



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: Abhay Jha

Roll No: 321

Programme: BSc 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(class code: 2037UCSMD)** for the partial fulfilment of Third Semester of BSc 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. CS Sem- III**Roll No: 321****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 Starter Template	

Practical No. 1

Aim : To display AngularJS Data Bindings.

Theory :

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.

AngularJS follows Two-Way data binding model.

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.

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.

Code :

One-Way Data Binding

```
app.html x
1 <!DOCTYPE html>
2 <html>
3 <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
4 <body>
5
6 <div ng-app="myApp" ng-controller="myCtrl">
7   <p>{{ firstname }}</p>
8 </div>
9
10 <script>
11 var app = angular.module('myApp', []);
12 app.controller('myCtrl', function($scope) {
13   $scope.firstname = "John";
14   $scope.lastname = "Doe";
15 });
16 </script>
17
18 <p>One Way Binding.</p>
19
20 </body>
21 </html>
```

Two-Way Data Binding

```
app.html x
1 <!DOCTYPE html>
2 <html>
3 <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
4 <body>
5
6 <div ng-app="myApp" ng-controller="myCtrl">
7   <p>
8     <input type="text" ng-model="firstname">{{ firstname }}</p>
9 </div>
10
11 <script>
12 var app = angular.module('myApp', []);
13 app.controller('myCtrl', function($scope) {
14   $scope.firstname = "John";
15   $scope.lastname = "Doe";
16 });
17 </script>
18
19 <p>Two Way Binding.</p>
20
21 </body>
22 </html>
```

Output :

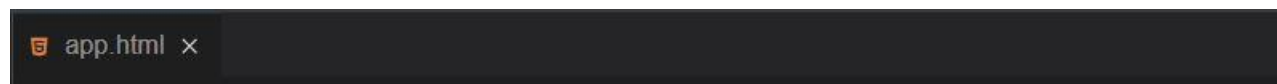
One-Way Data Binding



John

One Way Binding.

Two-Way Data Binding



John John

Two Way Binding.

Practical No. 2

Aim : To display AngularJS Directives.

Theory :

AngularJS facilitates you to extend HTML with new attributes. These attributes are called directives.

There is a set of built-in directive in AngularJS which offers functionality to your applications. You can also define your own directives.

Directives are special attributes starting with ng- prefix. Following are the most common directives:

ng-app: This directive starts an AngularJS Application.

ng-init: This directive initializes application data.

ng-model: This directive defines the model that is variable to be used in AngularJS.

ng-repeat: This directive repeats html elements for each item in a collection.

Code :

```
app.html Directives.html x
1 <!DOCTYPE html>
2 <html>
3 <head><script src = "https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script></head>
4 <body>
5     <h1>Directives Examples</h1>
6     <div ng-app="myApp"
7         ng-init="countries = [{locale:'en-IND',name:'India'},
8                               {locale:'en-USA',name:'USA'},
9                               {locale:'en-AUS',name:'Australia'}]">
10         <p>Enter your Name: <input type="text" ng-model = "firstname"></p>
11         <p>Hello <span ng-bind="firstname">{{firstname}}</span>!</p>
12         <p>List of Countries with locale:</p>
13         <ol>
14             <li ng-repeat = "country in countries">
15                 {{ 'Country: ' + country.name + ', Locale: ' + country.locale }}
16             </li>
17         </ol>
18     </div>
19 <script>
20 var app = angular.module('myApp', []);
21 app.controller('myCtrl', function($scope) {
22     $scope.firstname = "John";
23 });
24 </script>
25 </body>
26 </html>
```

Output :

Directives.html ×

Directives Examples

Enter your Name:

Hello John!

List of Countries with locale:

1. Country: India, Locale: en-IND
2. Country: USA, Locale: en-USA
3. Country: Australia, Locale: en-AUS

Snipping Tool

Practical No. 3

Aim : To display AngularJS Controllers.

Theory :

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.

Code :

```
app.html Directives.html Controllers.html x
1 <!DOCTYPE html>
2 <html>
3 <head><script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script></head>
4 <body>
5   <div ng-app="myApp" ng-controller="myCtrl">
6     <p>
7       First Name : <input type="text" ng-model="firstname"><br>
8       Last Name : <input type="text" ng-model="lastname"><br>
9       {{ firstname + " " + lastname }}
10    </p>
11  </div>
12  <script>
13    var app = angular.module('myApp',[])
14    app.controller('myCtrl',function($scope){
15      $scope.firstname = "John";
16      $scope.lastname = "Doe";
17    });
18  </script>
19 </body>
20 </html>
```

Output :

```
Controllers.html x
```

First Name :

Last Name :

John Doe

Practical No. 4

Aim : To display AngularJS Events.

Theory :

AngularJS includes certain directives which can be used to provide custom behavior on various DOM events, such as click, dblclick, mouseenter etc.

Some More DOM events used :

- ng-blur
- ng-change
- ng-click
- ng-copy
- ng-cut
- ng-dblclick
- ng-focus
- ng-keydown
- ng-keypress
- ng-keyup
- ng-mousedown
- ng-mouseenter
- ng-mouseleave
- ng-mousemove
- ng-mouseover
- ng-mouseup
- ng-paste

Code :

```
Events.html x
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
5 </head>
6 <body>
7   <div ng-app="myApp" ng-controller="myCtrl">
8     <h1>Events Example : ng-click</h1>
9     <button type="submit" ng-click="myFunction()">Click!</button><br>
10    Count : {{count}}
11    <hr>
12    <h1>Events Example : ng-mousemove</h1>
13    <h1 ng-mousemove="count = count + 1">Mouse over me!</h1>
14    Count : {{count}}
15  </div>
16  <script>
17    var app = angular.module('myApp', [])
18    app.controller('myCtrl', function($scope){
19      $scope.count = 0;
20      $scope.myFunction = function(){
21        $scope.count++;
22      };
23    });
24  </script>
25 </body>
26 </html>
```

Output :

```
Events.html x
```

Events Example : ng-click

Click!

Count : 124

Events Example : ng-mousemove

Mouse over me!

Count : 124

Practical No. 5

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

Theory :

What is Hybrid Application?

A (hybrid app) is a software application that combines elements of both native apps and web applications. Hybrid apps are essentially web apps that have been put in a native app shell. Once they are downloaded from an app store and installed locally, the shell is able to connect to whatever capabilities the mobile platform provides through a browser that's embedded in the app. The browser and its plug-ins run on the back end and are invisible to the end user.

Hybrid apps are popular because they allow developers to write code for a mobile app once and still accommodate multiple platforms. Because hybrid apps add an extra layer between the source code and the target platform, they may perform slightly slower than native or web versions of the same app.

How to create an ionic web app.

The requirement for creating an Ionic web app is:

1. Node
2. Node Package Manager(npm)

Steps for creating Ionic Web app:

Open windows command prompt and type the following Commands.

1. Install ionic using : `npm install -g ionic`
2. Get ionic info using : `ionic info`
3. To create an ionic app type : `ionic start <app_name> <template_type>`
4. Then, it will ask to choose a framework, if you want to make it in angular choose angular.
5. Change the folder and go your app_name created using : `cd your_app_name`

Old Roll No. 4023 New Roll No. 321

6. Then to start running the ionic web page in the server type ionic serve in the same folder.

```
C:\Users\user\abhay>cd hello_world

C:\Users\user\abhay\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\user\abhay\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\user\abhay\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 [rendered]
[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) 19.7 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 [rendered]
[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 [rendered]
[ng] chunk {swiper-bundle-95afeea2-js} swiper-bundle-95afeea2-js.js, swiper-bundle-95afeea2-js.js.map (swiper-bundle-95afeea2-js) 200 kB [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: 2020-09-16T14:22:10.982Z - Hash: 1913f2b5807aef1999b0 - Time: 43358ms
[INFO] ... and 42 additional chunks
[ng] : Compiled successfully.

[INFO] Development server running!

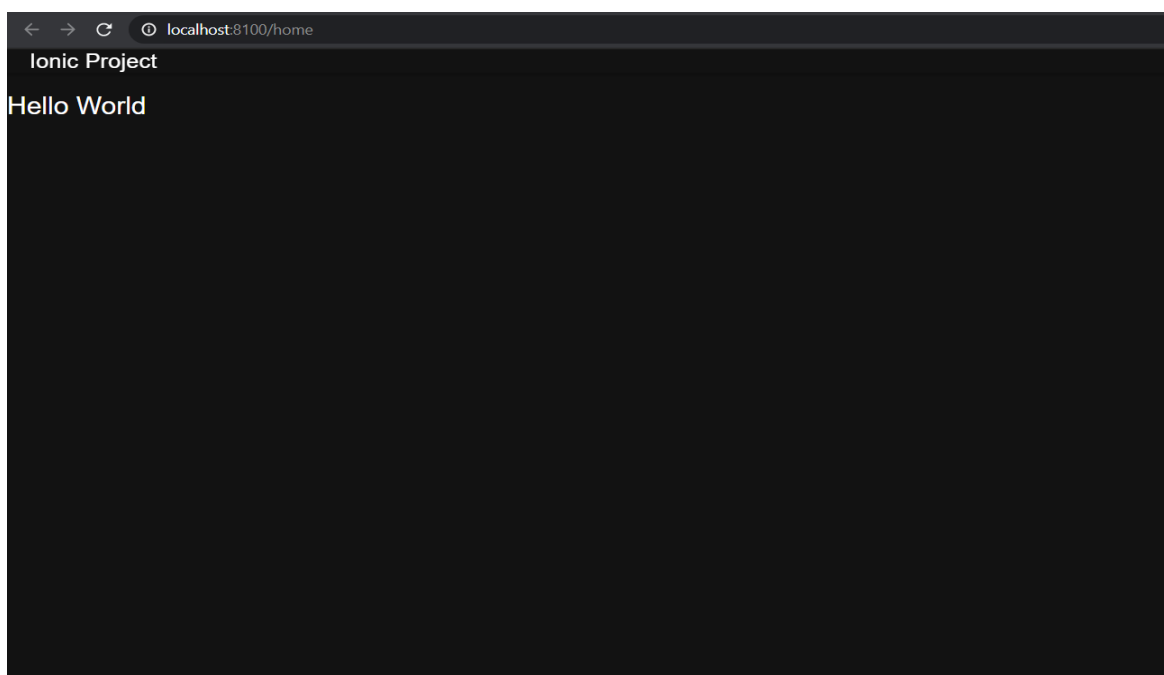
      Local: http://localhost:8100

      Use Ctrl+C to quit this process

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

7. To access the web page go to <http://localhost:8100> in your preferred browser

Output :



Practical No. 6

Aim : Ionic 2/Ionic 4-Adding Cordova Android Platform

Theory :

How to create an ionic Android app :

The requirement for creating an Ionic Android App is:

1. Android studio
2. Gradle
3. Android Device

Steps for creating Ionic Android app :

1. After you have built the web app, next command you should type : ionic cordova platform add android .

This will add Android platform to your web app directory.

2. Now type : ionic cordova build android ,to start building your Android app.

3. Now type : ionic cordova run android , If you have connect your machine to a phone and turned on USB debugging the app will export to your pc and if you are running an emulator the out will be like this.

4. If you are using Android phone, then install the app-debug.apk application in your phone.

It will be available in this path :

C:\Users\user\yourname\hello_world\platforms\android\app\build\outputs\apk\debug\app-debug.apk

Output :



Practical No. 7

Aim : Ionic 2/Ionic 4-Create, Generate and Add Pages

Theory :

You can create pages either manually or generating them using the Ionic CLI v4 which is the recommended method.

In this guide we'll look first at how to create a page manually or generate it with the Ionic CLI, then how to add it to the project.

Go ahead and open your terminal or command prompt and follow the instructions

Generating Pages Using Ionic CLI 4

Step 1 : You can generate a new page by running the following command:

`ionic generate page <pageName>`

Step 2 : For example to generate a contact page, run:

`ionic generate page contact`

Step 3 : This command will create a folder with name *my-page* and three files:

```
|— contact.html
|— contact.scss
└— contact.ts
```

contact.html: contains UI components for the page.

contact.scss: contains sass styles to style your page.

contact.ts: contains TypeScript code.

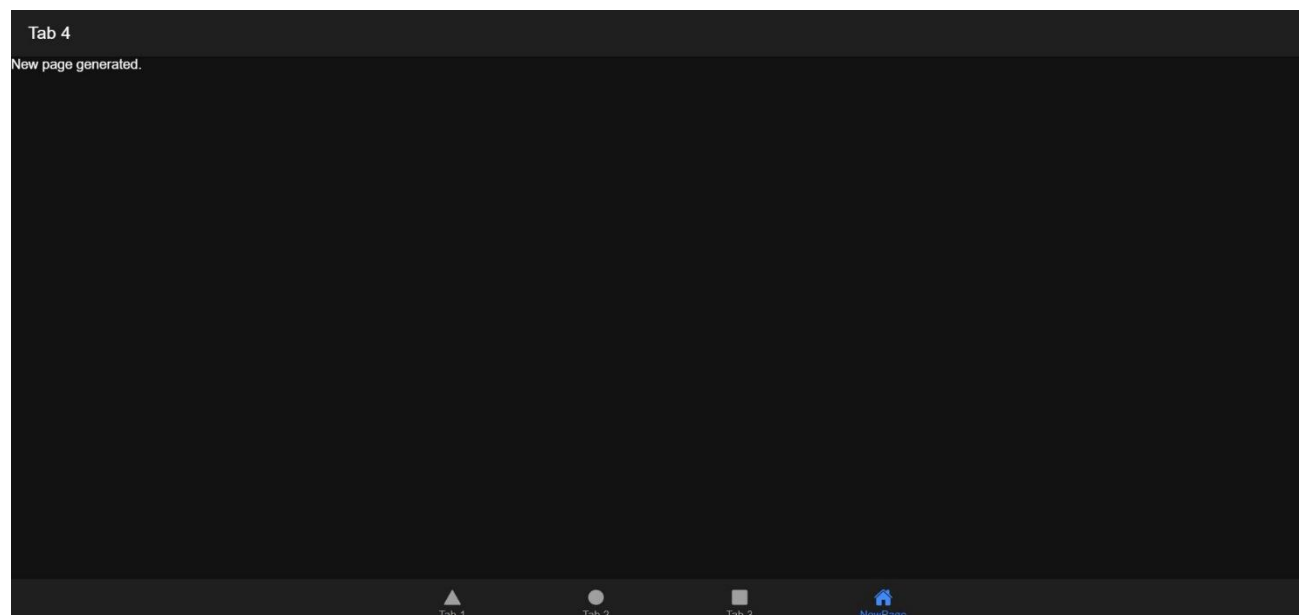
and some more files will be created.

We will be using tabs templates for this practical.

Output :

Old Roll No. 4023

New Roll No. 321



Practical No. 8

Aim : Ionic CLI v3-start command templates and options

Theory :

Ionic starter templates, which are ready-to-go starter packs for your next Ionic app.

There are many templates available and here is the list :

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.

maps : An Ionic starter project using Google Maps and a side menu

There are examples of tabs, blank and sidemenu templates in here :

Steps to create different template:

Step 1: Type the command in command prompt

```
ionic start <project_name> <template_type>
```

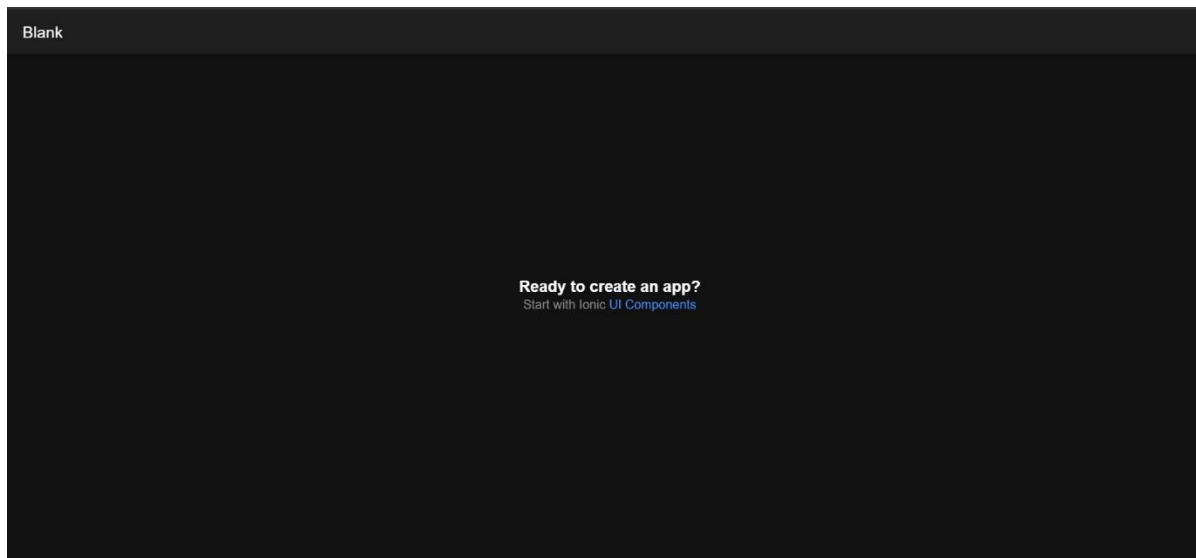
For Example:

```
ionic start blank_template blank
```

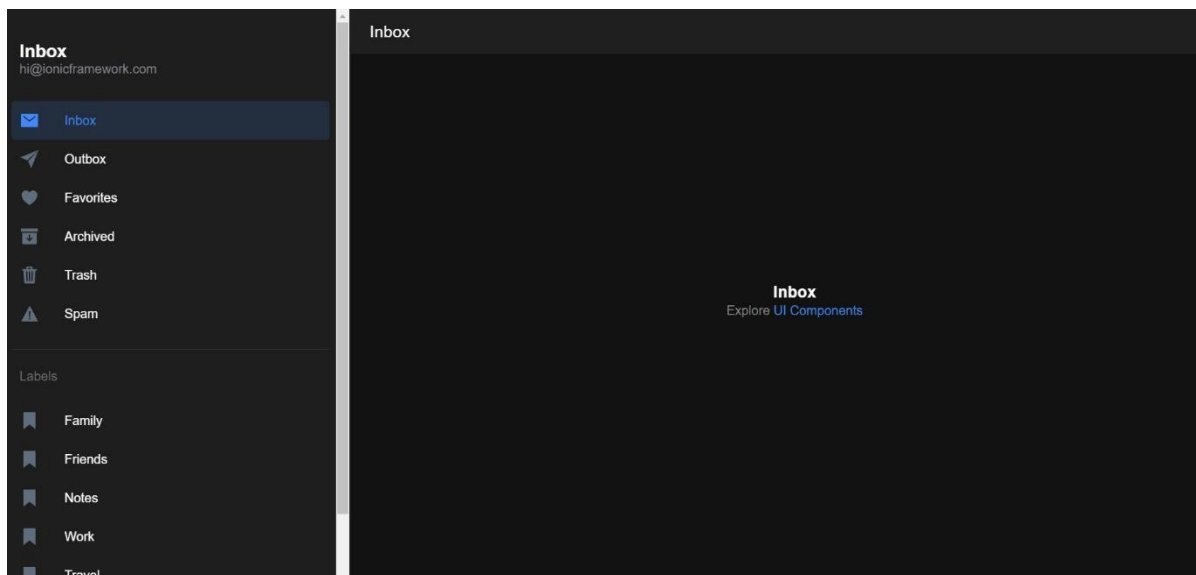
Step 2: Now, type ionic serve to run the command to serve the site.

Output :

Blank Template:



Sidemenu Template:



Tabs Template:

