Practical-1: program to demonstrate Databinding

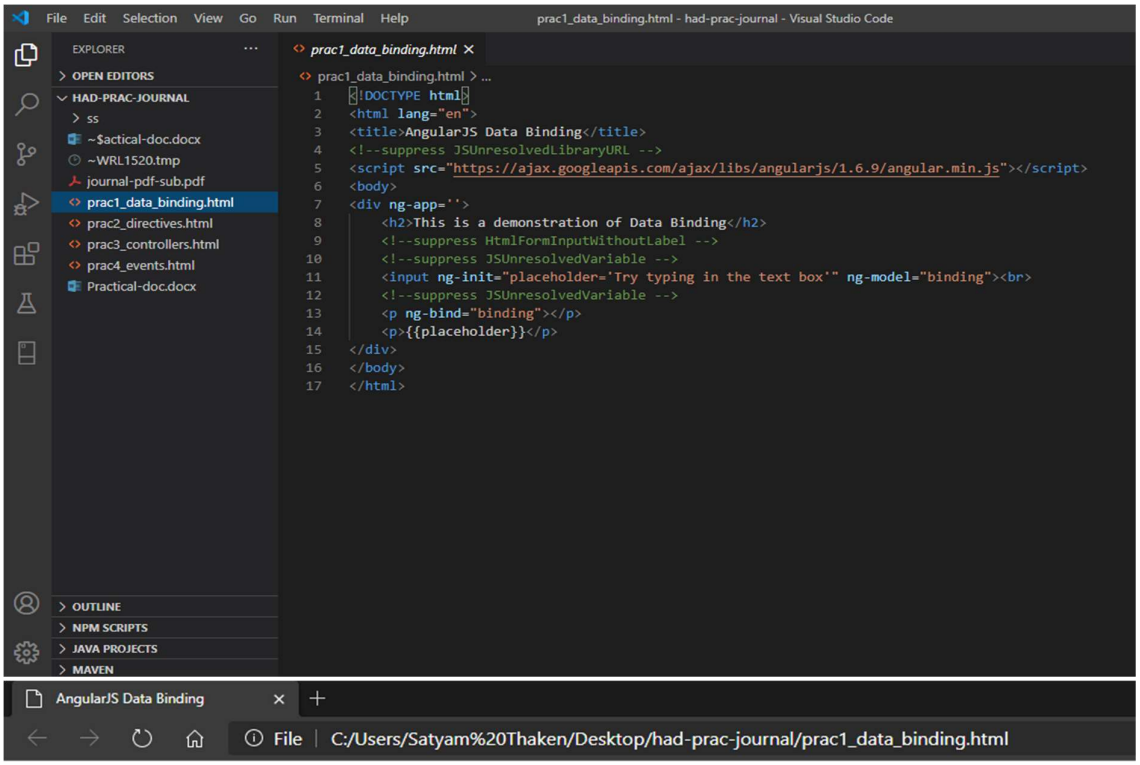
Theory

- 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.

Code and Output:

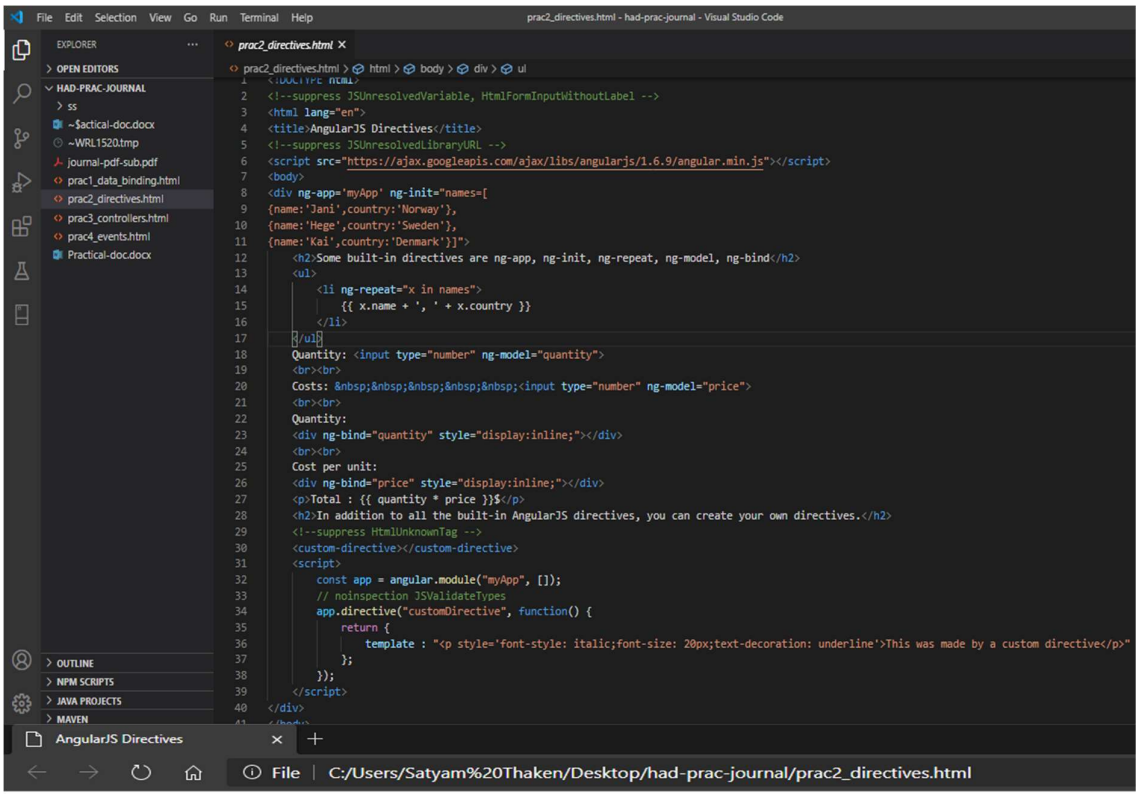


Practical-2: program to demonstrate Directives

Theory

- AngularJS directives are extended HTML attributes with the prefix `ng-`.
- 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, textarea) to application data.

Code and Output:



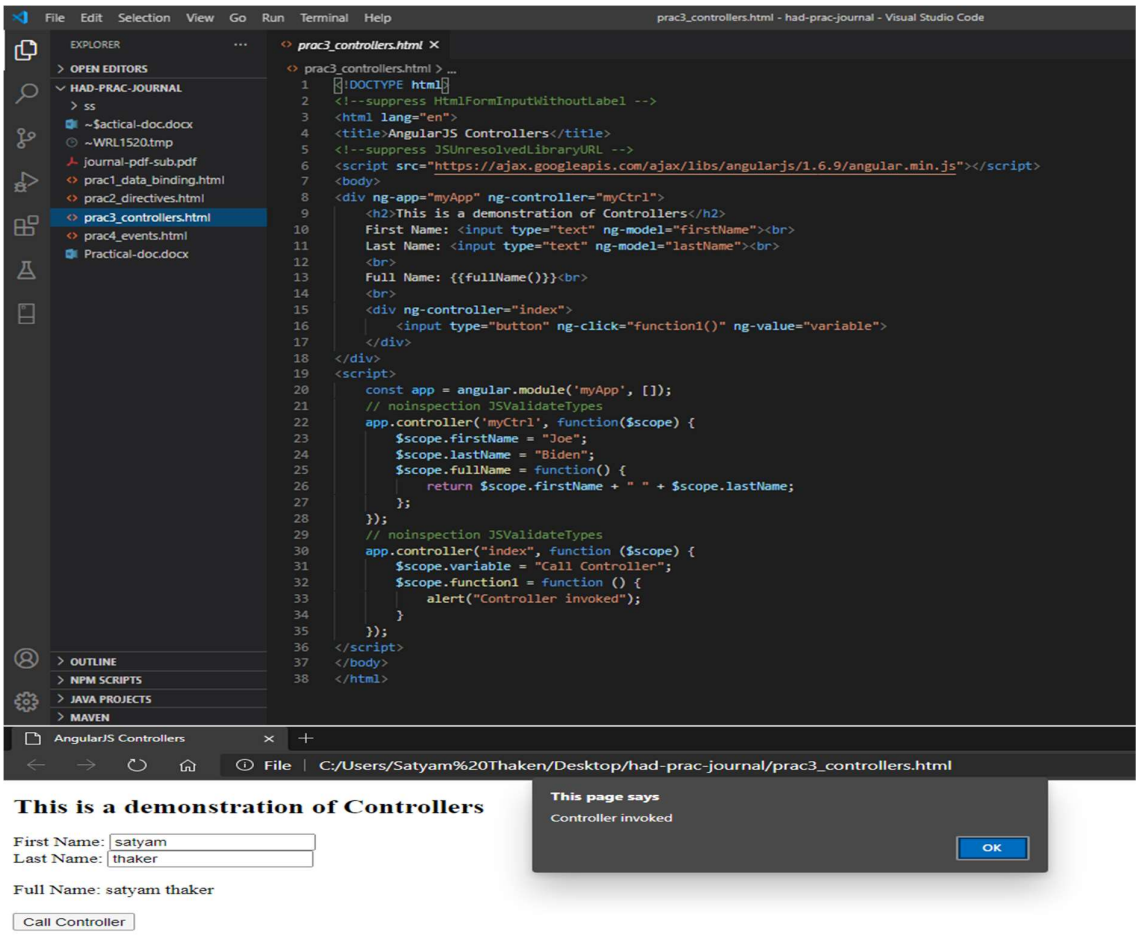
Practical-3: program to demonstrate controllers

Theory

- 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:
  - 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.
  - a route controller in a $route definition.
  - the controller of a regular directive, or a component directive.

Code and Output:

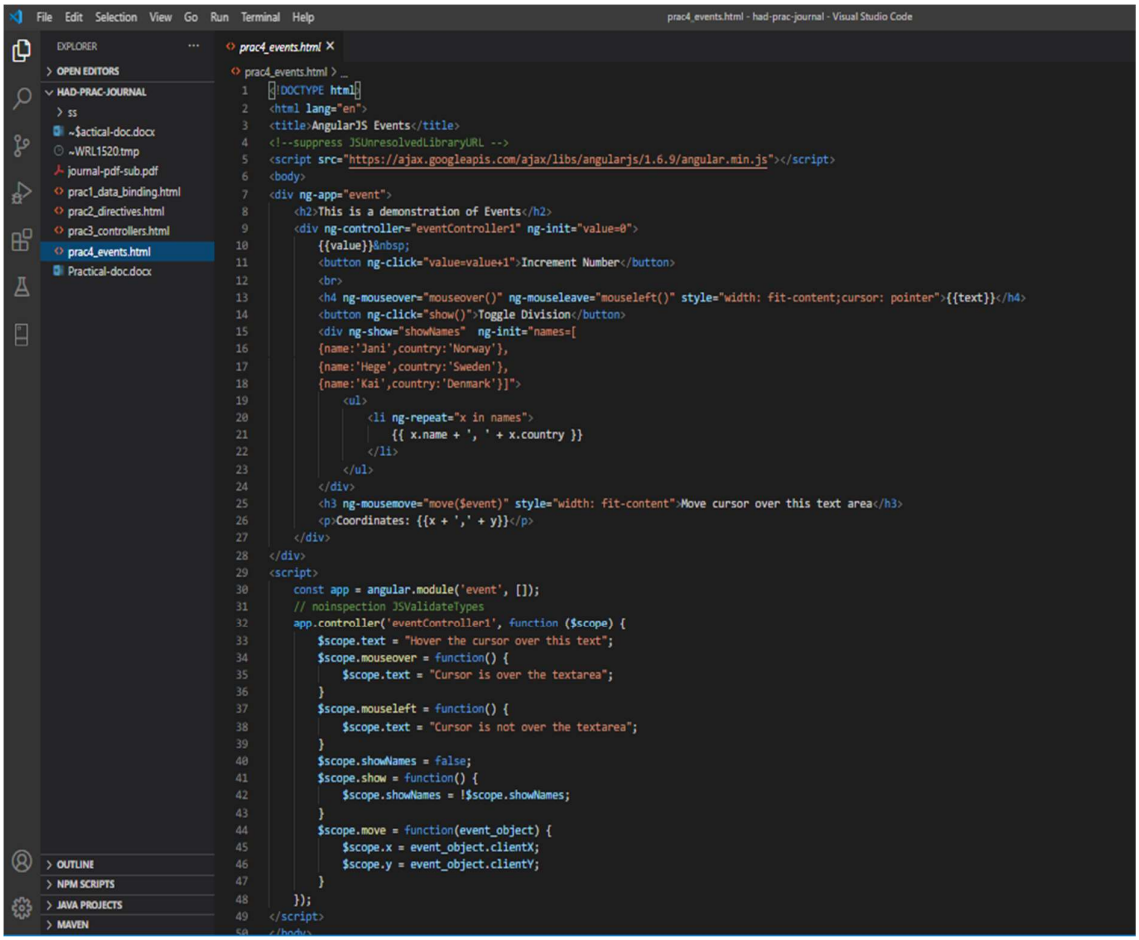


Practical-4: program to demonstrate Events

Theory

- 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.
- AngularJS includes certain directives which can be used to provide custom behavior on various DOM events, such as click, mouseover etc.
- You can pass the $event object as an argument when calling the function. The $event object contains the browser's event object.

Code and Output:



```
<!DOCTYPE html>
<html lang="en">
<title>AngularJS Events</title>
<!--suppress JSUnresolvedLibraryURL -->
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
<body>
<div ng-app="event">
    <h2>This is a demonstration of Events</h2>
    <div ng-controller="eventController1" ng-init="value=0">
        {{value}} 
        <button ng-click="value=value+1">Increment Number</button>
        <br>
        <h4 ng-mouseover="mouseover()" ng-mouseleave="mouseleft()" style="width: fit-content;cursor: pointer">{{text}}</h4>
        <button ng-click="show()">Toggle Division</button>
        <div ng-show="showNames"  ng-init="names=[
        {name:'Jani',country:'Norway'},
        {name:'Hege',country:'Sweden'},
        {name:'Kai',country:'Denmark'}]">
            <ul>
                <li ng-repeat="x in names">
                    {{ x.name + ', ' + x.country }}
                </li>
            </ul>
        </div>
        <h3 ng-mousemove="move($event)" style="width: fit-content">Move cursor over this text area</h3>
        <p>Coordinates: {{x + ',' + y}}</p>
    </div>
</div>
<script>
    const app = angular.module('event', []);
    // noinspection JSValidateTypes
    app.controller('eventController1', function ($scope) {
        $scope.text = "Hover the cursor over this text";
        $scope.mouseover = function() {
            $scope.text = "Cursor is over the textarea";
        }
        $scope.mouseleft = function() {
            $scope.text = "Cursor is not over the textarea";
        }
        $scope.showNames = false;
        $scope.show = function() {
            $scope.showNames = !$scope.showNames;
        }
        $scope.move = function(event_object) {
            $scope.x = event_object.clientX;
            $scope.y = event_object.clientY;
        }
    });
</script>
```

AngularJS Events

File | C:/Users/Satyam%20Thaken/Desktop/had-prac-journal/prac4_events.html

This is a demonstration of Events

6 Increment Number

Cursor is not over the textarea

Toggle Division

- Jani, Norway
- Hege, Sweden
- Kai, Denmark

Move cursor over this text area

Coordinates: 112,279

Practical 5: Ionic Create and Build First Project

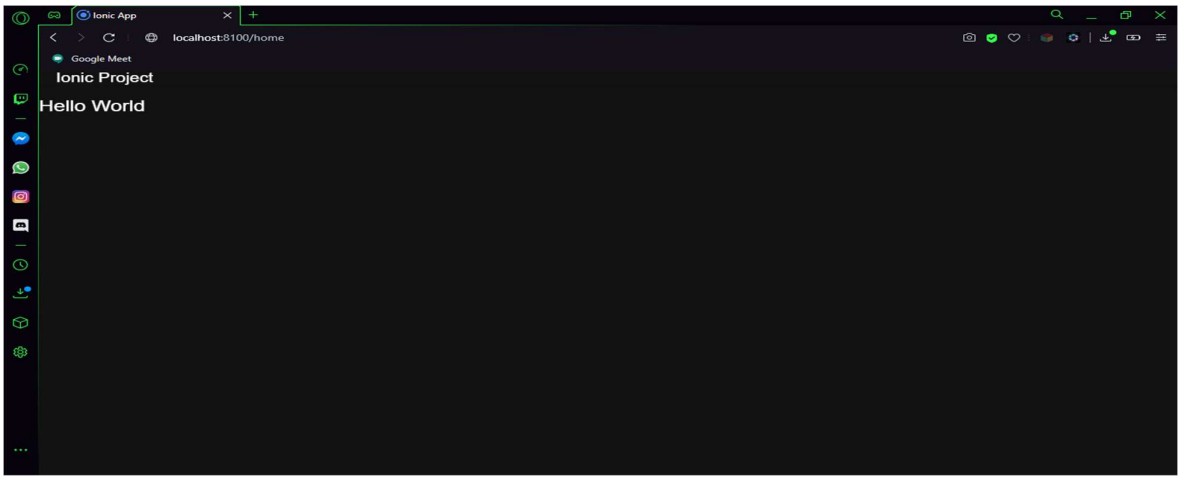
Theory

What is a hybrid app?

This solution is a blend, hence the name hybrid, of both native and web solutions. Where the core of the application is written using web technologies (HTML, CSS, and JavaScript), which are then encapsulated within a native application. Through the use of plugins, these applications can have full access to the mobile device's features. To better understand this approach, let's break down how it all fits together

Steps for creating Ionic Web app:

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>
4. Then it will ask to choose a framework if you want to make it in angular choose angular
5. Then to start running the ionic web page in the server type ionic serve.
6. To access the web page go on http://localhost:8100



Practical-6: Ionic Adding Cordova Android Platform

Theory

How to create and Ionic Android app:

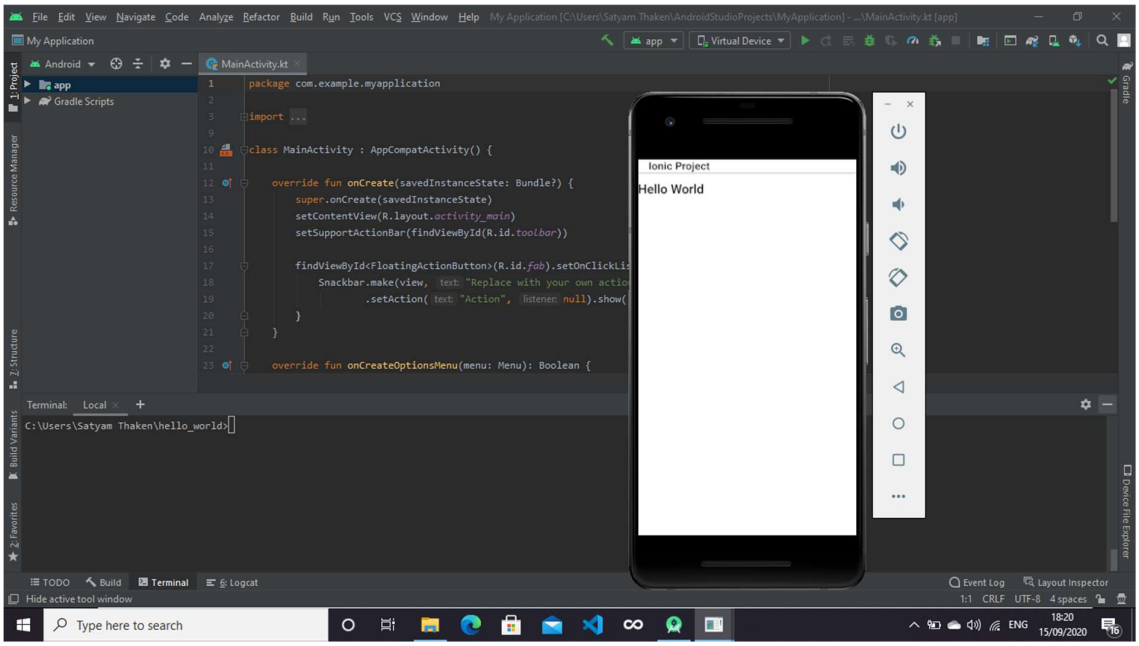
Steps to create an ionic android app:

- After you have built the web app next command you should type is ionic cordova platform add android. This will add and android platform to
  - your web app directory.

- Now type ionic cordova build android to start building your android app.

- Now type ionic cordova run android, if you have connected 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.

ANDROID OUTPUT:



Practical 7: Ionic Create, Generate and Add Pages

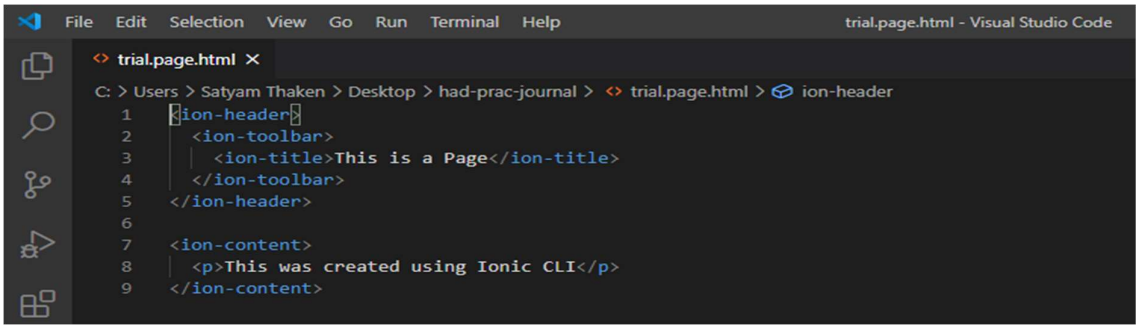
Theory

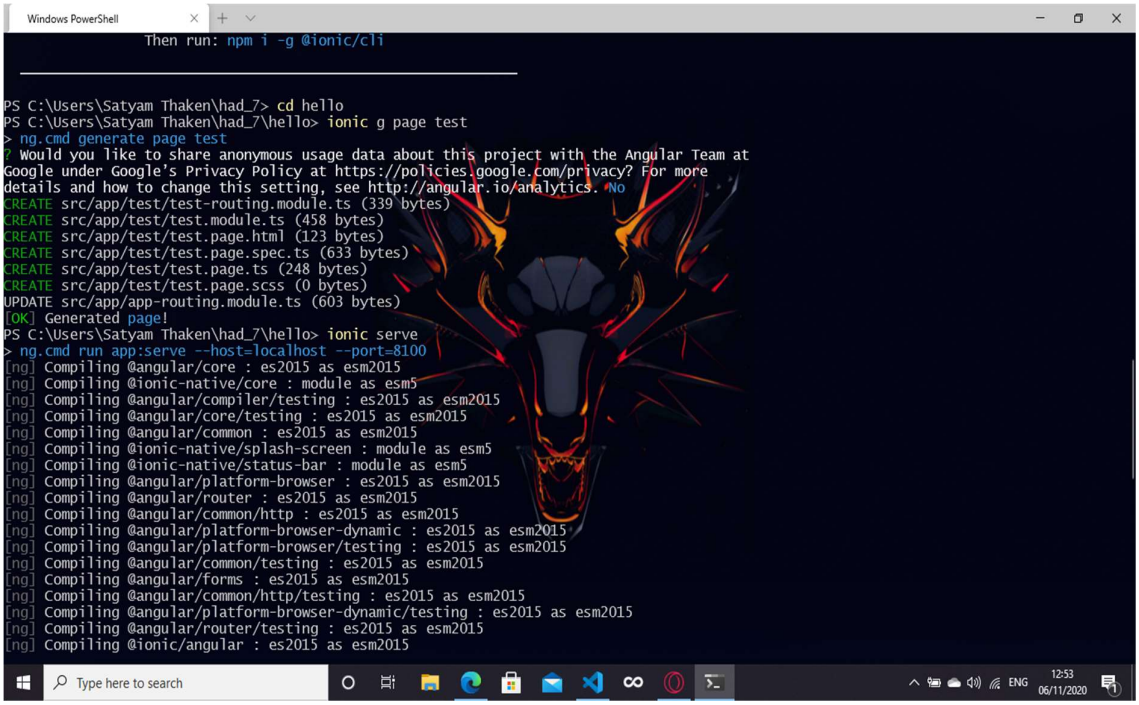
- An Ionic page is just an Angular component.
- The Ionic Page handles registering and displaying specific pages based on URLs. It's used underneath NavController so it will never have to be interacted with directly. When a new page is pushed with NavController, the URL is updated to match the path to this page.
- Unlike traditional web apps, URLs don't dictate navigation in Ionic apps. Instead, URLs help us link to specific pieces of content as a breadcrumb. The current URL gets updated as we navigate, but we use the NavController push and pop, or NavPush and NavPop to move around. This makes it much easier to handle complicated nested navigation.

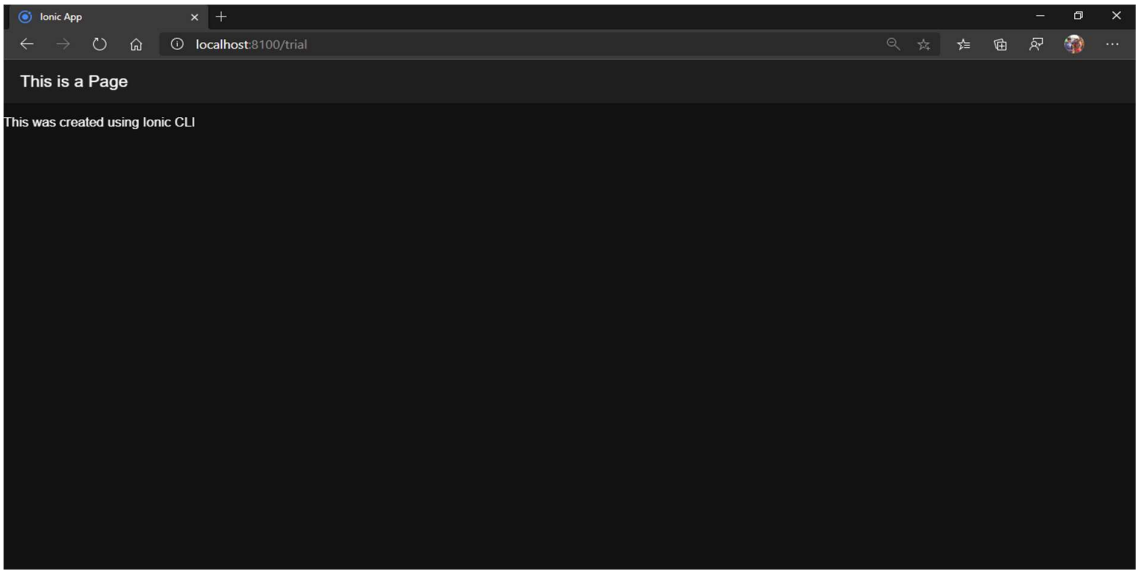
Steps:

1. Open command prompt as administrator
2. Create a folder for your ionic project by running md <folder_name>
3. Navigate to the folder by running cd <folder_name>
4. To create an ionic app run ionic start <app_name> blank
5. Then it will ask to choose a framework so choose `Angular JS`
6. Navigate to the folder by running cd <app_name>
7. To generate and add a page run ionic g page <page_name>
8. This will create a folder with all the components of your page in <pagename>
9. Then to start running the ionic web page on the server type ionic serve
10. To access the web page go on http://localhost:8100//<pagename>

Code, Commands and Output:







Practical 8: Ionic Use Tabs Starter Template

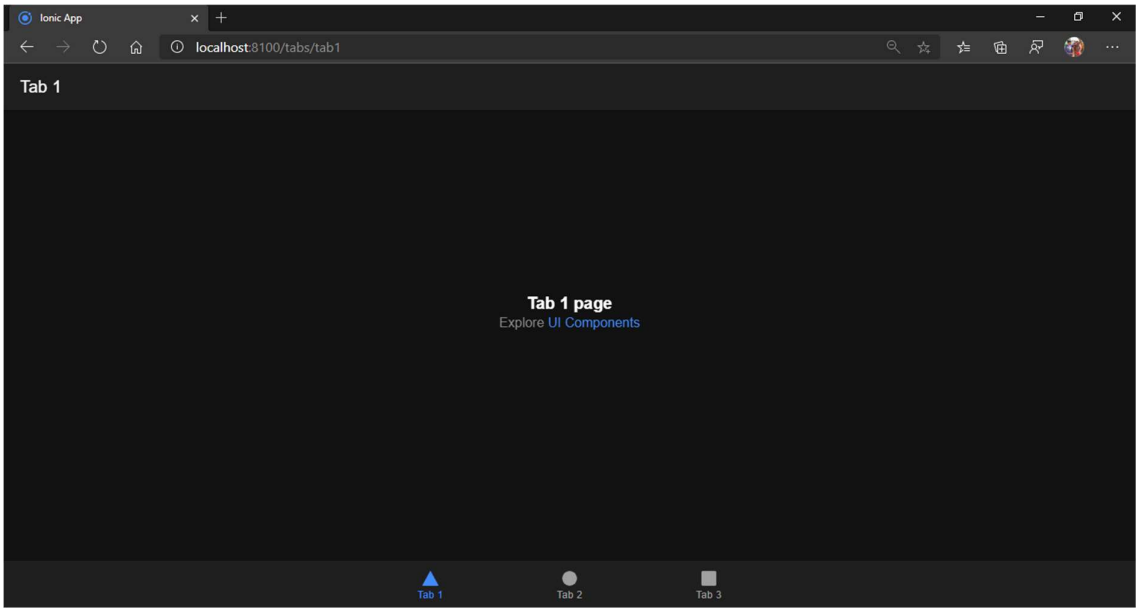
Theory

- Ionic Starter Templates are ready-to-go starter packs for your next Ionic app.
- The ionic starter templates are:
  - Tabs: A starting project for Ionic using a simple tabbed interface
  - Blank: A blank starter project for Ionic
  - Sidemenu: A starting project for Ionic using a side menu with navigation in the content area
  - Maps: An Ionic starter project using Google Maps and a side menu

Steps:

1. Open command prompt as administrator
2. Create a folder for your ionic project by running md <folder_name>
3. Navigate to the folder by running cd <folder_name>
4. To create an ionic app run ionic start <app_name> tabs
5. Then it will ask to choose a framework so choose Angular JS
6. Navigate to the folder by running cd <app_name>
7. Then to start running the ionic web page on the server type ionic serve
8. To access the web page go on http://localhost:8100

Output:


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