## **Practical – 1**

## **What is Hybrid App Development?**

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.

The heart of a hybrid-mobile application is still just an application that is written with HTML, CSS, and JavaScript. However, instead of the app being shown within the user’s browser, it is run from within a native application and its own embedded browser, which is essentially invisible to the user. For example, an iOS application would use the WKWebView to display our application, while on Android it would use the WebView element to do the same function.

This code is then embedded into a native application wrapper using a solution like Apache Cordova (also known as PhoneGap) or [Ionic’s Capacitor](https://capacitor.ionicframework.com/" \t "_blank). These solutions create a native shell application that is just the platform’s webview component in which it will load your web application. This gives you the ability to create and publish true native applications that can be submitted to each of the platform’s app stores for sale.

Additionally, both Cordova and Capacitor have a plugin system that allows you to extend beyond the limitations of the ‘browser’ and access the full suite of capabilities of a user’s mobile device. So, if you wanted to use TouchID on an iOS device as a login option, or wanted to connect to a Bluetooth device, this can be easily done by installing a plugin. These plugins are created by a wide range of developers and many are actively supported. Ionic even offers a complete ecosystem of supported plugins as part of its [Enterprise](https://ionicframework.com/enterprise) solution. So, the limitations of a web-only application are easily overcome, allowing your application to have parity with native applications in their features.

However, there are some drawbacks with this option. Similarly to the web-only application solution, the UI library has to be recreated. Here is where solutions like Ionic, NativeScript, Xamarin, React Native, and others step in. These options all provide robust UI components that look and feel like their native counterparts, giving you a full suite of building blocks for your application.

The only other consideration to take into account is if your application is still running within the device’s native browser. If so, you may encounter performance issues or other quirks specific to each platform or operating version.

## **How to create and ionic web app:**

The requirement for creating an Ionic web app is:

**Node.js** for interacting with the Ionic ecosystem [Download the LTS version here](https://nodejs.org/en/).

**A code editor** for... writing code! We are fans of [Visual Studio Code](https://code.visualstudio.com/).

**Command-line interface/terminal (CLI)**:

**Windows** users: for the best Ionic experience, we recommend the built-in command line (cmd) or the PowerShell CLI, running in Administrator mode.

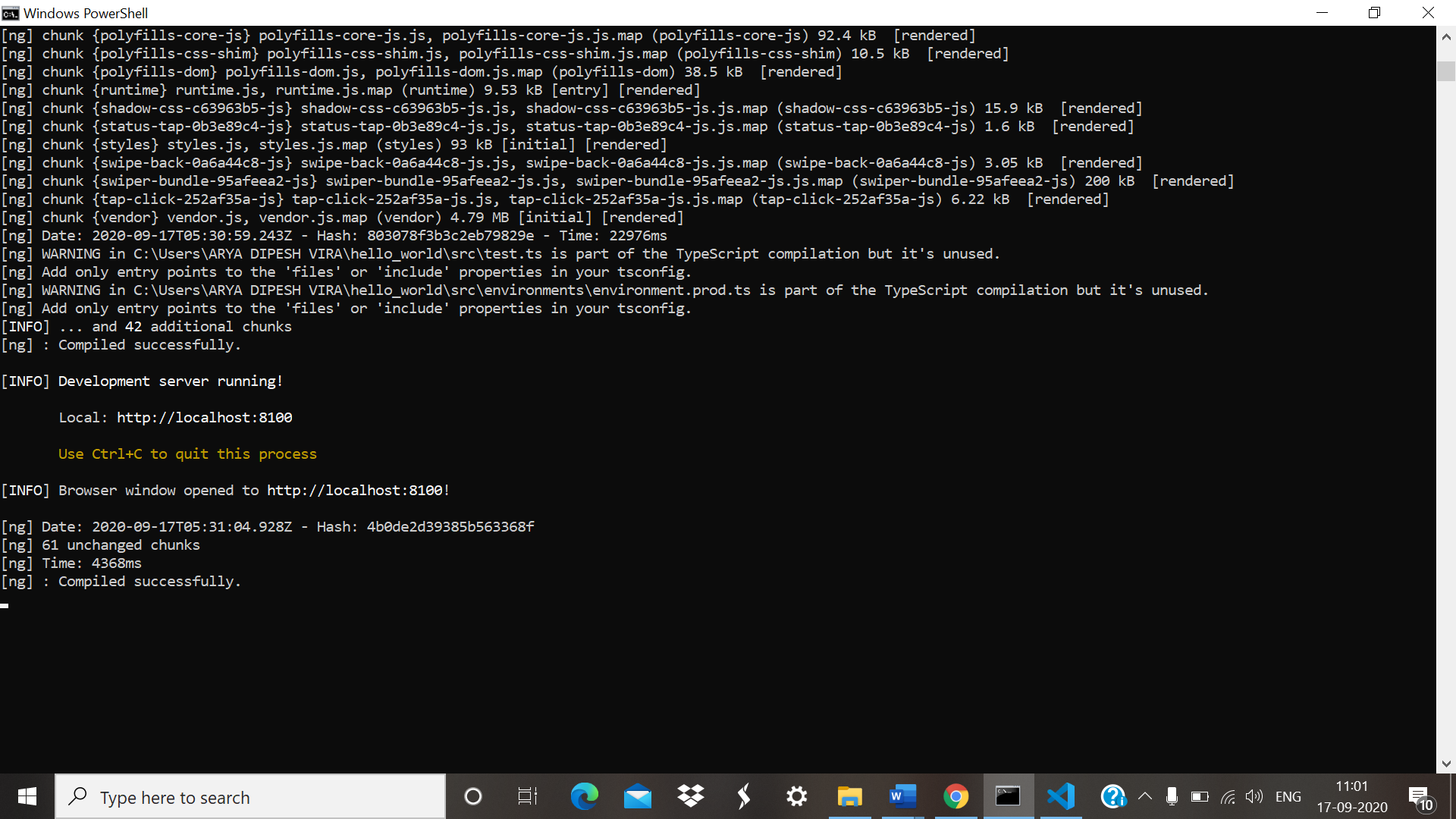
**Mac/Linux** users, virtually any terminal will work.

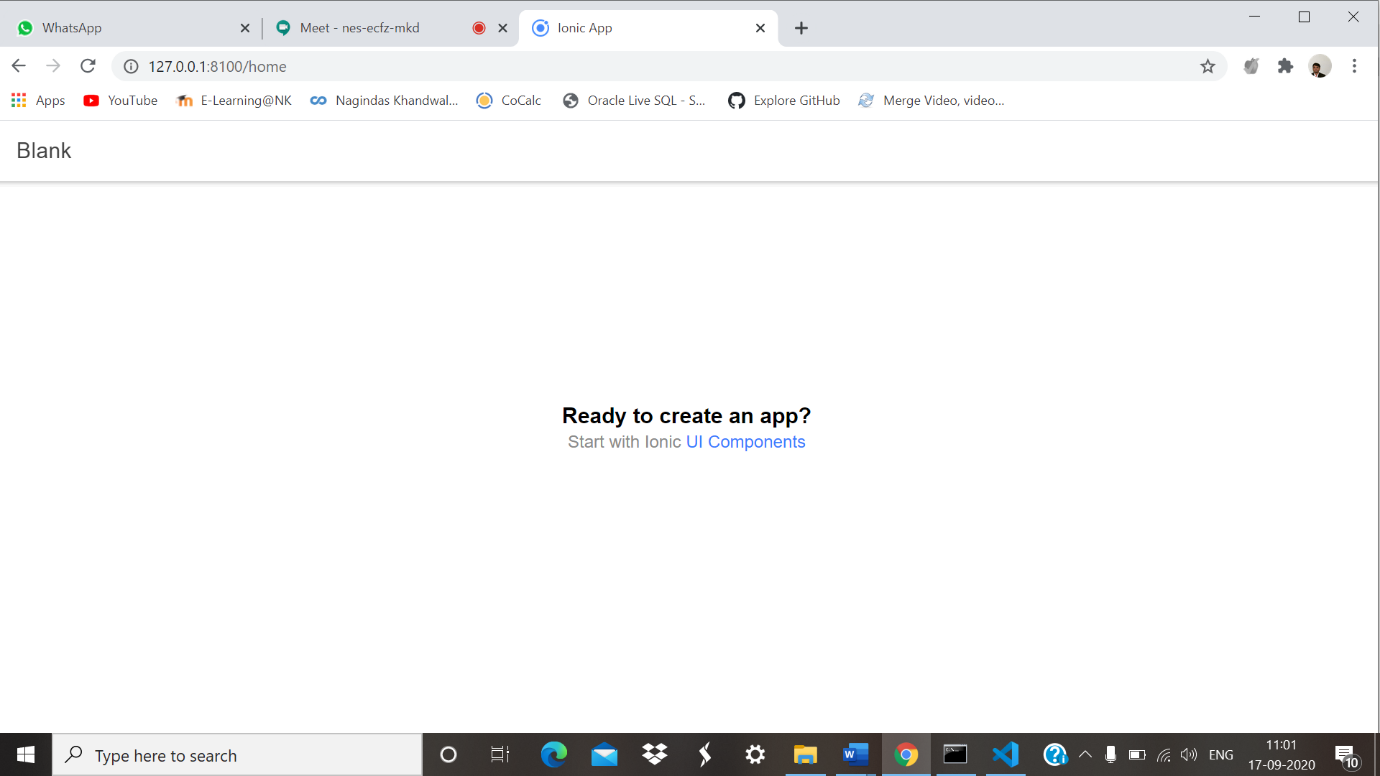
**To open a terminal in Visual Studio Code, go to Terminal -> New Terminal.**

1. $ npm install -g @**ionic**/cli native-run Cordova-res. The -g option means install globally.
2. $ **ionic** start dummy-project tabs
3. Then it will ask to choose a framework if you want to make it in angular choose angular
4. $ cd dummy-project
5. $ **ionic** serve.
6. . To access the web page go on <http://localhost:8100>

## **Steps to perform the practical:**

|  |
| --- |
| 1. Installing nodejs:   https://nodejs.org/en/ |
|  |
| 1. Checking nodejs is installed.   Go to your preferred path in cmd.  Type node --version  And  npm -- version  If both are installed. |
|  |
| 1. Installing angularjs   Type the below code in cmd where node and npm is installed.  “npm install -g @angular/cli” |
|  |
| 1. Installing ionic   Type the below code in cmd where node, angular and npm is installed.  “npm install -g @ionic/cli” |
|  |
| 1. To start an app in ionic.   Type in cmd in the same path.  “Ionic start hello world blank” |
|  |
| 1. Now a folder will be created in the same path   Change directory to that folder, by using.  “cd hello world” |
|  |
| 1. and then in the hello\_world folder and type   ionic serve  Now open,  “Chrome and type 127.0.0.1:8100”  And your app is created. |

****



## **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.

### 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.