



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: Ms. SACHI VIRESH SHAH

Roll No: 381

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 (CourseCode: 2037UCSMD)** for the partial fulfillment 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)

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	

Hybrid App Development

Practical 1

Aim: AngularJS Data Binding

Theory:

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

AngularJS applications usually have a data model. The data model is a collection of data available for the application.

The HTML container where the AngularJS application is displayed is 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. Or you can use the **ng-model** directive on HTML controls to bind the model to the view.

The ng-model Directive

Use the **ng-model** directive to bind data from the model to the view on HTML controls (input, select, textarea). The **ng-model** directive provides a two-way binding between the model and the view.

Two-way Binding

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

When data in the model changes, the *view* reflects the change, and when data in the view changes, the model is updated as well. This happens immediately and automatically, which makes sure that the model and the view is updated at all times.

Code: Two-way Data Binding

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>AngularJs Two Binding Example</title>
5     <script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>
6     <script type="text/javascript">
7
8       var app = angular.module('myapp', []);
9       app.controller('mycontroller', function ($scope) {
10         $scope.name = '';
11       });
12     </script>
13   </head>
14   <body ng-app="myapp">
15
16     <h1>AngularJs Two Way Binding</h1>
17     <div ng-controller="mycontroller">
18       <!-- whenever we change input control value automatically the appearance value also will
19       get changed -->
20       Enter Name : <input type="text" ng-model="name" />
21       <p>
22         Entered Name:   {{ name }}
23       </p>
24     </div>
25   </body>
26 </html>
```

Output:

AngularJs Two Binding Example

File | C:/Users/Dhwani/Desktop/sachiit/sem3_bscit/hybridAppDev/practicals/prac5_8/prac5_binding.html

Apps | New Tab | Google | Welcome to Nagin... | E | Dashboard | Nagindas Khandwal... | Convert Word to P... | CSS Font Stack: We... | Goog

AngularJs Two Way Binding

Enter Name :

Entered Name: Sachi Shah

Practical 2

Aim: AngularJS Directives

Theory:

AngularJS lets you extend HTML with new attributes called **Directives**. AngularJS has a set of built-in directives which offers functionality to your applications. **AngularJS also lets you define your own directives.** AngularJS directives are extended HTML attributes with the prefix **ng-**.

1. The **ng-app** directive initializes an AngularJS application. The **ng-app** directive also tells AngularJS that the <div> element is the "owner" of the AngularJS application.
2. The **ng-init** directive initializes application data.
3. The **ng-model** directive binds the value of HTML controls (input, select, textarea) to application data.
4. The **ng-repeat** directive repeats an HTML element. The ng-repeat directive actually clones HTML elements once for each item in a collection.

The ng-app Directive

The **ng-app** directive defines the root element of an AngularJS application. The **ng-app** directive will auto-bootstrap (automatically initialize) the application when a web page is loaded.

The ng-init Directive

The **ng-init** directive defines initial values for an AngularJS application. Normally, you will not use ng-init. You will use a controller or module instead.

The ng-model Directive

The **ng-model** directive binds the value of HTML controls (input, select, textarea) to application data. The ng-model directive can also:

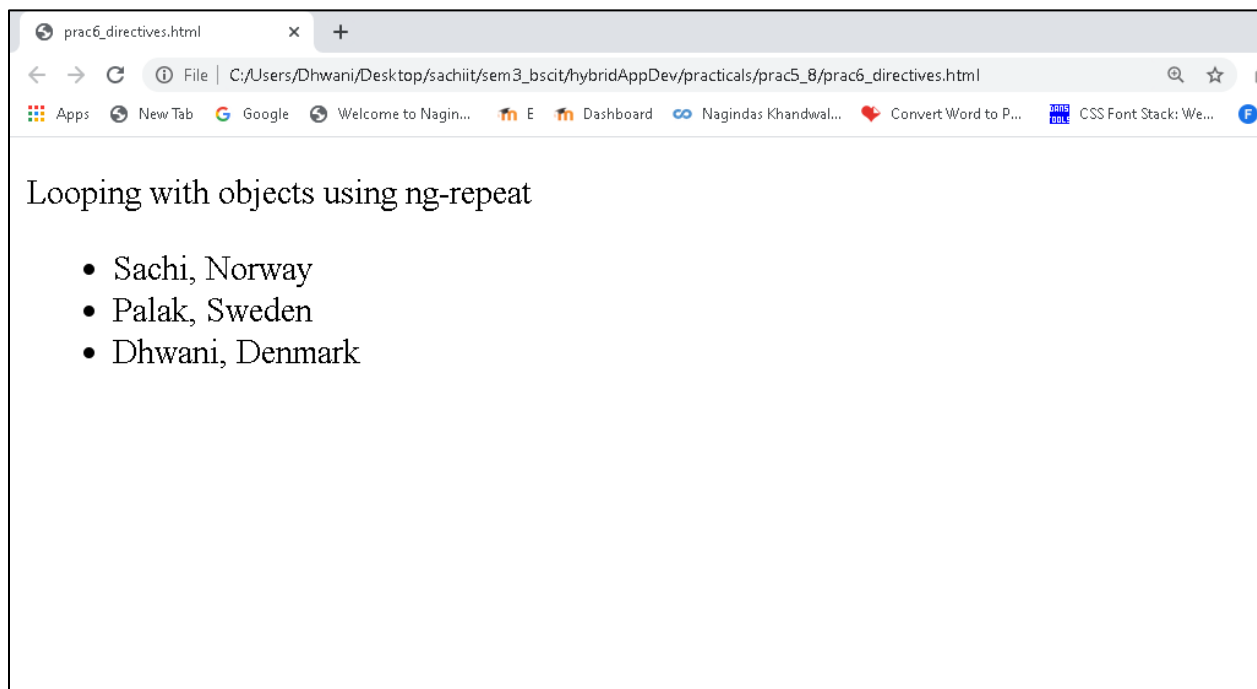
- Provide type validation for application data (number, email, required).
- Provide status for application data (invalid, dirty, touched, error).
- Provide CSS classes for HTML elements.
- Bind HTML elements to HTML forms.

Create New Directives

In addition to all the built-in AngularJS directives, you can create your own directives. New directives are created by using the **.directive** function. To invoke the new directive, make an HTML element with the same tag name as the new directive.

Code:

```
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     <!-- The ng-app directive initializes an AngularJS application.
7     The ng-init directive initializes application data. -->
8     <div ng-app="" ng-init="names=[
9       {name:'Sachi',country:'Norway'},
10      {name:'Palak',country:'Sweden'},
11      {name:'Dhwani',country:'Denmark'}]">
12
13       <p>Looping with objects using ng-repeat</p>
14       <ul>
15         <!-- The ng-repeat directive used on an array of objects: -->
16
17         <li ng-repeat="x in names">
18           {{ x.name + ', ' + x.country }}
19         </li>
20       </ul>
21     </div>
22   </body>
23 </html>
24
```

Output:

Practical 3

Aim: AngularJS Controllers

Theory:

AngularJS controllers **control the data** of AngularJS applications. AngularJS controllers are regular JavaScript Objects. AngularJS applications are controlled by controllers. The **ng-controller** directive defines the application controller. A controller is a **JavaScript Object**, created by a standard JavaScript **object constructor**. A controller can also have methods (variables as functions). In larger applications, it is common to store controllers in external files.

Use controllers to:

- Set up the initial state of the **\$scope** object.
- Add behavior to the \$scope object.

Do not use controllers to:

- Manipulate DOM — Controllers should contain only business logic. Putting any presentation logic into Controllers significantly affects its testability. AngularJS has data binding for most cases and directives to encapsulate manual DOM manipulation.
- Format input — Use AngularJS form controls instead.
- Filter output — Use AngularJS filters instead.
- Share code or state across controllers — Use AngularJS services instead.
- Manage the life-cycle of other components (for example, to create service instances).

In general, a Controller shouldn't try to do too much. It should contain only the business logic needed for a single view. The most common way to keep Controllers slim is by encapsulating work that doesn't belong to controllers into services and then using these services in Controllers via dependency injection.

Code:

```
1 <html>
2   <head>
3     <title>Angular JS Controller</title>
4     <script src = "https://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js">
5     </script>
6   </head>
7   <body>
8     <h2>AngularJS Controller</h2>
9     <!-- we declare a controller named studentController, using the ng-controller directive. -->
10    <div ng-app = "mainApp" ng-controller = "studentController">
11      Enter first name: <input type = "text" ng-model = "student.firstName"><br><br>
12      Enter last name: <input type = "text" ng-model = "student.lastName"><br><br>
13      You are entering: {{student.fullName()}}
14    </div>
15
16    <script>
17      var mainApp = angular.module("mainApp", []);
18      mainApp.controller('studentController', function($scope) {
19        $scope.student = {
20          firstName: "",
21          lastName: "",
22
23          fullName: function() {
24            var studentObject;
25            studentObject = $scope.student;
26            return studentObject.firstName + " " + studentObject.lastName;}
27        };
28      });
29    </script>
30  </body>
31 </html>
```

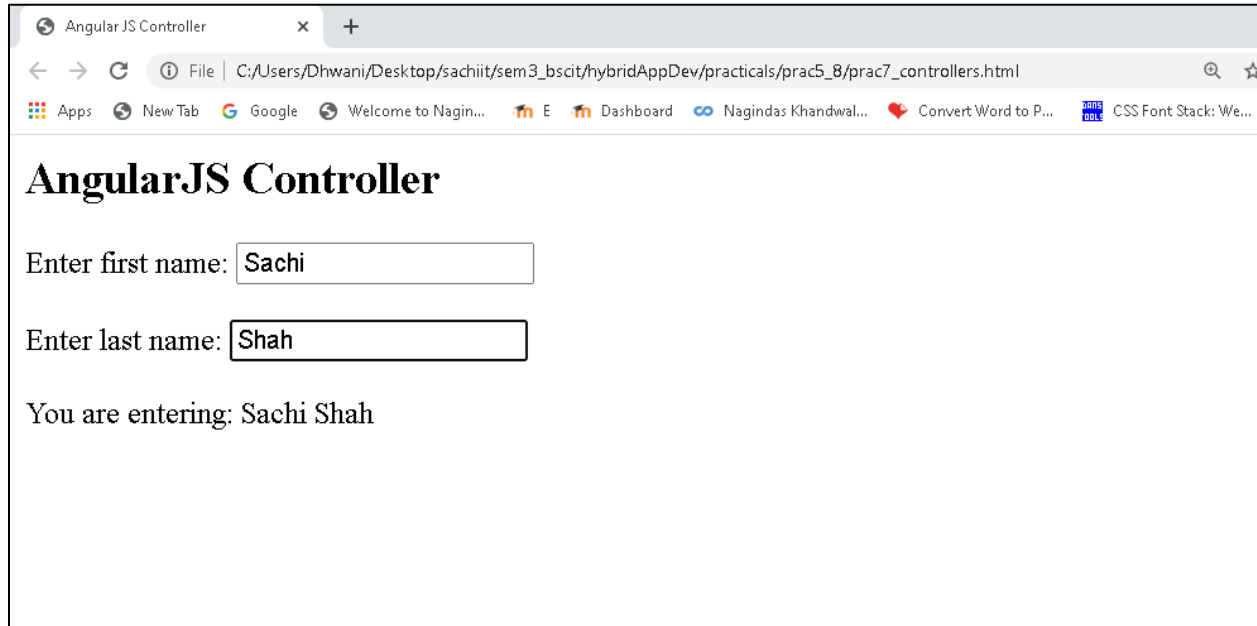
NAME: SACHI VIRESH SHAH

NEW ROLL NO: 381

CLASS: SYIT

OLD ROLL NO: 3066

Output:



AngularJS Controller

Enter first name:

Enter last name:

You are entering: Sachi Shah

Practical 4

Aim: AngularJS Events

Theory:

AngularJS has its own HTML events directives. The event directives allow us to run AngularJS functions at certain user events. An AngularJS event will not overwrite an HTML event, both events will be executed.

Mouse Events

Mouse events occur when the cursor moves over an element, in this order:

- ng-mouseover
- ng-mouseenter
- ng-mousemove
- ng-mouseleave
- ng-mousedown
- ng-mouseup
- ng-click

The ng-click Directive

The **ng-click** directive defines AngularJS code that will be executed when the element is being clicked.

Toggle, True/False

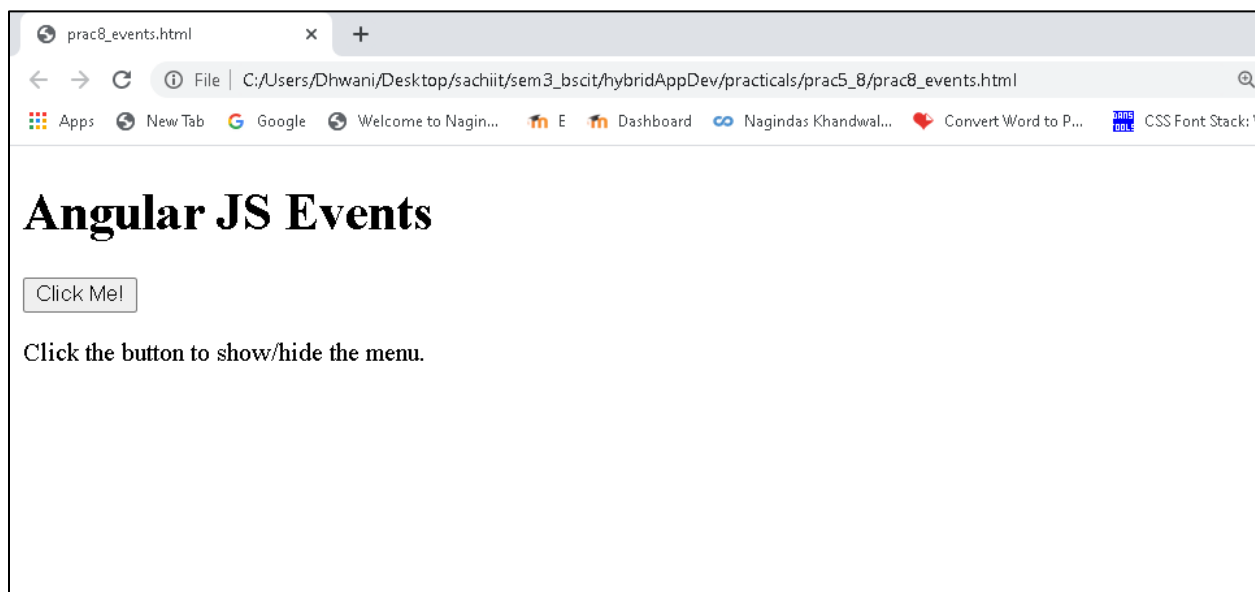
If you want to show a section of HTML code when a button is clicked, and hide when the button is clicked again, like a dropdown menu, make the button behave like a toggle switch

\$event Object

You can pass the **\$event** object as an argument when calling the function.

Code:

```
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     <h1>Angular JS Events</h1>
6     <div ng-app="myApp" ng-controller="myCtrl">
7       <button ng-click="myFunc()">Click Me!</button>
8       <div ng-show="showMe">
9         <h1>Menu:</h1>
10        <div>Pizza</div>
11        <div>Pasta</div>
12        <div>Noodles</div>
13      </div>
14    </div>
15    <script>
16      var app = angular.module('myApp', []);
17      app.controller('myCtrl', function($scope) {
18        // The showMe variable starts out as the Boolean value false.
19        // The myFunc function sets the showMe variable to the opposite of what it is,
20        // by using the ! (not) operator.
21        $scope.showMe = false;
22        $scope.myFunc = function() {
23          $scope.showMe = !$scope.showMe;
24        };
25      </script>
26      <p>Click the button to show/hide the menu.</p>
27    </body>
28  </html>
```

Output: Before clicking click Me !

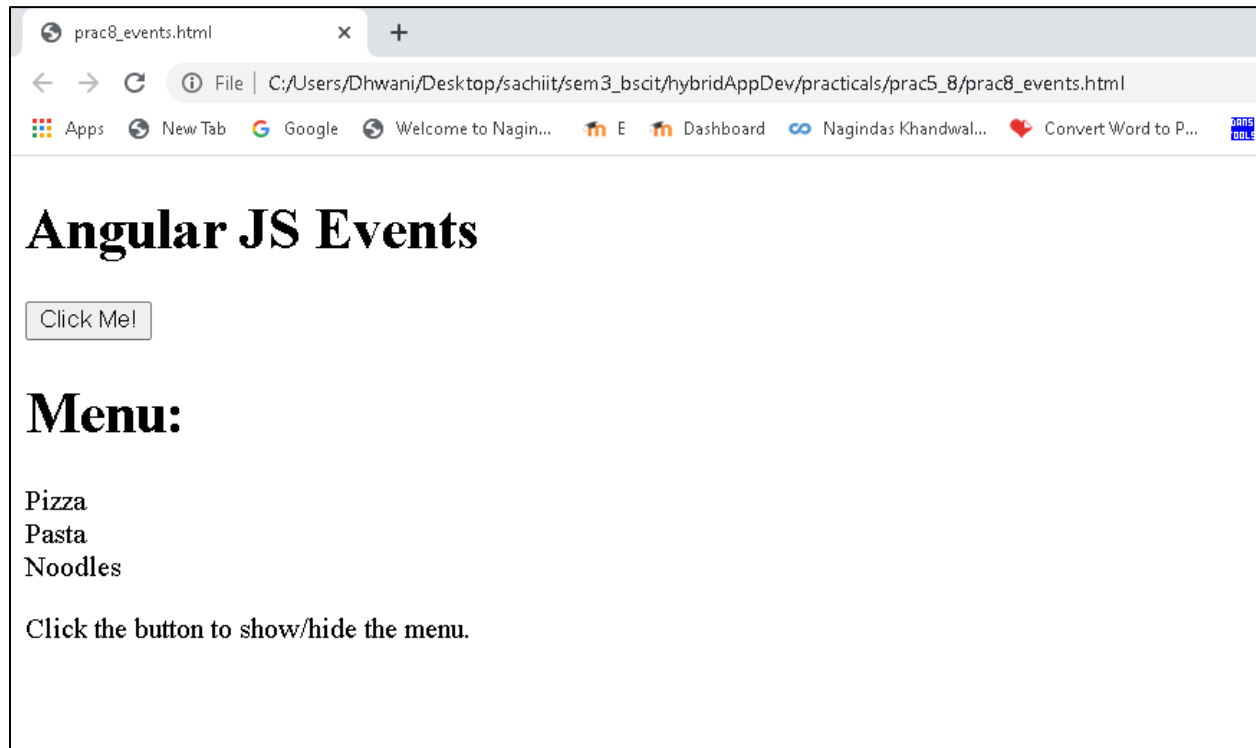
NAME: SACHI VIRESH SHAH

NEW ROLL NO: 381

CLASS: SYIT

OLD ROLL NO: 3066

Output: After clicking click Me!



Practical 5

Aim: Ionic Create and build first project or application (Android)

Theory:

Hybrid App Development helps to deploy the code on different platforms using one source code.

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

Commands to setup Ionic project:

1. Npm install -g ionic Cordova
2. Ionic start hello_world blank
(blank is the template, hello_world folder will be created)
3. cd hello_world
4. Ionic serve

NAME: SACHI VIRESH SHAH

NEW ROLL NO: 381

CLASS: SYIT

OLD ROLL NO: 3066

Steps:

```
C:\Users\Dhwani>npm install -g ionic cordova
npm WARN deprecated ionic@5.4.16: The Ionic CLI now uses @ionic/cli
npm WARN deprecated request@2.88.2: request has been deprecated
npm WARN deprecated har-validator@5.1.5: this library is no longer maintained
[.....] | extract:underscore: sill extract underscore
[.....] / refresh-package-json:ms: sill refresh-package-json
es\cordova\bin\cordova
C:\Program Files\nodejs\ionic -> C:\Program Files\nodejs\node_
+ cordova@10.0.0
+ ionic@5.4.16
added 634 packages from 321 contributors in 337.66s

C:\Users\Dhwani>ionic start hello_world blank

Pick a framework!
```

```
C:\Users\Dhwani>cd hello_world
```

```
C:\Users\Dhwani\hello_world>ionic serve
> ng.cmd run app:serve --host=localhost --port=8100
[ng] Compiling @angular/compiler/testing : es2015 as esm2015
[ng] Compiling @angular/core : es2015 as esm2015
[ng] Compiling @ionic-native/core : module as esm5
[ng] Compiling @angular/common : es2015 as esm2015
[ng] Compiling @ionic-native/splash-screen : module as esm5
[ng] Compiling @angular/core/testing : es2015 as esm2015
[ng] Compiling @ionic-native/status-bar : module as esm5
[ng] Compiling @angular/platform-browser : es2015 as esm2015
[ng] Compiling @angular/router : es2015 as esm2015
```

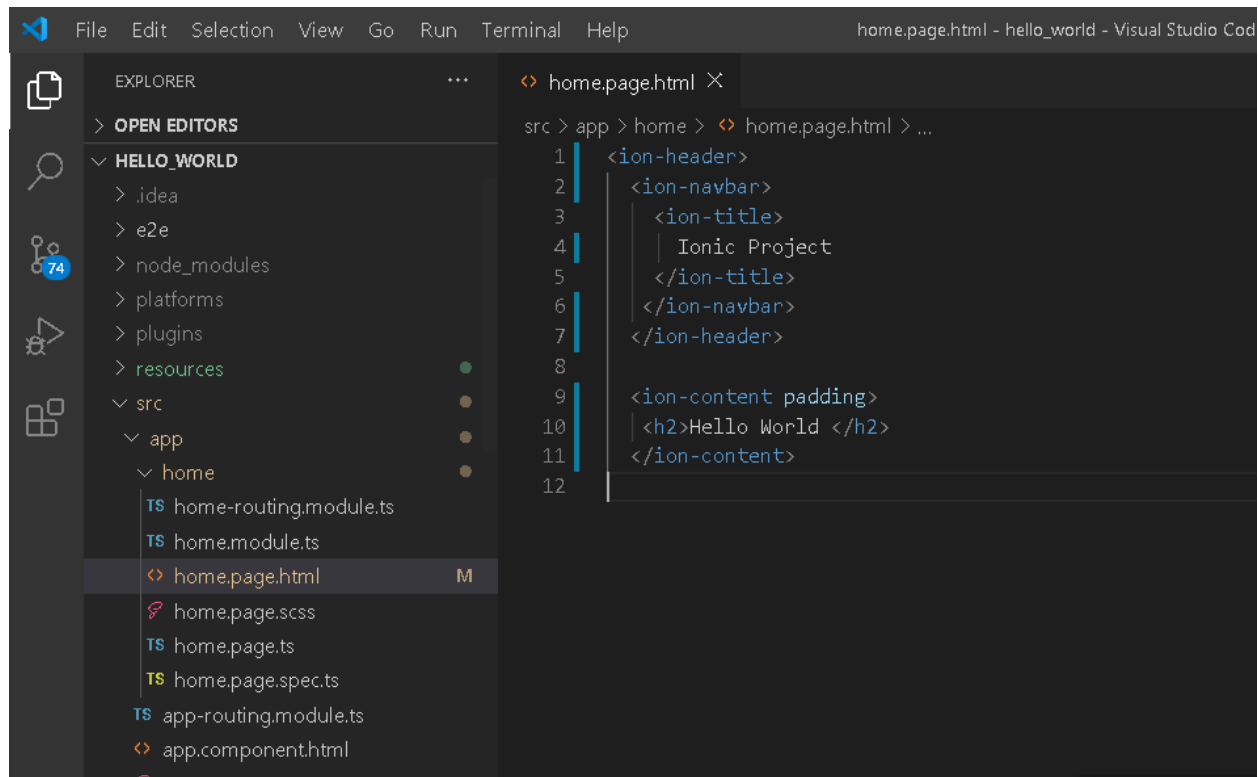
NAME: SACHI VIRESH SHAH

NEW ROLL NO: 381

CLASS: SYIT

OLD ROLL NO: 3066

Code:



The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays the project structure for 'HELLO_WORLD'. The file 'home.page.html' is selected under the 'home' directory. The main editor area shows the content of 'home.page.html', which is an Ionic page template. The code includes an ion-header with an ion-navbar containing an ion-title 'Ionic Project'. The main content area has an ion-content with a padding and an h2 element displaying 'Hello World'.

```
src > app > home > home.page.html > ...
1 | <ion-header>
2 |   <ion-navbar>
3 |     <ion-title>
4 |       Ionic Project
5 |     </ion-title>
6 |   </ion-navbar>
7 | </ion-header>
8 |
9 | <ion-content padding>
10 |   <h2>Hello World </h2>
11 | </ion-content>
12 |
```

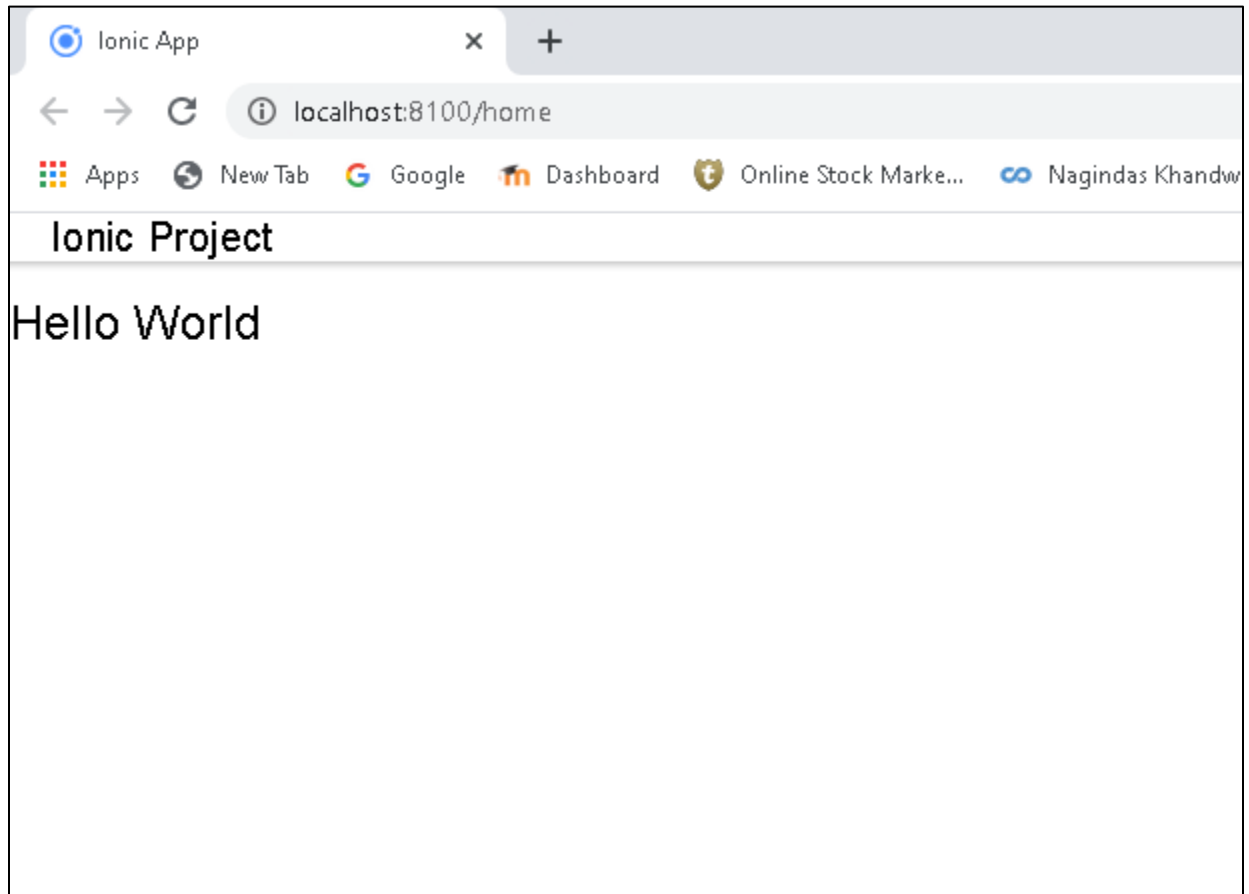

NAME: SACHI VIRESH SHAH

NEW ROLL NO: 381

CLASS: SYIT

OLD ROLL NO: 3066

Output:



Practical 6

Aim: Ionic Adding Cordova Android Platform

Theory:

Commands to setup Ionic project:

1. Npm install -g ionic Cordova
2. Ionic start hello_world blank
(blank is the template, hello_world folder will be created)
3. cd hello_world
4. Ionic serve
5. Ionic Cordova build android -
6. Now to run the ionic Cordova app on android, we type

Ionic Cordova run android

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.

Another way is to run by setting up an emulator in the Android Studio.

NAME: SACHI VIRESH SHAH

NEW ROLL NO: 381

CLASS: SYIT

OLD ROLL NO: 3066

Steps:

```
C:\Users\Dhwani>npm install -g ionic cordova
npm WARN deprecated ionic@5.4.16: The Ionic CLI now uses @ionic/cli
npm WARN deprecated request@2.88.2: request has been deprecated
npm WARN deprecated har-validator@5.1.5: this library is no longer maintained
[.....] | extract:underscore: sill extract underscore
[.....] / refresh-package-json:ms: sill refresh-package-json
es\cordova\bin\cordova
C:\Program Files\nodejs\ionic -> C:\Program Files\nodejs\node_modules
+ cordova@10.0.0
+ ionic@5.4.16
added 634 packages from 321 contributors in 337.66s

C:\Users\Dhwani>ionic start hello_world blank

Pick a framework!
```

```
C:\Users\Dhwani>cd hello_world
```

```
C:\Users\Dhwani\hello_world>ionic serve
> ng.cmd run app:serve --host=localhost --port=8100
[ng] Compiling @angular/compiler/testing : es2015 as esm2015
[ng] Compiling @angular/core : es2015 as esm2015
[ng] Compiling @ionic-native/core : module as esm5
[ng] Compiling @angular/common : es2015 as esm2015
[ng] Compiling @ionic-native/splash-screen : module as esm5
[ng] Compiling @angular/core/testing : es2015 as esm2015
[ng] Compiling @ionic-native/status-bar : module as esm5
[ng] Compiling @angular/platform-browser : es2015 as esm2015
[ng] Compiling @angular/router : es2015 as esm2015
```

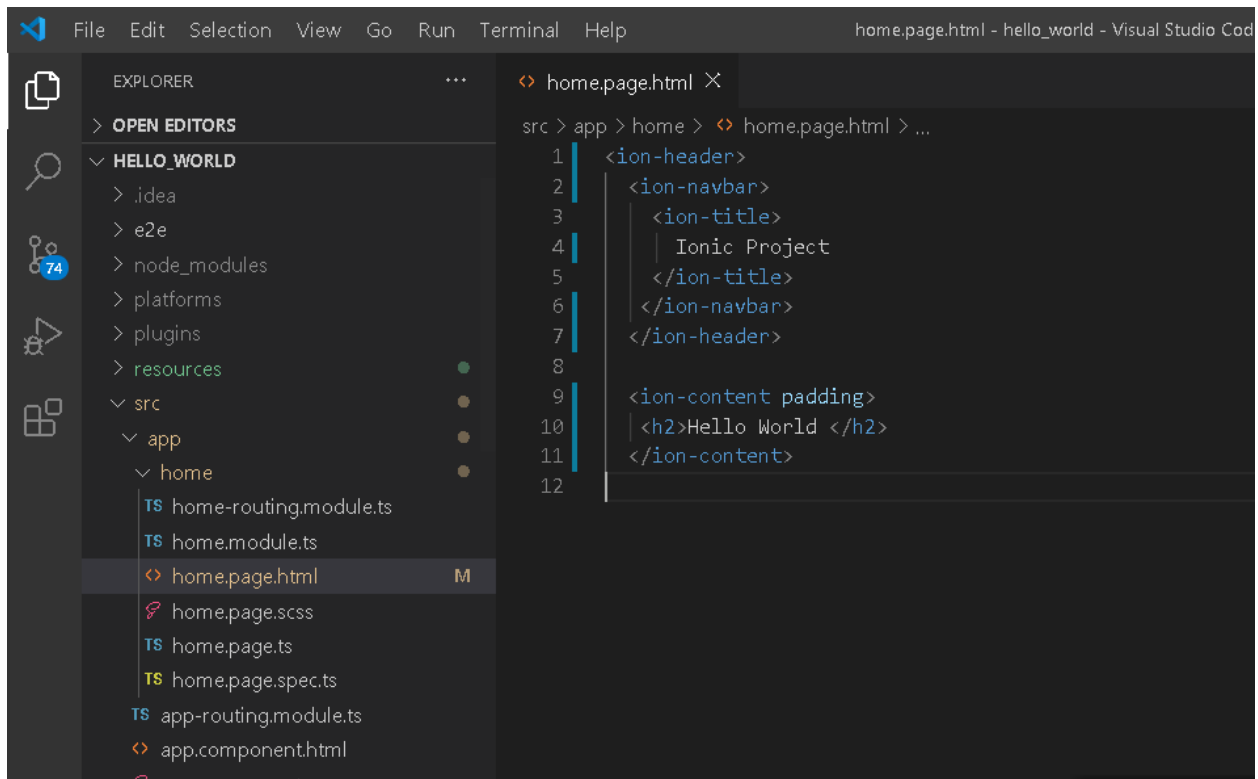
NAME: SACHI VIRESH SHAH

NEW ROLL NO: 381

CLASS: SYIT

OLD ROLL NO: 3066

Code:



The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays the project structure under 'HELLO_WORLD'. The 'src' folder is expanded, showing 'app' > 'home'. The file 'home.page.html' is selected and highlighted in blue. The main editor area shows the content of 'home.page.html' with the following HTML code:

```
src > app > home > home.page.html > ...
1  <ion-header>
2    <ion-navbar>
3      <ion-title>
4        Ionic Project
5      </ion-title>
6    </ion-navbar>
7  </ion-header>
8
9  <ion-content padding>
10    <h2>Hello World </h2>
11  </ion-content>
12
```

NAME: SACHI VIRESH SHAH

NEW ROLL NO: 381

CLASS: SYIT

OLD ROLL NO: 3066

Steps to add Cordova Android Platform

```
Terminal
+ Microsoft Windows [Version 10.0.18362.1016]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Dhwani\hello_world>ionic cordova build android
> ng.cmd run app:ionic-cordova-build --platform=android
Generating ES5 bundles for differential loading...
ES5 bundle generation complete.

chunk {polyfills} polyfills-es2015.js, polyfills-es2015.js.map (polyfills) 268 kB [initial] [rendered]
chunk {polyfills-es5} polyfills-es5.js, polyfills-es5.js.map (polyfills-es5) 735 kB [initial] [rendered]
chunk {4} 4-es2015.js, 4-es2015.js.map () 16.3 kB [rendered]
chunk {4} 4-es5.js, 4-es5.js.map () 20.6 kB [rendered]
chunk {3} 3-es2015.js, 3-es2015.js.map () 6.16 kB [rendered]
chunk {3} 3-es5.js, 3-es5.js.map () 8.09 kB [rendered]
chunk {0} 0-es2015.js, 0-es2015.js.map () 31.2 kB [rendered]
```

```
Task :app:compileDebugJavaWithJavac
Note: Some input files use or override a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

Deprecated Gradle features were used in this build, making it incompatible with Gradle 7.0.
Use '--warning-mode all' to show the individual deprecation warnings.
See https://docs.gradle.org/6.5/userguide/command_line_interface.html#sec:command_line_warnings

BUILD SUCCESSFUL in 7m 8s
40 actionable tasks: 40 executed
Built the following apk(s):
  C:\Users\Dhwani\hello_world\platforms\android\app\build\outputs\apk\debug\app-debug.apk
```

```
[ng] Date: 2020-09-12T13:10:16.347Z - Hash: 4ce19082dfe7b1ee10c9 - Time: 68777ms
[INFO] ... and 42 additional chunks
[ng] : Compiled successfully.

[INFO] Development server running!

Local: http://localhost:8100

Use Ctrl+C to quit this process
```

NAME: SACHI VIRESH SHAH

NEW ROLL NO: 381

CLASS: SYIT

OLD ROLL NO: 3066

```
C:\Users\Dhwani\hello_world>ionic cordova run android
[INFO] Hardware device(s) found for android. Using --device.
> ng.cmd run app:ionic-cordova-build --platform=android
Generating ES5 bundles for differential loading...
ES5 bundle generation complete.
```

```
> cordova.cmd build android --device
Checking Java JDK and Android SDK versions
ANDROID_SDK_ROOT=undefined (recommended setting)
ANDROID_HOME=undefined (DEPRECATED)
Using Android SDK: C:\Users\Dhwani\AppData\Local\Android\sdk
Subproject Path: CordovaLib
Subproject Path: app

Deprecated Gradle features were used in this build, making it incompatible with Gradle 7.0.
Use '--warning-mode all' to show the individual deprecation warnings.
See https://docs.gradle.org/6.5/userguide/command_line_interface.html#sec:command_line_warnings

BUILD SUCCESSFUL in 8s
40 actionable tasks: 40 up-to-date
Built the following apk(s):
   C:\Users\Dhwani\hello_world\platforms\android\app\build\outputs\apk\debug\app-debug.apk
> native-run.cmd android --app platforms\android\app\build\outputs\apk\debug\app-debug.apk --device
[native-run] Selected hardware device d9cec42f
[native-run] Installing platforms\android\app\build\outputs\apk\debug\app-debug.apk...
[native-run] Starting application activity io.ionic.starter/io.ionic.starter.MainActivity...
[native-run] Run Successful
```

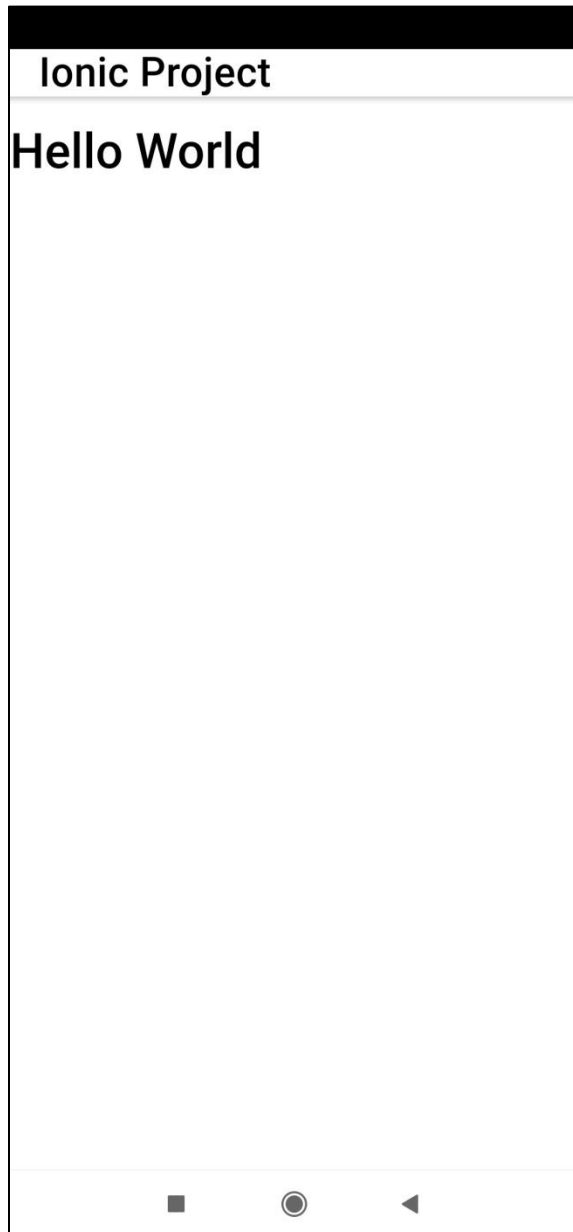
NAME: SACHI VIRESH SHAH

NEW ROLL NO: 381

CLASS: SYIT

OLD ROLL NO: 3066

Output on Android device:



PRACTICAL 7

AIM: Ionic Create, Generate and Add Pages

Theory:

1. Creating a new folder.

```
C:\Users\Dhwani>mkdir angular_ionic  
C:\Users\Dhwani>cd angular_ionic  
C:\Users\Dhwani\angular_ionic>
```

2. Starting a New Ionic App

- To start a new app, open your terminal/command prompt and run:

```
ionic start MyIonicProject tutorial --type=ionic-angular
```

- A. **start** will tell the CLI create a new app.
- B. **MyIonicProject** will be the directory name and the app name from your project.
- C. **tutorial** will be the starter template for your project.

3. Along with creating your project, this will also install node modules for the application, and prompt you if you want Cordova set up.

NAME: SACHI VIRESH SHAH

NEW ROLL NO: 381

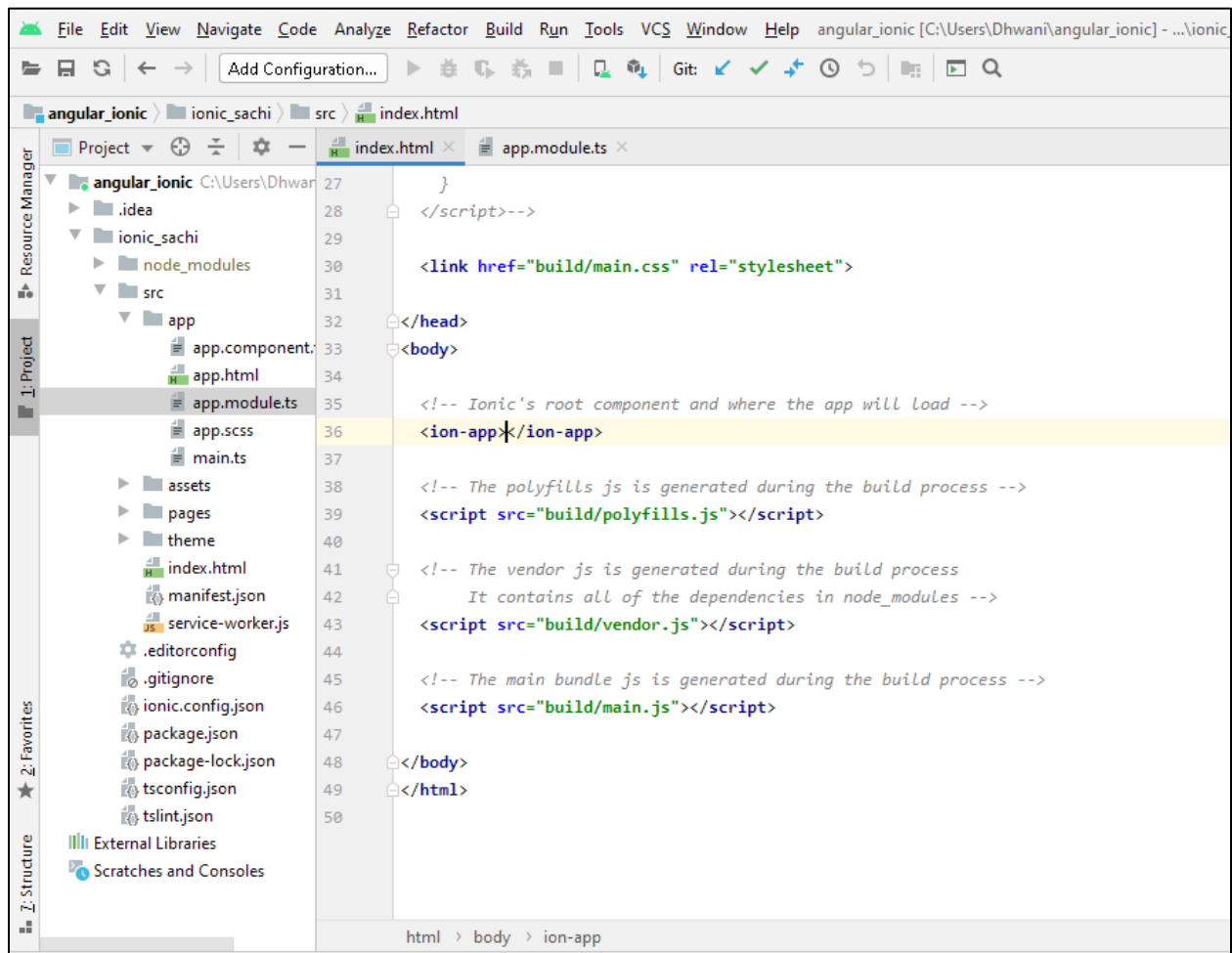
CLASS: SYIT

OLD ROLL NO: 3066

```
C:\Users\Dhwani\angular_ionic>
C:\Users\Dhwani\angular_ionic>ionic start ionic_sachi tutorial --type=ionic-angular
✓ Preparing directory .\ionic_sachi - done!
✓ Downloading and extracting tutorial starter - done!

Installing dependencies may take several minutes.
```

Code:



```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help angular_ionic [C:\Users\Dhwani\angular_ionic] - ...\ionic_
Add Configuration...
angular_ionic > ionic_sachi > src > index.html
Project
angular_ionic C:\Users\Dhwar
  .idea
  ionic_sachi
    node_modules
    src
      app
        app.component.ts
        app.html
        app.module.ts
        app.scss
        main.ts
      assets
      pages
      theme
      index.html
      manifest.json
      service-worker.js
      .editorconfig
      .gitignore
      ionic.config.json
      package.json
      package-lock.json
      tsconfig.json
      tslint.json
  External Libraries
  Scratches and Consoles
27      }
28      </script>-->
29
30      <link href="build/main.css" rel="stylesheet">
31
32    </head>
33    <body>
34
35      <!-- Ionic's root component and where the app will load -->
36      <ion-app></ion-app>
37
38      <!-- The polyfills js is generated during the build process -->
39      <script src="build/polyfills.js"></script>
40
41      <!-- The vendor js is generated during the build process
42           It contains all of the dependencies in node_modules -->
43      <script src="build/vendor.js"></script>
44
45      <!-- The main bundle js is generated during the build process -->
46      <script src="build/main.js"></script>
47
48    </body>
49  </html>
50
html > body > ion-app
```

NAME: SACHI VIRESH SHAH

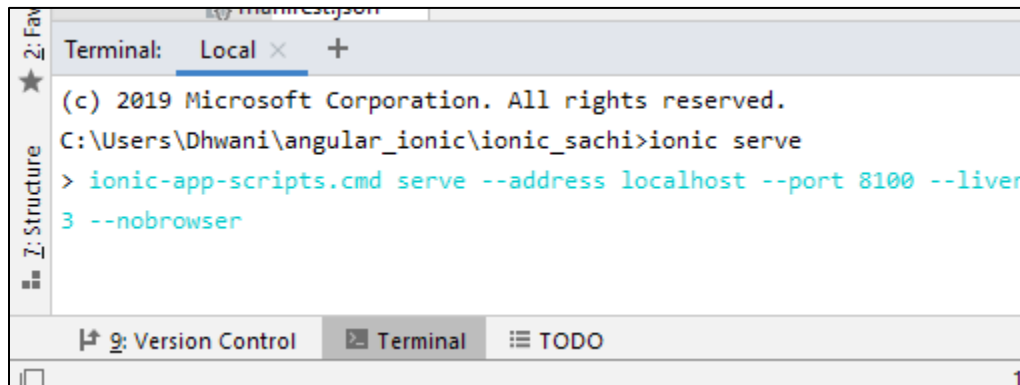
NEW ROLL NO: 381

CLASS: SYIT

OLD ROLL NO: 3066

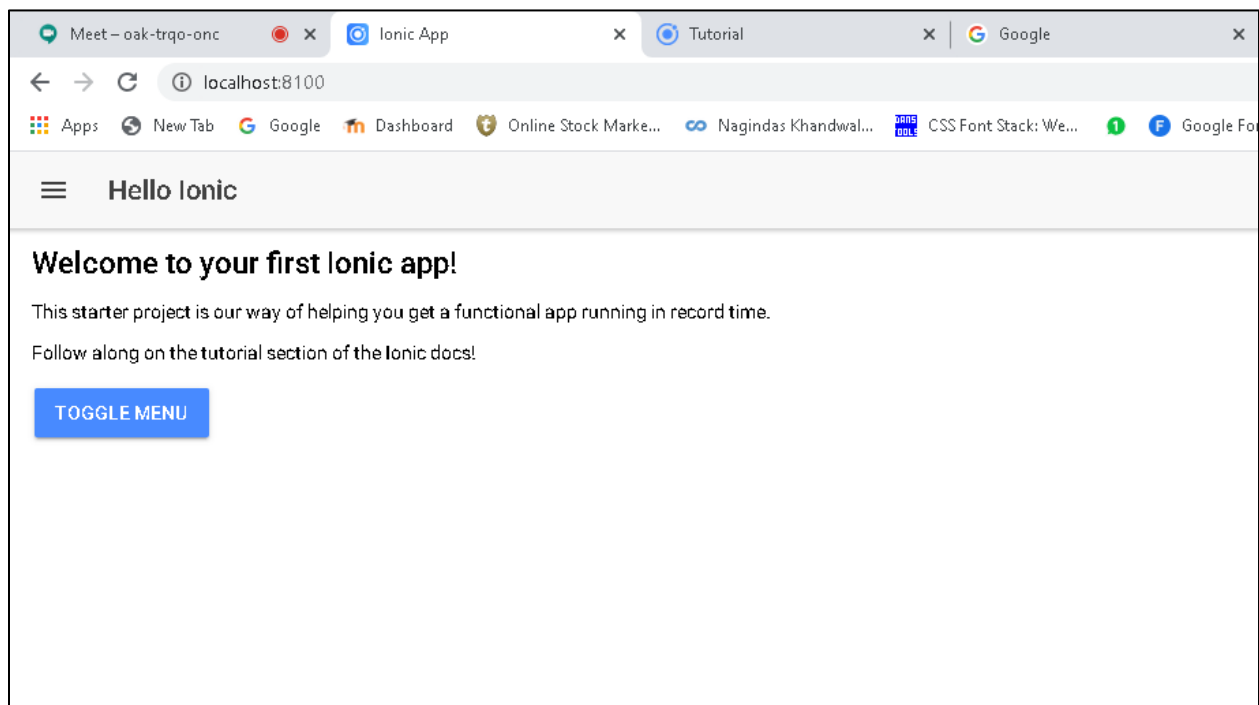
Viewing the app in a browser

Now, you can `cd` into the folder that was created. To get a quick preview of your app in the browser, use the `serve` command.



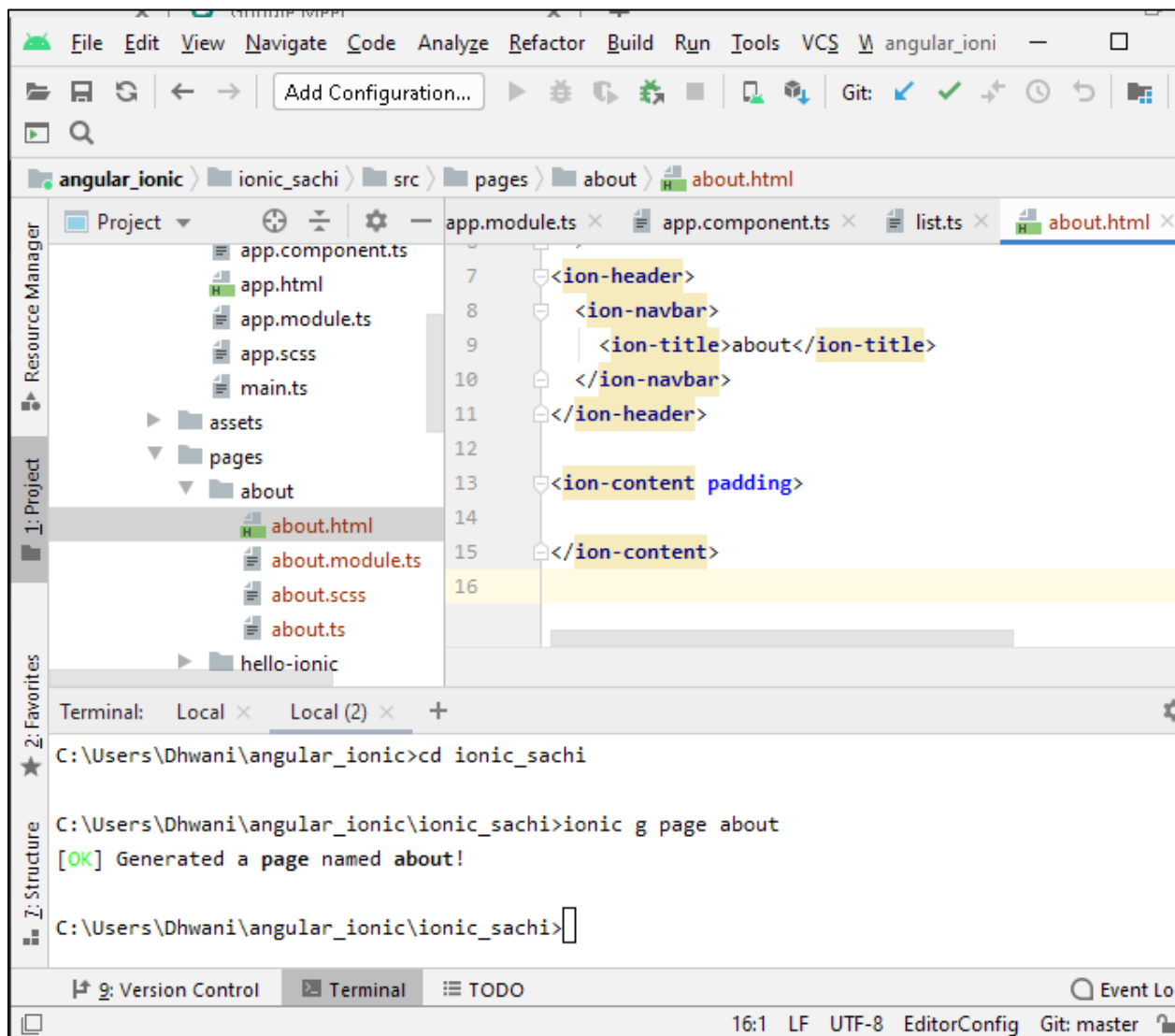
```
Terminal: Local x +
(c) 2019 Microsoft Corporation. All rights reserved.
C:\Users\Dhwani\angular_ionic\ionic_sachi>ionic serve
> ionic-app-scripts.cmd serve --address localhost --port 8100 --livereload
3 --nobrowser
```

Output:



Creating a new page:

- **ionic generate <type> <name> [options]**
- This command uses the Angular CLI to generate features such as pages, components, directives, services, etc.

Command:**ionic g page about**

NAME: SACHI VIRESH SHAH

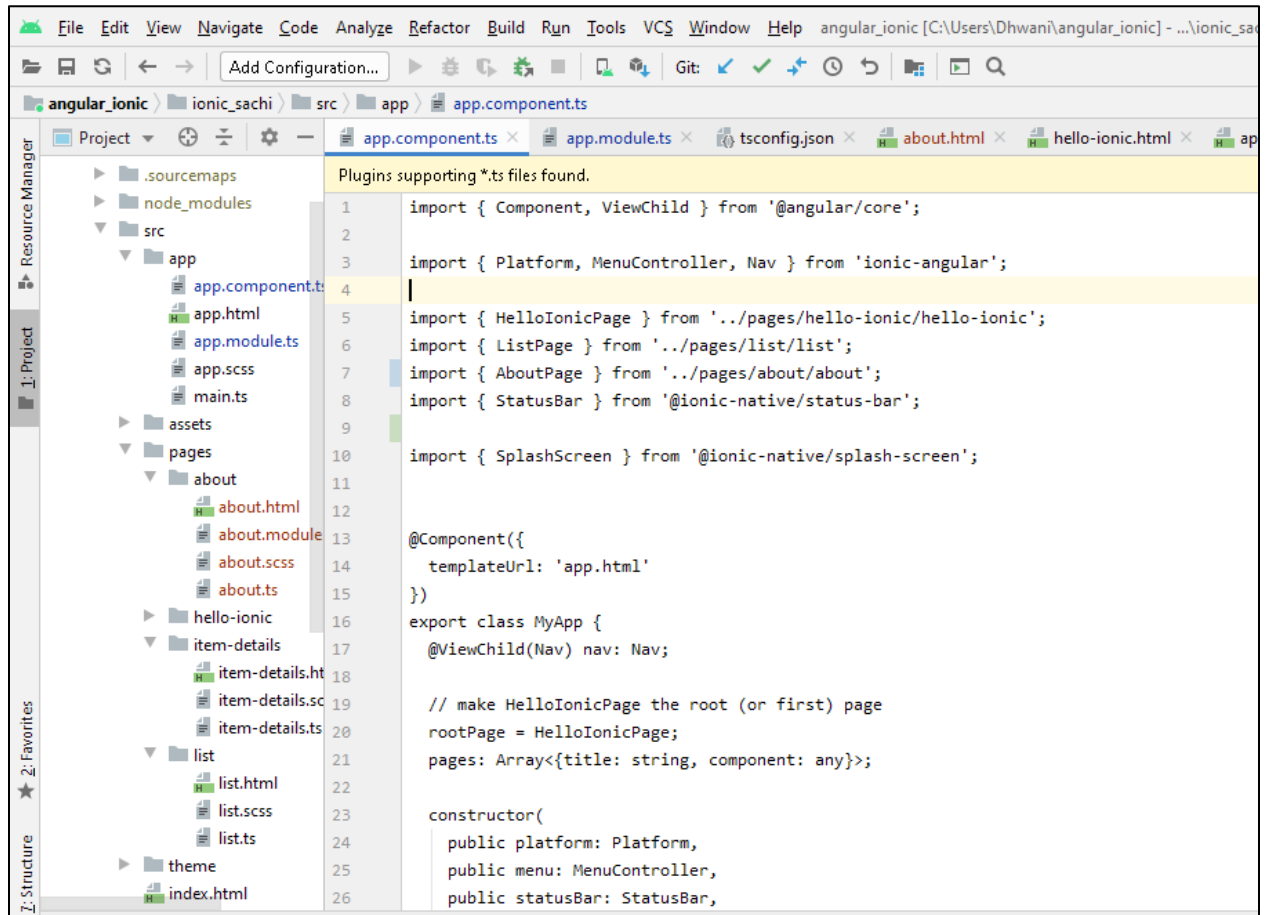
NEW ROLL NO: 381

CLASS: SYIT

OLD ROLL NO: 3066

Code:

import {AboutPage} from '../about/about';



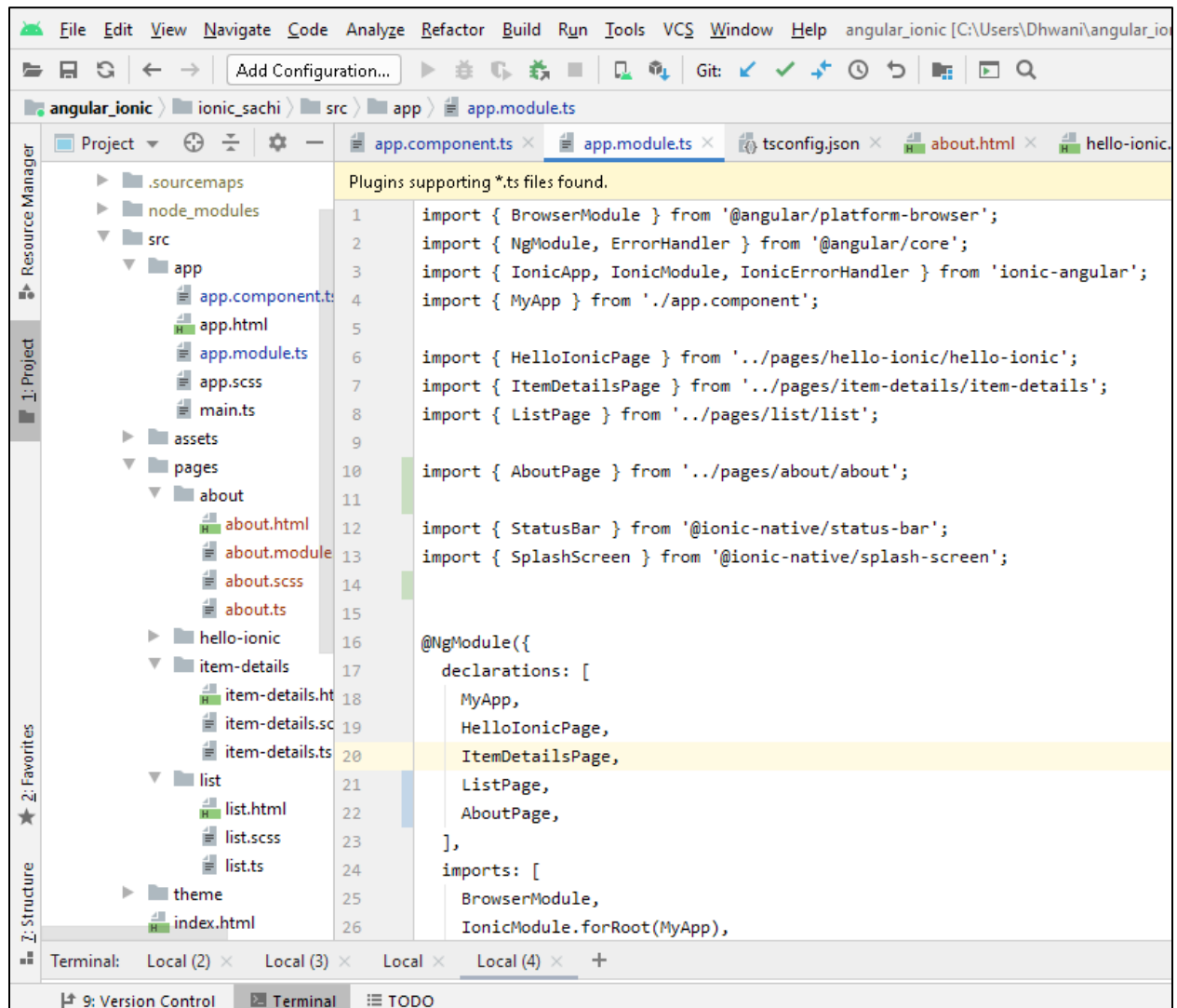
```
1 import { Component, ViewChild } from '@angular/core';
2
3 import { Platform, MenuController, Nav } from 'ionic-angular';
4
5 import { HelloIonicPage } from '../pages/hello-ionic/hello-ionic';
6 import { ListPage } from '../pages/list/list';
7 import { AboutPage } from '../pages/about/about';
8 import { StatusBar } from '@ionic-native/status-bar';
9
10 import { SplashScreen } from '@ionic-native/splash-screen';
11
12 @Component({
13   templateUrl: 'app.html'
14 })
15 export class MyApp {
16   @ViewChild(Nav) nav: Nav;
17
18   // make HelloIonicPage the root (or first) page
19   rootPage = HelloIonicPage;
20   pages: Array<{title: string, component: any}>;
21
22   constructor(
23     public platform: Platform,
24     public menu: MenuController,
25     public statusBar: StatusBar,
```

NAME: SACHI VIRESH SHAH

NEW ROLL NO: 381

CLASS: SYIT

OLD ROLL NO: 3066



```
C:\Users\Dhwani\angular_ionic\ionic_sachi>ionic serve
> ionic-app-scripts.cmd serve --address localhost --port 8103 --livereload-port 35732 --dev-logger-port
[app-scripts] [15:59:25] ionic-app-scripts 3.2.4
[app-scripts] [15:59:25] watch started ...
[app-scripts] [15:59:25] build dev started ...
[app-scripts] [15:59:25] clean started ...
[app-scripts] [15:59:25] clean finished in 145 ms
[app-scripts] [15:59:25] copy started ...
[app-scripts] [15:59:26] deeplinks started ...
[app-scripts] [15:59:26] deeplinks finished in 31 ms
[app-scripts] [15:59:26] transpile started ...
```

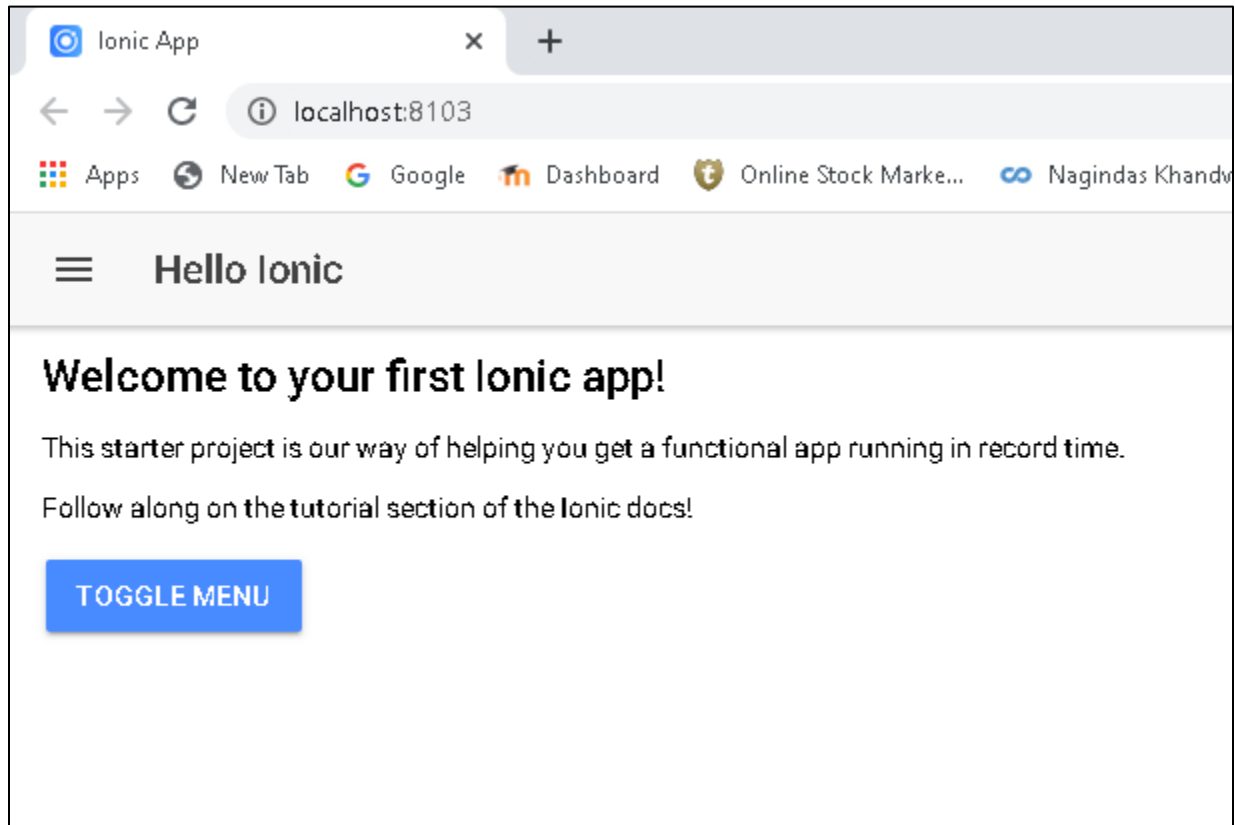
NAME: SACHI VIRESH SHAH

NEW ROLL NO: 381

CLASS: SYIT

OLD ROLL NO: 3066

Output:

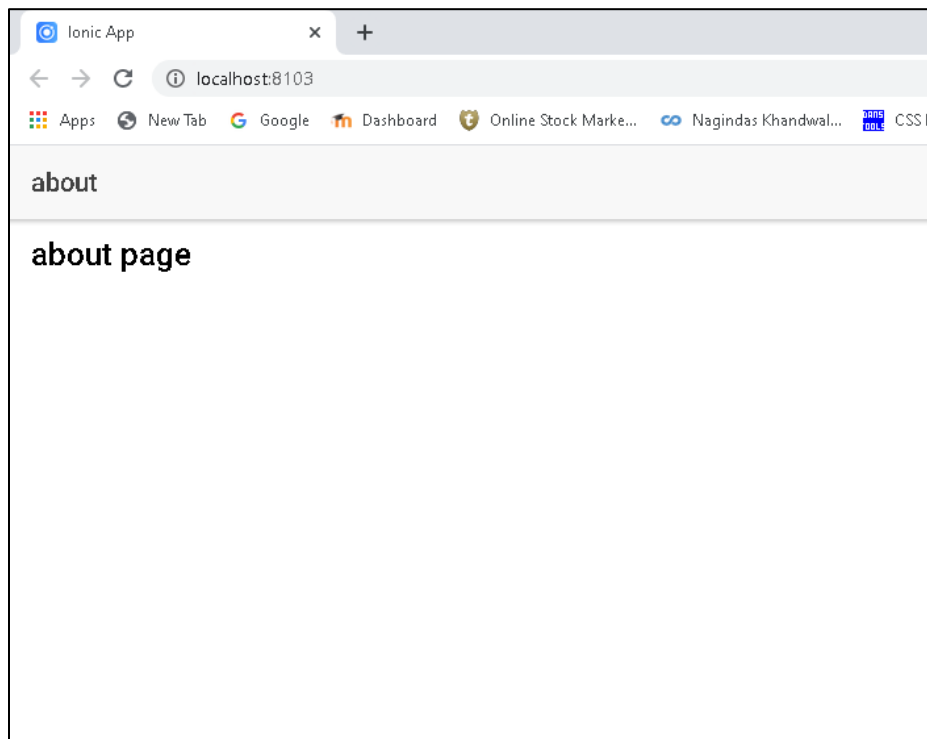
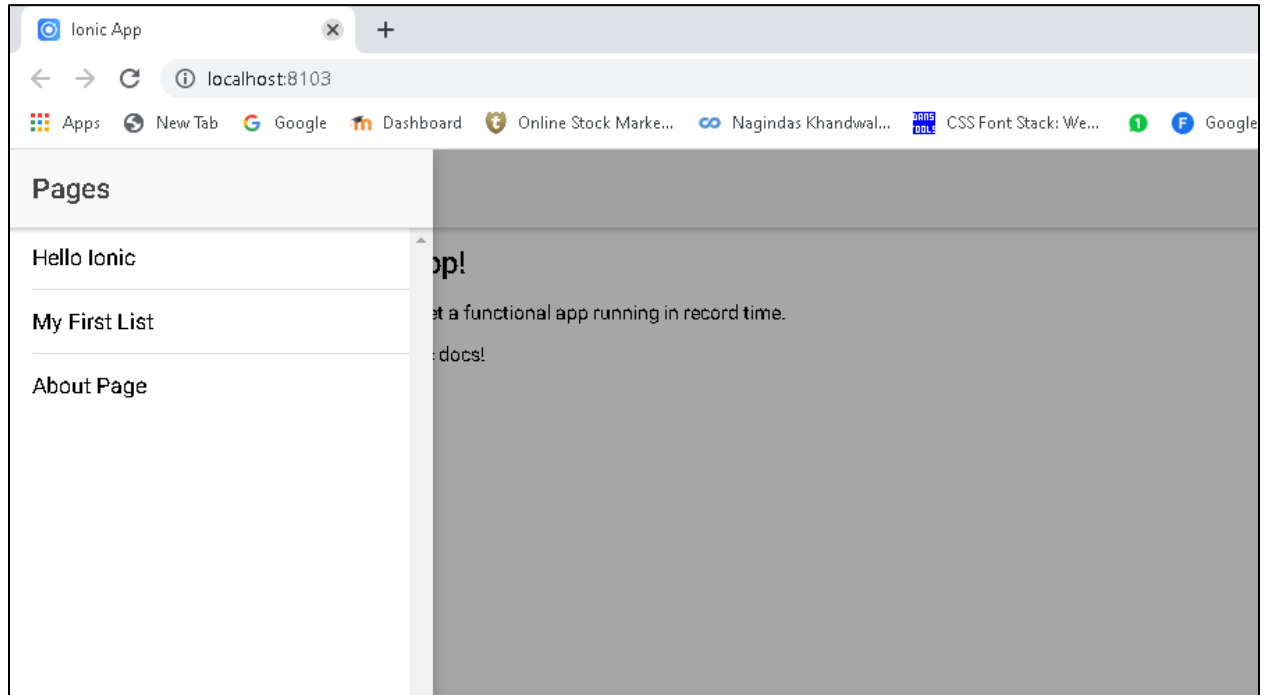


NAME: SACHI VIRESH SHAH

NEW ROLL NO: 381

CLASS: SYIT

OLD ROLL NO: 3066



Practical 8

Aim: Ionic Use Tabs starter command template and options

Theory:

Angular templates are dynamic. When **Angular** renders them, it transforms the DOM according to the instructions given by directives

Along with the tutorial template, Ionic also provides the following official templates:

- tabs : a simple 3 tab layout
- sidemenu: a layout with a swipable menu on the side
- blank: a bare starter with a single page
- super: starter project with over 14 ready to use page designs
- tutorial: a guided starter project

If you don't specify a template at the start, you will be prompted to pick one.

Creating a project with **tabs** template

```
npm
Microsoft Windows [Version 10.0.18362.1016]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Dhwani>ionic start ionic_proj 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 .\ionic_proj - done!
✓ Downloading and extracting tabs starter - done!

Installing dependencies may take several minutes.

-----

Ionic Studio, a powerful, local editor made with love by Ionic

    Lightning fast app creation
    Quickest & easiest way to get started with Ionic

Learn more: https://ion.link/studio

-----
```

Code for Tab1:

```
<? tab1.page.html X <? tab2.page.html TS tab1.page.ts
src > app > tab1 > <? tab1.page.html > ion-content > ion-header
1 <ion-header [translucent]="true">
2   <ion-toolbar>
3     <ion-title>
4       Tab 1
5     </ion-title>
6   </ion-toolbar>
7 </ion-header>
8
9 <ion-content [fullscreen]="true">
10   <ion-header collapse="condense">
11     <ion-toolbar>
12       <ion-title size="large">Tab 1</ion-title>
13     </ion-toolbar>
14   </ion-header>
15
16   <ion-button>Default</ion-button>
17   <ion-button (click)='changeTab()' color="dark">Dark</ion-button>
18   <ion-button expand="full" fill="outline">Outline + Full</ion-button>
19
20   <ion-list>
21     <ion-item>
22       <ion-range color="danger" pin="true"></ion-range>
23     </ion-item>
24   </ion-list>
25
26   <ion-card>
27     <ion-card-header>
28       <ion-card-subtitle>Card Subtitle</ion-card-subtitle>
29       <ion-card-title>Card Title</ion-card-title>
30     </ion-card-header>
31
32     <ion-card-content>
33       Keep close to Nature's heart... and break clear away, once in awhile,
34       and climb a mountain or spend a week in the woods. Wash your spirit clean.
35     </ion-card-content>
36   </ion-card>
37
38   <ion-list>
39     <ion-item>
40       <ion-label>Input</ion-label>
41     </ion-item>
42   </ion-list>
```

NAME: SACHI VIRESH SHAH

NEW ROLL NO: 381

CLASS: SYIT

OLD ROLL NO: 3066

```
<ion-list>
  <ion-item>
    <ion-label>Input</ion-label>
    <ion-input></ion-input>
  </ion-item>
  <ion-item>
    <ion-label>Toggle</ion-label>
    <ion-toggle slot="end"></ion-toggle>
  </ion-item>
  <ion-item>
    <ion-radio-group value="biff">
      <ion-list-header>
        <ion-label>Name</ion-label>
      </ion-list-header>

      <ion-item>
        <ion-label>Biff</ion-label>
        <ion-radio slot="start" value="biff"></ion-radio>
      </ion-item>

      <ion-item>
        <ion-label>Griff</ion-label>
        <ion-radio slot="start" value="griff"></ion-radio>
      </ion-item>

      <ion-item>
        <ion-label>Buford</ion-label>
        <ion-radio slot="start" value="buford"></ion-radio>
      </ion-item>
    </ion-radio-group>
  </ion-item>
  <ion-item>
    <ion-label>Checkbox</ion-label>
    <ion-checkbox slot="start"></ion-checkbox>
  </ion-item>
</ion-list>
</ion-content>
```

NAME: SACHI VIRESH SHAH

NEW ROLL NO: 381

CLASS: SYIT

OLD ROLL NO: 3066

Output for Tab1:

Tab 1

DEFAULT DARK

OUTLINE + FULL

Card Subtitle
Card Title
Keep close to Nature's heart... and break clear away, once in awhile, and climb a mountain or spend a week in the woods. Wash your spirit clean.

Input

Toggle

Name

☒ Biff

☐ Griff

☐ Buford

☒ Checkbox

Tab 1

Tab 2

Tab 3

Code for Tab2:

```
src > app > tab2 > tab2.page.html > ion-content > ion-item > ion-input#myinput.ion-text-center
1  <ion-header>
2    <ion-toolbar color="dark">
3      <ion-title class="ion-text-center">To-Do App</ion-title>
4    </ion-toolbar>
5
6    <ion-toolbar>
7      <ion-title class="ion-text-center">MY LIST</ion-title>
8    </ion-toolbar>
9  </ion-header>
10
11 <ion-content [fullscreen]="true">
12   <ion-item>
13     <ion-input id="myinput" [(ngModel)]="todo" placeholder="Insert your todo here" type="text"
14     class="ion-text-center"></ion-input>
15     <ion-icon class="item-note" name="add" slot="end" (click)="add()"></ion-icon>
16   </ion-item>
17
18   <ion-list>
19     <ion-item *ngFor="let item of todos">
20       <ion-item>
21         {{item}}
22       </ion-item>
23
24       <ion-item side="right" slot="end">
25         <ion-button color="danger" (click)="delete(item)">
26           <ion-icon name="trash" ></ion-icon>
27         </ion-button>
28       </ion-item>
29     </ion-item>
30   </ion-list>
31 </ion-content>
```

NAME: SACHI VIRESH SHAH

NEW ROLL NO: 381

CLASS: SYIT

OLD ROLL NO: 3066

Output for Tab2:

