



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: Smit Dhruve

Roll No: 390

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: 390

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	

1-AngularJS Data Binding

- *Data-binding in AngularJS apps is the automatic synchronization of data between the model and view components.*
- *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.*
- *AngularJS templates work differently.*
- *First the template (which is the uncompiled HTML along with any additional markup or directives) is compiled on the browser.*
- *The compilation step produces a live view. Any changes to the view are immediately reflected in the model, and any changes in the model are propagated to the view.*
- *The model is the single-source-of-truth for the application state, greatly simplifying the programming model for the developer. You can think of the view as simply an instant projection of your model.*
- *Because the view is just a projection of the model, the controller is completely separated from the view and unaware of it. This makes testing a snap because it is easy to test your controller in isolation without the view and the related DOM/browser dependency.*

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">
  <p ng-bind="firstname"></p>
</div>

<script>
var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope) {
  $scope.firstname = "John";
  $scope.lastname = "Doe";
});
</script>

<p>Use the ng-bind directive to bind the innerHTML of an element to a property
in the data model.</p>

</body>
</html>
```

OUTPUT:

John

Use the ng-bind directive to bind the innerHTML of an element to a property in the data model.

2-AngularJS Directives

- AngularJS comes with a set of these directives built-in, like `ngBind`, `ngModel`, and `ngClass`. Much like you create controllers and services, you can create your own directives for AngularJS to use.
- When AngularJS bootstraps your application, the HTML compiler traverses the DOM matching directives against the DOM elements.
- Some of the directives are :
 - The `ng-app` directive initializes an AngularJS application.
 - The `ng-init` directive initializes application data.
 - The `ng-model` directive binds the value of HTML controls (input, select, text area) to application data.
 - The `ng-repeat` directive actually **clones HTML elements** once for each item in a collection.

CODE:

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

<div ng-app="" ng-init="firstName='John'">

<p>Input something in the input box:</p>
<p>Name: <input type="text" ng-model="firstName"></p>
<p>You wrote: {{ firstName }}</p>

</div>

</body>
</html>
```

OUTPUT:

Input something in the input box:

Name:

You wrote: John

3- AngularJS Controllers

- In AngularJS, a Controller is defined by a JavaScript **constructor function** that is used to augment the AngularJS Scope.
- Controllers can be attached to the DOM in different ways. For each of them, AngularJS will instantiate a new Controller object, using the specified Controller's constructor function:
 - a) The `ngController` directive. A new child scope will be created and made available as an injectable parameter to the Controller's constructor function as `$scope`.
 - b) A route controller in a `$route` definition.
 - c) The controller of a regular directive, or a component directive.

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="personCtrl">

First Name: <input type="text" ng-model="firstName"><br>
Last Name: <input type="text" ng-model="lastName"><br>
<br>
Full Name: {{fullName()}}

</div>

<script>
var app = angular.module('myApp', []);
app.controller('personCtrl', function($scope) {
    $scope.firstName = "John";
    $scope.lastName = "Doe";
    $scope.fullName = function() {
        return $scope.firstName + " " + $scope.lastName;
    };
});
</script>
```

OUTPUT:

First Name:

Last Name:

Full Name: John Doe

4-AngularJS Events

- AngularJS includes certain directives which can be used to provide custom behavior on various DOM events, such as click, dblclick, mouseenter etc.
- You can add AngularJS event listeners to your HTML elements by using one or more of these directives:
 - ng-blur
 - ng-change
 - ng-click
 - ng-dblclick
 - ng-focus
 - ng-keydown
 - ng-keyup
 - ng-keypress
 - ng-mousedown
 - ng-mouseenter
 - ng-mouseleave
 - ng-mousemove
 - ng-mouseover
 - ng-mouseup

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">

<h1 ng-mousemove="count = count + 1">Mouse Over Me!</h1>

<h2>{{ count }}</h2>

</div>
<script>
var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope) {
    $scope.count = 0;
});
</script>

</body>
</html>
```

OUTPUT:

Mouse Over Me!

4

5-Ionic Create and Build First Project

STEPS TO CREATE AN HYBRID APP:-

- First you should install Node js and npm installed in your pc
- Then open command prompt and type the following commands :-
 - *npm install -g cordova:-*
 - This code will install the required attributes and packages for npm and cordova.
 - *ionic start helloworld blank*
 - This command will make a new project as helloworld with all the files and data from ionic .
 - *Change the directory to helloworld*
 - Cd helloworld
 - *Change the code in files in index.html*
 - Replace the code in index.html code with the following code....

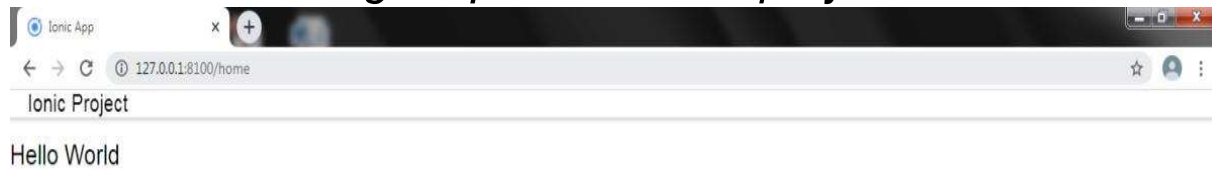
```
<ion-header>
  <ion-navbar>
    <ion-title>
      Ionic Project
    </ion-title>
  </ion-navbar>
</ion-header>
<ion-content padding>
  <h2>Hello World </h2>
</ion-content>
```

- *ionic serve*

- This step will execute your ionic code in browser in local host site .
- This step will give the output of the code.

- *open the app in 127.0.0.1:8100*

- *The following output will be displayed in browser*

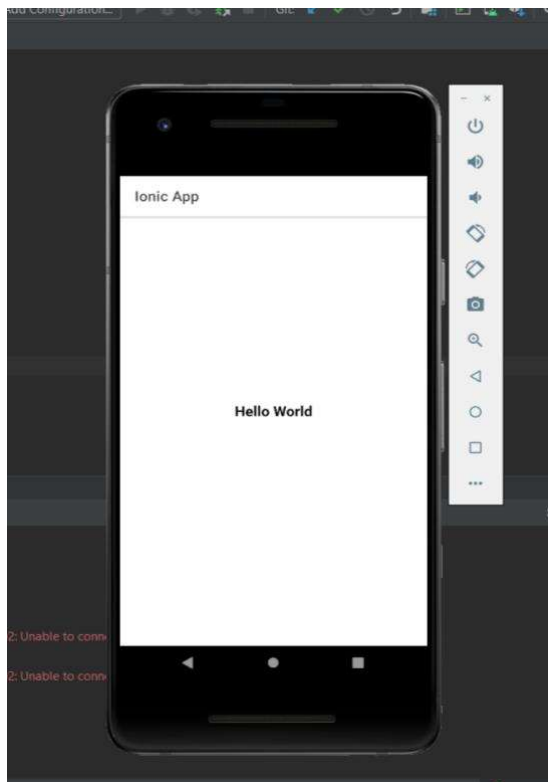


6-Ionic Adding Cordova Android Platform

Steps to create an ionic android app:

1. After you have built the web app next command you should type is `ionic cordova platform add android`. This will add an 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.

OUTPUT:



7-Ionic Create, Generate and Add Pages

- Ionic generate uses the Angular CLI to generate features such as pages, components, directives, services, etc.
- For a full list of available types, use `npx ng g --help`
- For a list of options for a types, use `npx ng g <type> --help`

You can specify a path to nest your feature within any number of subdirectories. For example, specify a name of "pages/New Page" to generate page files at `src/app/pages/new-page/`.

- Examples:
 - `ionic generate`
 - `ionic generate page`
 - `ionic generate page contact`
 - `ionic generate component contact/form`
 - `ionic generate component login-form --change-detection=OnPush`
 - `ionic generate directive ripple --skip-import`
 - `ionic generate service api/user`

OUTPUT:

```
C:\Users\Smit\helloworld>ionic generate page contact
> ng.cmd generate page contact
CREATE src/app/contact/contact-routing.module.ts (351 bytes)
CREATE src/app/contact/contact.module.ts (479 bytes)
CREATE src/app/contact/contact.page.html (126 bytes)
CREATE src/app/contact/contact.page.spec.ts (654 bytes)
CREATE src/app/contact/contact.page.ts (260 bytes)
CREATE src/app/contact/contact.page.scss (0 bytes)
UPDATE src/app/app-routing.module.ts (616 bytes)
[OK] Generated page!

C:\Users\Smit\helloworld>ionic generate component contact
> ng.cmd generate component contact
CREATE src/app/contact/contact.component.html (26 bytes)
CREATE src/app/contact/contact.component.spec.ts (689 bytes)
CREATE src/app/contact/contact.component.ts (272 bytes)
CREATE src/app/contact/contact.component.scss (0 bytes)
[OK] Generated component!

C:\Users\Smit\helloworld>
```

8-Ionic Use Tabs Starter Template

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

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

- `start` will tell the CLI create a new app.
- `MyIonicProject` will be the directory name and the app name from your project.
- `tabs` will be the starter template for your project.

OUTPUT:

