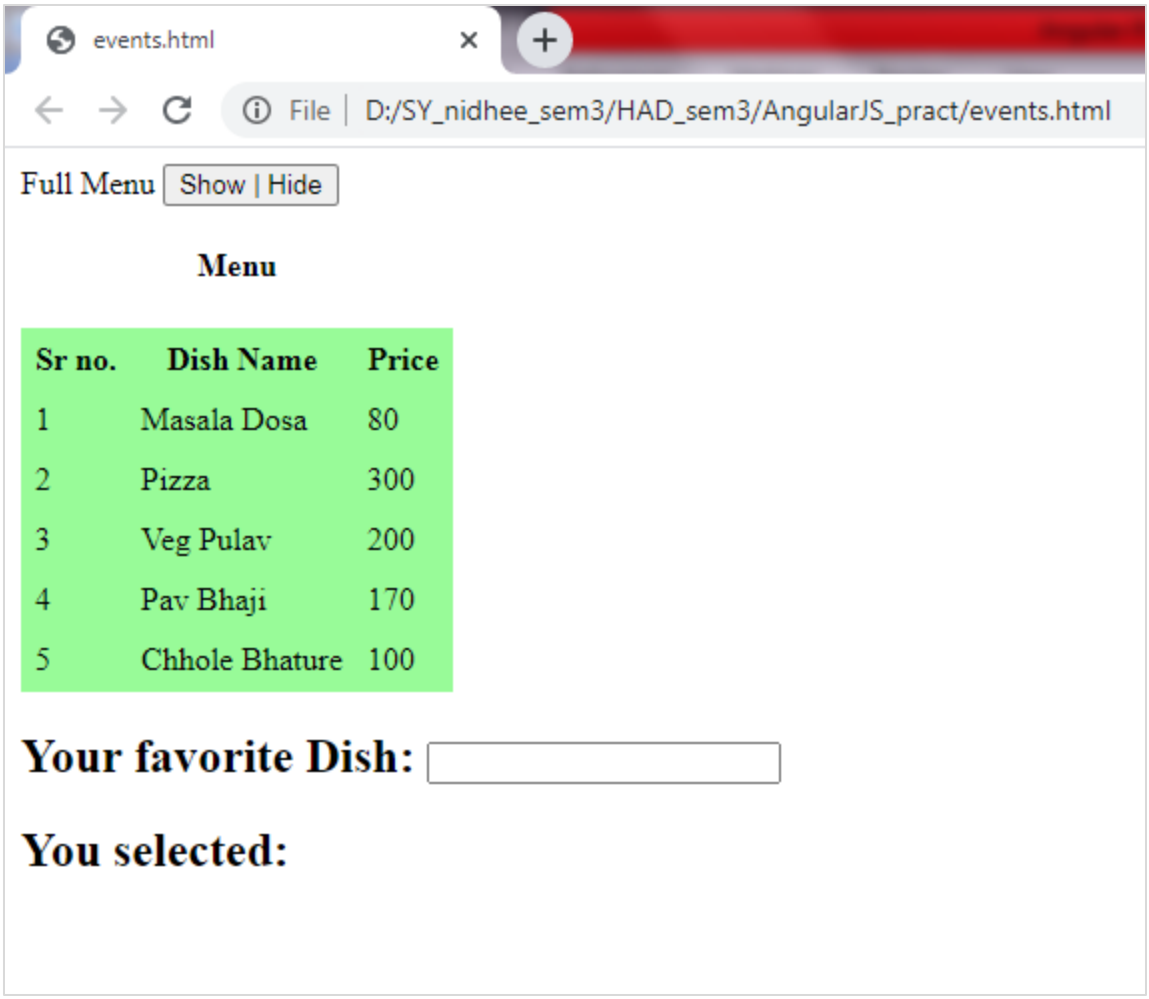
**ng-click:** Single click leads to the execution of the event.

**ng-blur:** When user clicks anywhere out of the textbox and the textbox is not focused then this event executes.

events.html > html > body > div > table > tr

```
<!DOCTYPE html>
<html lang="en-US">
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script
<body>
<div ng-app="eventsApp" ng-controller="myCtrl">
  Full Menu <button ng-click="myfunc()"> Show | Hide </button>
  <table ng-show="showMe" cellpadding=5px bgcolor="palegreen">
    <caption><h4> Menu </h4></caption>
    <tr>
        <th>Sr no.</th>
        <th>Dish Name</th>
        <th>Price</th>
    </tr>
    <tr ng-repeat = "dish in menu_item">
        <td>{{ dish.num }}</td>
        <td>{{ dish.name }}</td>
        <td>{{ dish.price }}</td>
    </tr>
  </table>
  <h2>Your favorite Dish:
      <input type="text" ng-model="name"
      ng-blur="result = 'ng-blur event. You came out of the textbox!! '">
  </h2>
  <h2>You selected: {{ name | uppercase }}</h2>
  <span style="color: red;">{{result}}</span>
</div>
```

```
  <script>
      var app = angular.module('eventsApp', []);
      app.controller('myCtrl', function($scope) {
          $scope.name = "";
          $scope.menu_item = [
              {num:1, name:'Masala Dosa', price:80},
              {num:2, name:'Pizza', price:300},
              {num:3, name:'Veg Pulav', price:200},
              {num:4, name:'Pav Bhaji', price:170},
              {num:5, name:'Chhole Bhature', price:100},
          ]
          $scope.myfunc = function() {
              $scope.showMe = !$scope.showMe;
              }
          });
      </script>
  </body>
  </html>
```


events.html
File | D:/SY_nidhee_sem3/HAD_sem3/AngularJS_pract/events.html
Full Menu Show | Hide
Menu
Sr no. Dish Name Price
1 Masala Dosa 80
2 Pizza 300
3 Veg Pulav 200
4 Pav Bhaji 170
5 Chhole Bhature 100
Your favorite Dish:
You selected:


events.html
File | D:/SY_nidhee_sem3/HAD_sem3/AngularJS_pract/events.html
Full Menu Show | Hide
Your favorite Dish: Veg pulav
You selected: VEG PULAV
ng-blur event. You came out of the textbox!!

# PRACTICAL 5

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

### Introduction:

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

Ionic and Cordova

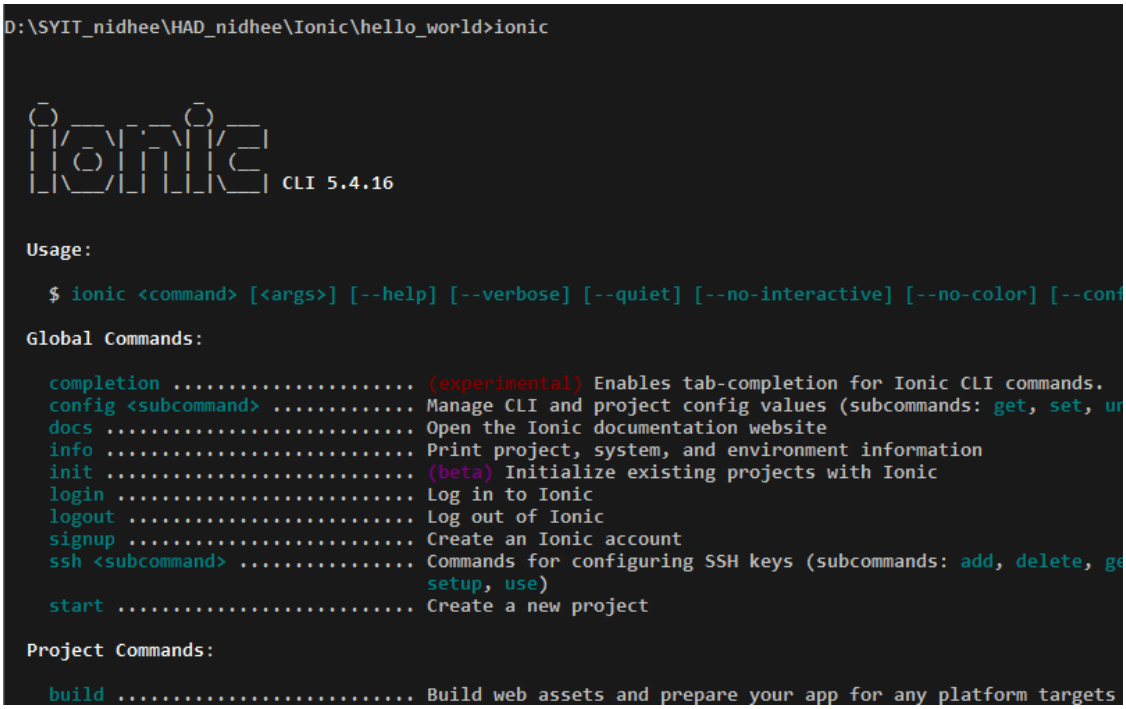
While Cordova provides the solutions to use native mobile functionality and to create fully native applications, it doesn't include a UI SDK. With over 100 UI components, plus navigation and platform-specific styling, Ionic allows you to develop high performing, native-like apps.

### Steps to install Ionic Cordova:

*Note:* npm i.e. Node Package Manager should be installed.

To check whether it is already installed type `node – v` and `npm – v`.If not installed then install npm.

- Go to command prompt
- Type the following commands to install Ionic Cordova (-g means globally),
  `npm install – g ionic cordova`
- To check if it is installed type,
  `ionic`

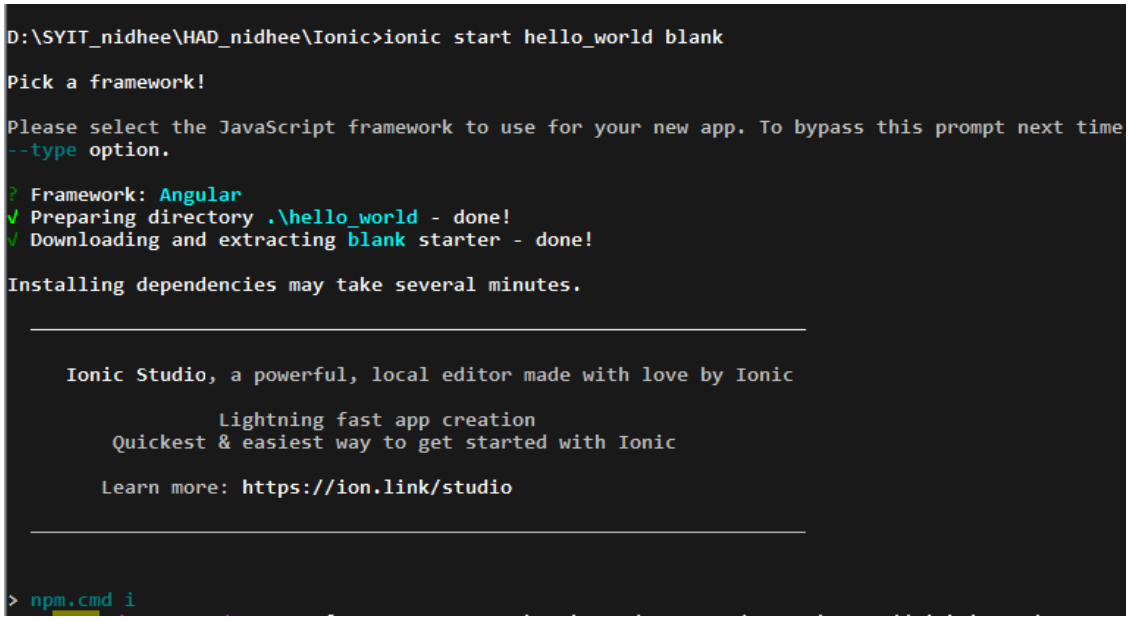



**Start/build a new project:**

Type the following commands in command prompt,

- ionic start hello_world blank

  Here, project name is **hello_world** and template is blank. At this step you have to choose a one of the Framework from Angular and React. Choose Angular and press enter.



- Now the project named as hello_world will be build. Change the directory to hello_world,

  `cd hello_world`

- To start the server type following command,

  `ionic serve`

  Remember: Copy the local host address and paste it in the browser.

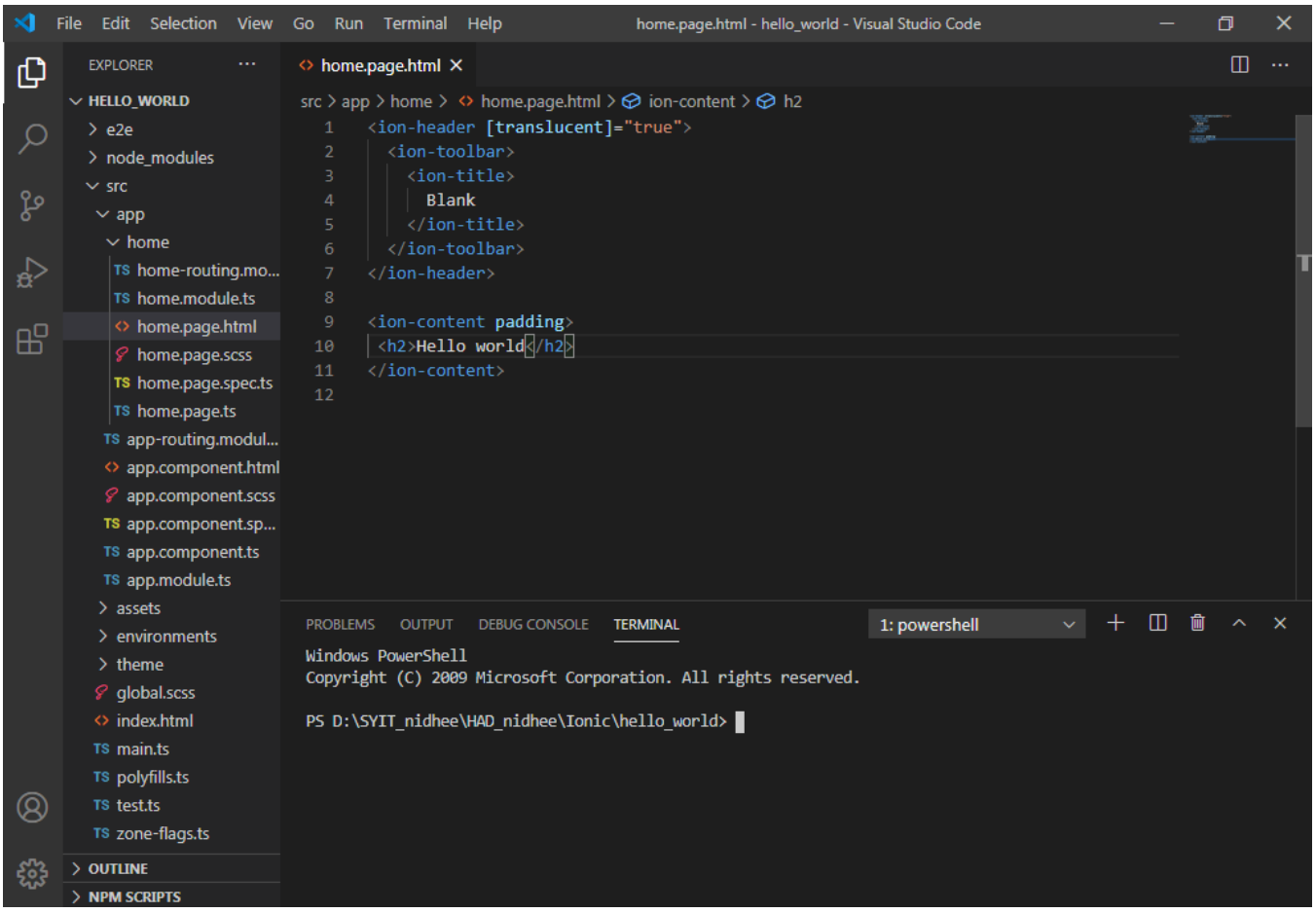
  To terminate/exit, press `ctrl+c`

Now, open the text editor and open the project hello_world.

Then go to

/src/app/home/home.page.html

Write the code as shown below:



After changes,

Again run the command ionic serve, you can now see the changes in the browser and you can now see the changes which we have made in our project.

**Build android:**

Pre-requisites:

- Java jdk installation and JAVA_HOME
- ANDROID_HOME
- Android SDK build Tools
- Android SDK platforms Tools

Type the following commands in the prompt to build android apk,

```
ionic cordova platform add android
ionic cordova build android
```

# PRACTCAL 6

## Ionic Adding Cordova Android Platform

After creating a project we need to test it on a device to ensure the performance of our application.

**Deploying to a Device:**

Testing your app in the browser with **ionic serve** or with an emulator is fast, easy and convenient when your app is in development, but eventually you're going to have to test on a device. Not only is it the only way to accurately test how your app will behave and perform, many Ionic Native plugins will only work when they are run on actual hardware.

Pre-requisites:

- Java jdk installation (version  and JAVA_HOME
- ANDROID_HOME
- Android SDK build Tools
- Android SDK platforms Tools

Type the following commands in the prompt to build android apk,

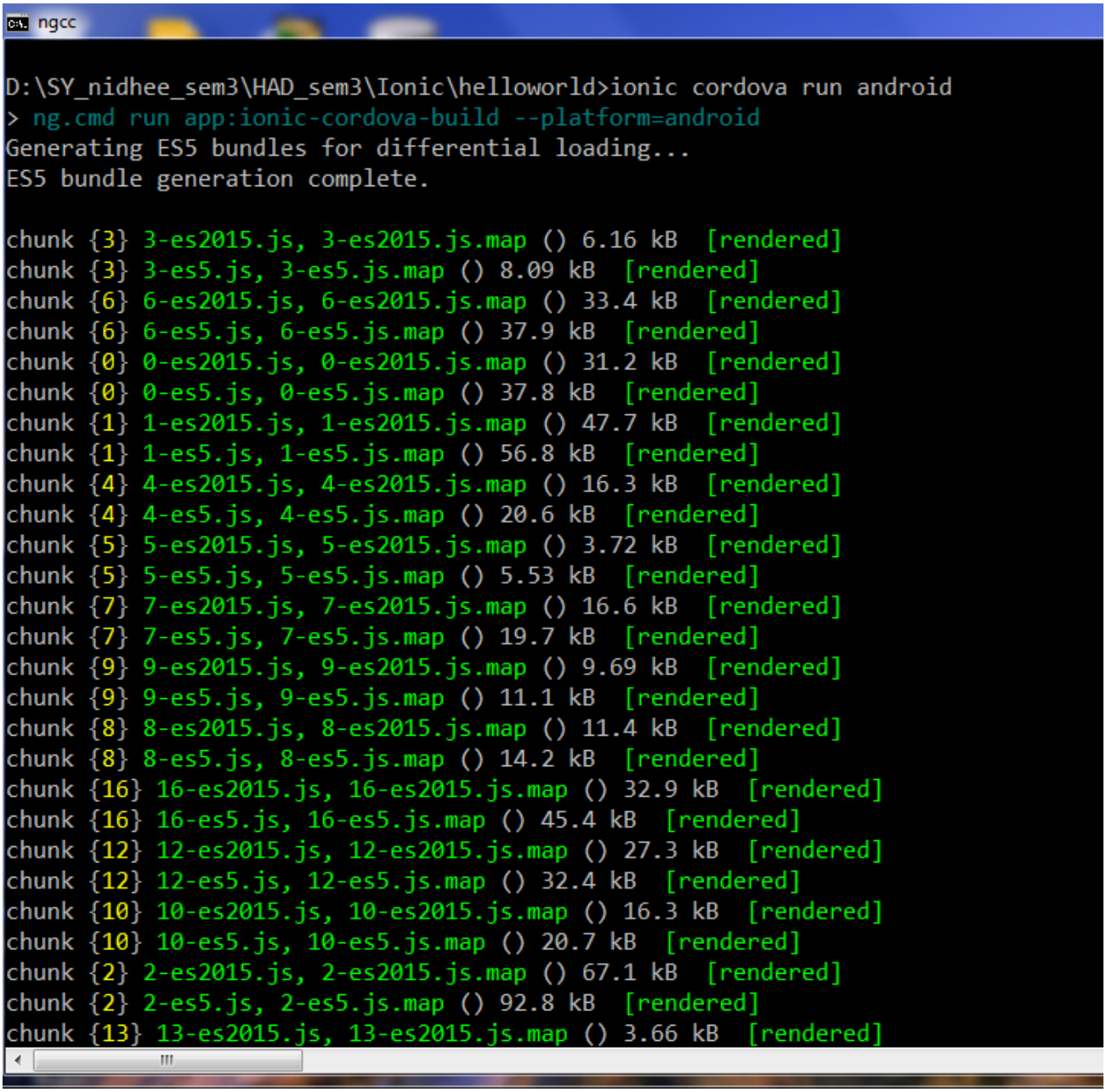
ionic cordova platform add android

```
Nidhi@Nidhi-PC MINGW64 /d/SY_nidhee_sem3/HAD_sem3/Ionic/helloworld (master)
$ ionic cordova platform add android
Platform android already exists.
```

ionic cordova build android

To **run** your app, all you have to do is enable USB debugging and Developer Mode on your **Android** device, then run **ionic cordova run android --device** from the command line.

ionic cordova run android --device




**Once the command runs successfully, the app will be displayed on your android screen.**

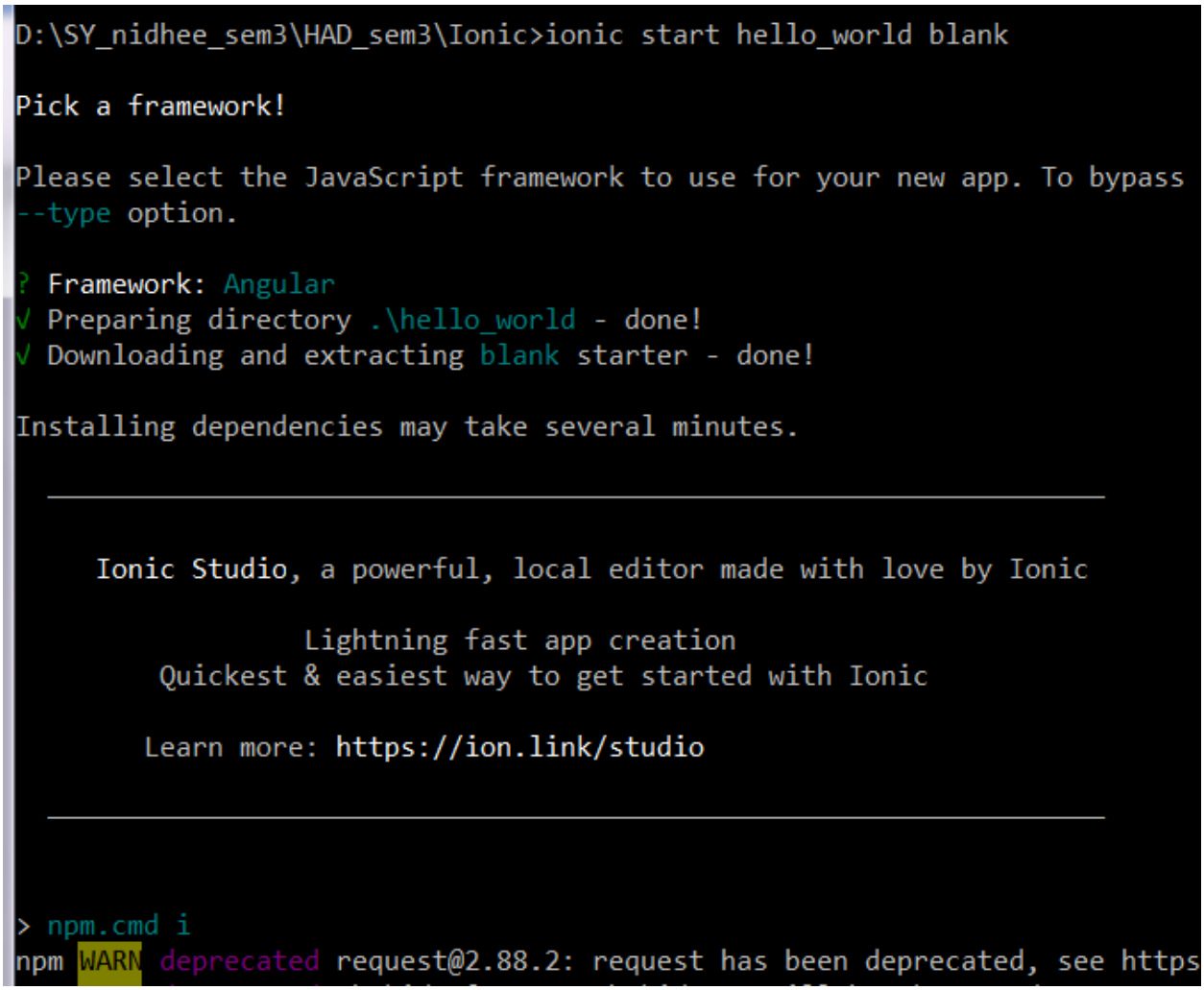
# PRACTICAL 7

## Ionic Create, Generate and Add Pages

### Start/build a new project:

Type the following commands in command prompt,

- ionic start hello_world blank




- Now the project named as hello_world will be build. Change the directory to hello_world,

  cd hello_world

- To start the server type following command,

  ionic serve

**Generate pages:**

Synopsis

ionic generate **[**<type>**]** **[**<name>]

Type the following command in cmd within the project folder to generate a page named as newpage,

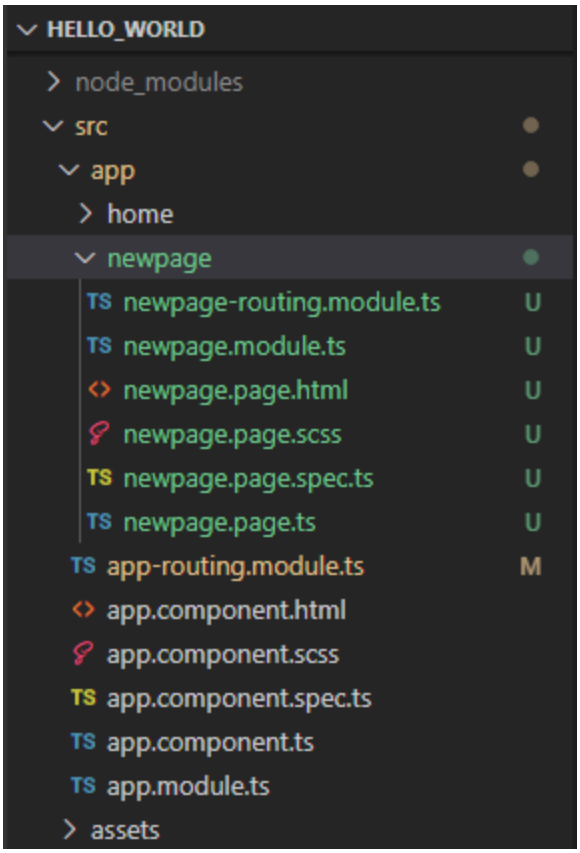
ng generate page newpage

```
D:\SY_nidhee_sem3\HAD_sem3\Ionic>cd hello_world
```

```
D:\SY_nidhee_sem3\HAD_sem3\Ionic\hello_world>ionic generate page newpage
> ng.cmd generate page newpage
CREATE src/app/newpage/newpage-routing.module.ts (351 bytes)
CREATE src/app/newpage/newpage.module.ts (479 bytes)
CREATE src/app/newpage/newpage.page.html (126 bytes)
CREATE src/app/newpage/newpage.page.spec.ts (654 bytes)
CREATE src/app/newpage/newpage.page.ts (260 bytes)
CREATE src/app/newpage/newpage.page.scss (0 bytes)
UPDATE src/app/app-routing.module.ts (736 bytes)
[OK] Generated page!

D:\SY_nidhee_sem3\HAD_sem3\Ionic\hello_world>
```

After generating the newpage ionic will generate many pages for us, our directory structure of the project will look like:




We will make some changes and add some text and a link to navigate the new page that we create just now.

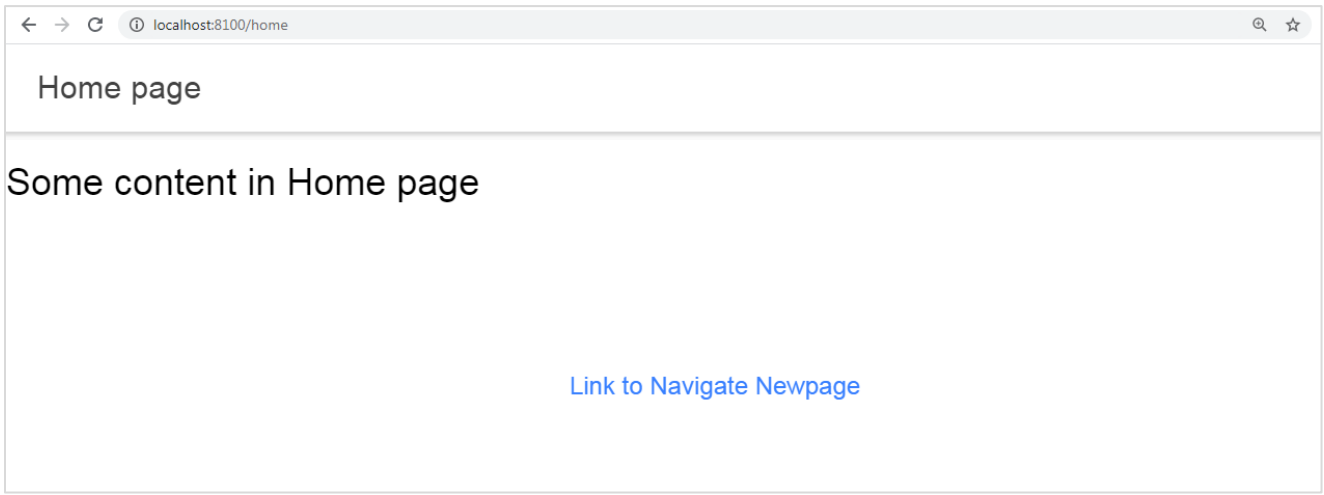
src > app > home > home.page.html > ion-content > div#container > a

```
<ion-header [translucent]="true">
  <ion-toolbar>
    <ion-title>
      Home page
    </ion-title>
  </ion-toolbar>
</ion-header>

<ion-content [fullscreen]="true">
  <ion-header collapse="condense">
    <ion-toolbar>
      <ion-title size="large">Home_page</ion-title>
    </ion-toolbar>
  </ion-header>
<div>
  <h2>Some content in Home page</h2>
</div>
  <div id="container">
    <a href="../newpage/">Link to Navigate Newpage</a>
  </div>
</ion-content>

```

After modifying the content of the files which were created by default in our project we will now run **ionic serve** command to see the changes we made.




We can see that our application is running properly and we can navigate to the newpage from the home page on clicking the link.

So with this we have generated and added pages to our ionic project.

# PRACTICAL 8

## Ionic Use Tabs Starter Template

This is list of official Ionic starter templates, which are ready-to-go starter packs for your next Ionic app.

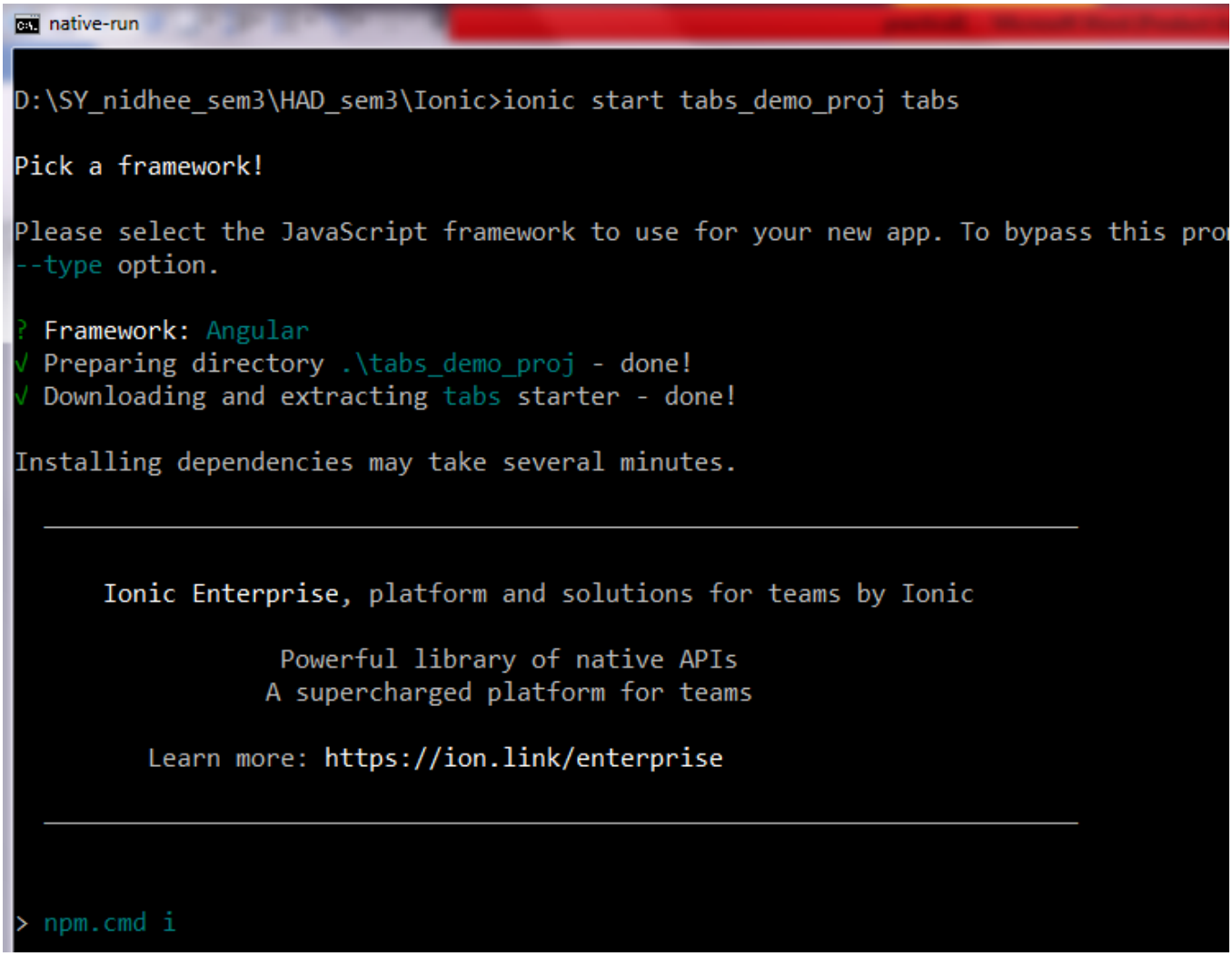
**Ionic Angular**

| Starter | Description |
|---|---|
| tabs | A starting project with a simple tabbed interface |
| blank | A blank starter project |
| sidemenu | A starting project with a side menu with navigation in the content area |
| super | A starting project complete with pre-built pages, providers and best practices for Ionic development. |
| conference | A project that demonstrates a realworld application |
| tutorial | A tutorial based project that goes along with the Ionic documentation |
| aws | AWS Mobile Hub Starter |

Starters are constructed within the Ionic Starters repository by overlaying a starter app onto a set of base files, constructing a compressed archive of the files, and uploading it around the world. The Ionic CLI then downloads and extracts the starter template archive and personalizes files for each new app.

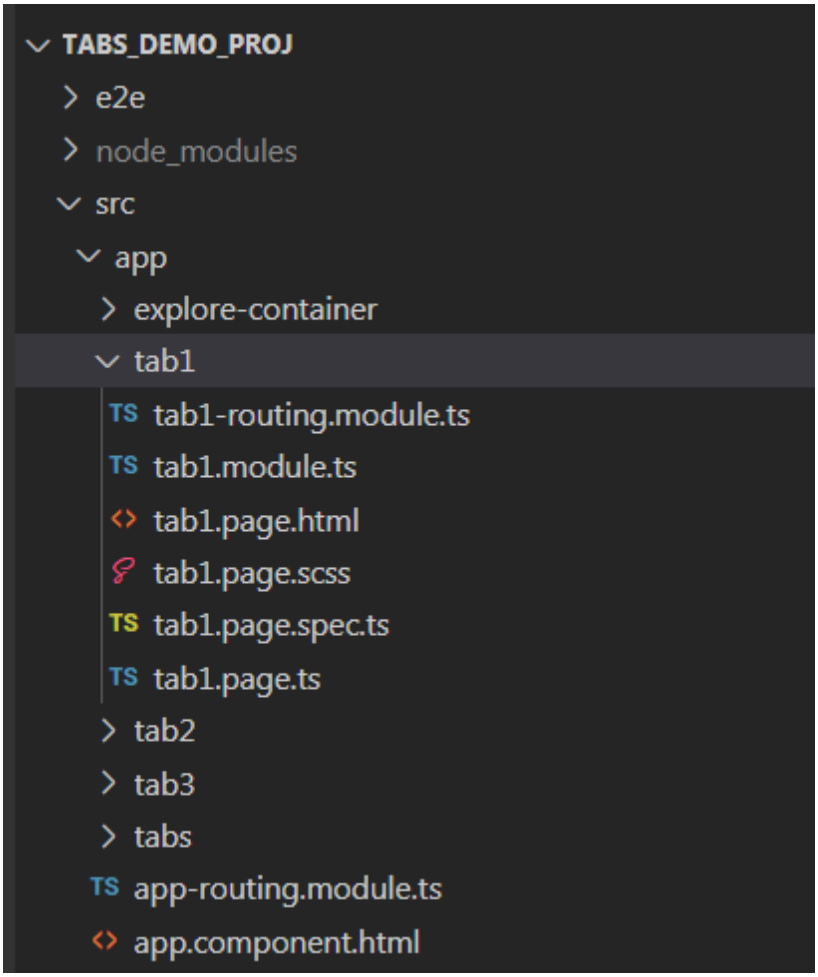
To use **tabs starter template** type the following command on cmd,

**ionic start project_name tabs**

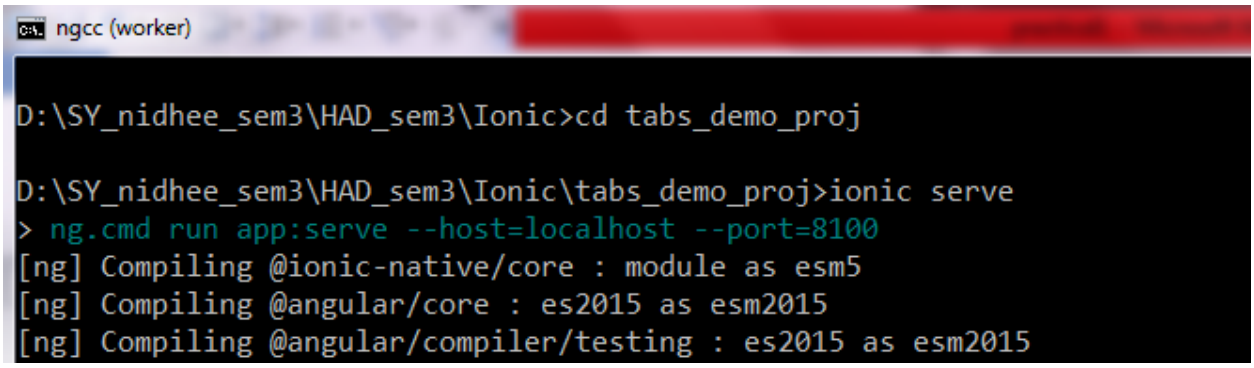


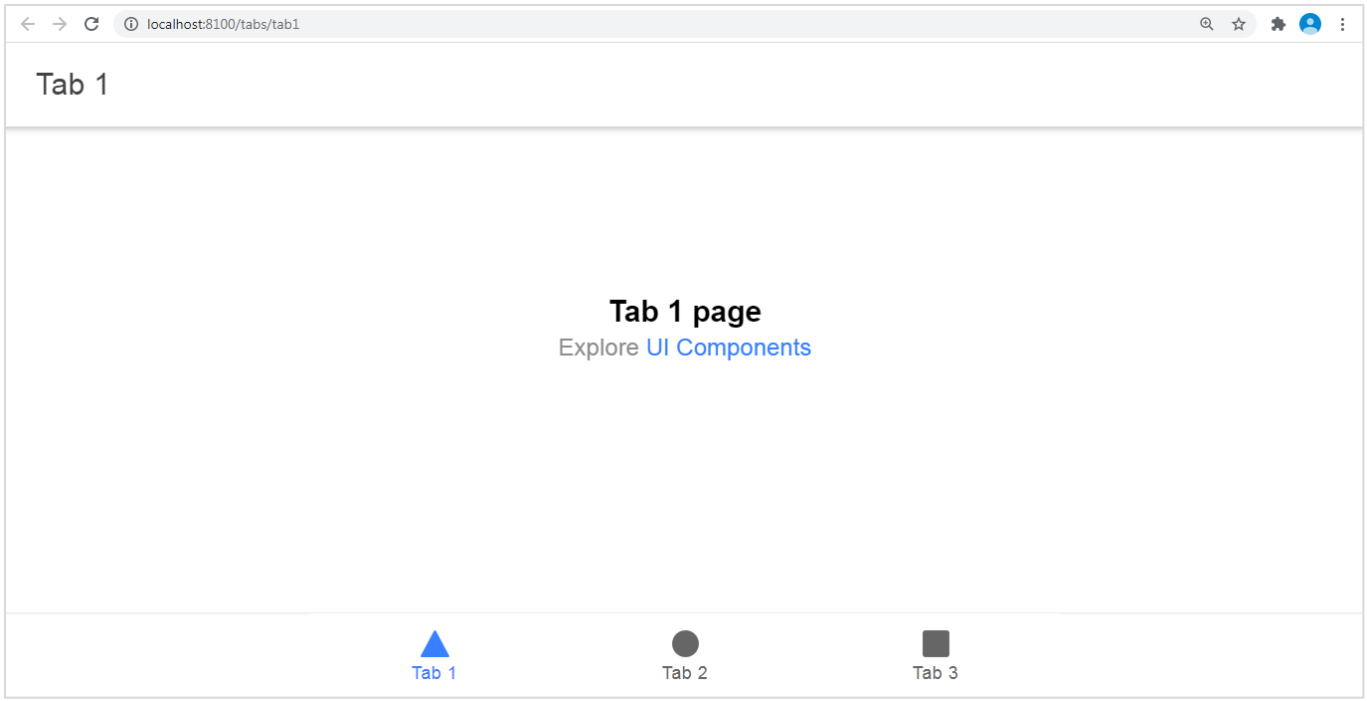

This template creates a starter Ionic app that has 3 different tabs at the bottom of the screen for a user to click and be able to switch between pages.

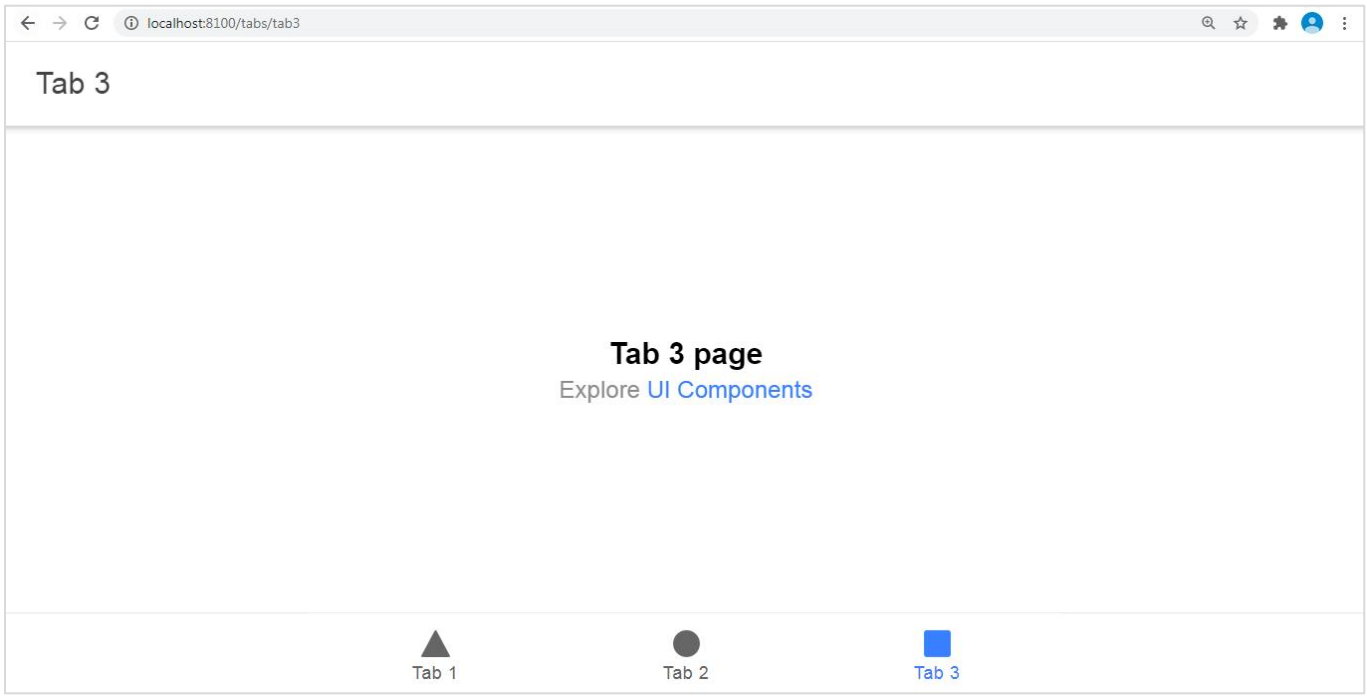
Directory structure would be:




We will run command **ionic serve** to run the app in web browser,




localhost:8100/tabs/tab1
Tab 1
Tab 1 page
Explore UI Components
Tab 1
Tab 2
Tab 3


localhost:8100/tabs/tab3
Tab 3
Tab 3 page
Explore UI Components
Tab 1
Tab 2
Tab 3