Hybrid Mobile

Background and Introduction to Ionic

COS216
AVINASH SINGH
DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF PRETORIA

BACKGROUND

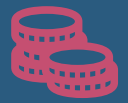
- What are Hybrid Apps?
  - Hybrid apps are a blend, hence the name hybrid, of both native and web solutions. Where the core of the application is written using web technologies.
  - Instead of coding directly in Android you use web technologies which are later converted to native applications (to an extent).
  - This allows you to code faster whilst minimizing the complexity of doing things natively.

ADVANTAGES

One source code base

Multiple platform

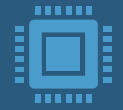
Uses web technologies



Cost effective

Quicker to market

Tons of support



Easier to code

DISADVANTAGES

PERFORMANCE

DEBUGGING

FEATURES

FRAMEWORKS

- Frameworks help simplify the process of making hybrid applications. They describe custom methods to perform certain functionality based on how the framework was designed.
- Each framework tends to exploit a certain programming language based on the functionalities and trade-offs.
- Most frameworks have large support and each have their own advantages and disadvantages.
- Some are simpler than others but it all depends on preferences and where your skills lie.

FLUTTER

- Developed by Google
- Open Source
- Supports: Android, IOS, Windows, Mac, Linux, Google Fuchsia
- Initial Release: May 2017
- Written in: Dart
- Flutter Engine is written in C++ to get low-level support and rendering
- https://flutter.dev/

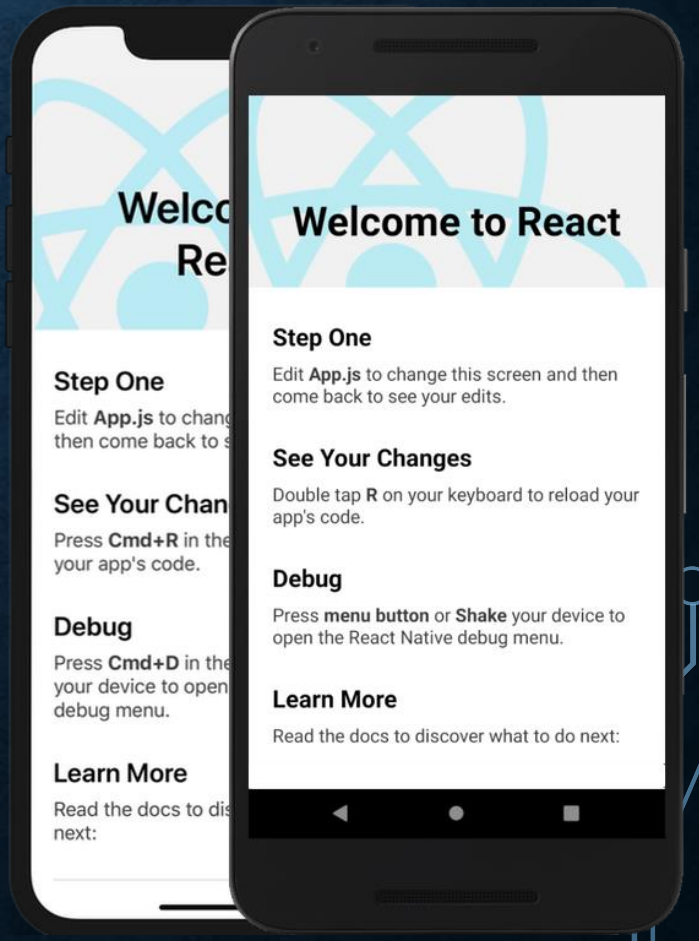
FLUTTER


```
import 'package:flutter/material.dart';
void main() => runApp(HelloWorldApp());

 class HelloWorldApp extends StatelessWidget {
    @override Widget build(BuildContext context)   {
    //MaterialApp acts as a wrapper to the app and //provides many features like
title,home,theme etc
    return MaterialApp(
        title: 'Hello World App', //Scaffold acts as a binder that binds the
            appBar, //bottom nav bar and other UI components at their places
        home: Scaffold( //AppBar() widget automatically creates a material app bar
        appBar: AppBar( title: Text('Hello World App'), ), //Center widget aligns
the child in center
        body: Center( child: Text('Hello World'), ), ), ); } }
```

REACT NATIVE

- Developed by Facebook
- Open Source
- Supports: Android, IOS, Web, UWP (Universal Windows Platform)
- Initial Release: March 2015
- Written in: React (JavaScript, C++, Objective-C, python)
- https://reactnative.dev/



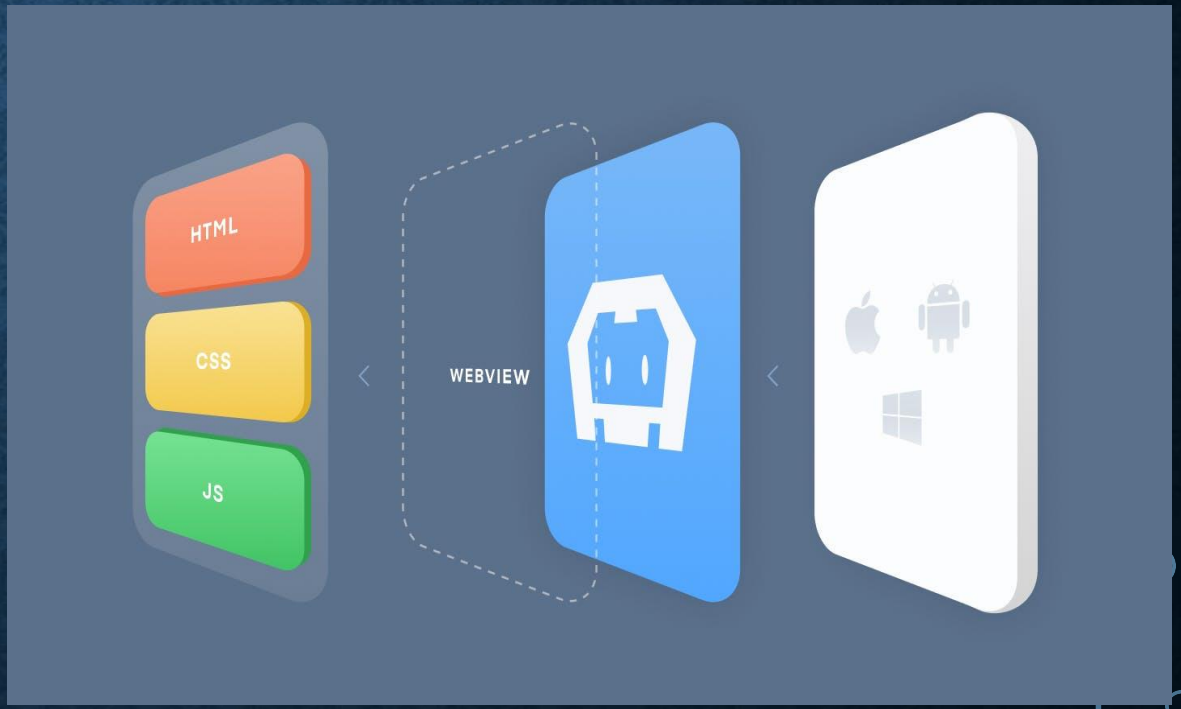
REACT NATIVE

```
import React from 'react';
import { AppRegistry, Text } from 'react-native';

const HelloWorldApp = () => <Text>Hello world!</Text>;
export default HelloWorldApp;
// Skip this line if using Create React Native App
AppRegistry.registerComponent('HelloWorld', () => HelloWorldApp);
// The React native code can also be imported from another component
with the following code:
import HelloWorldApp from './HelloWorldApp';
```

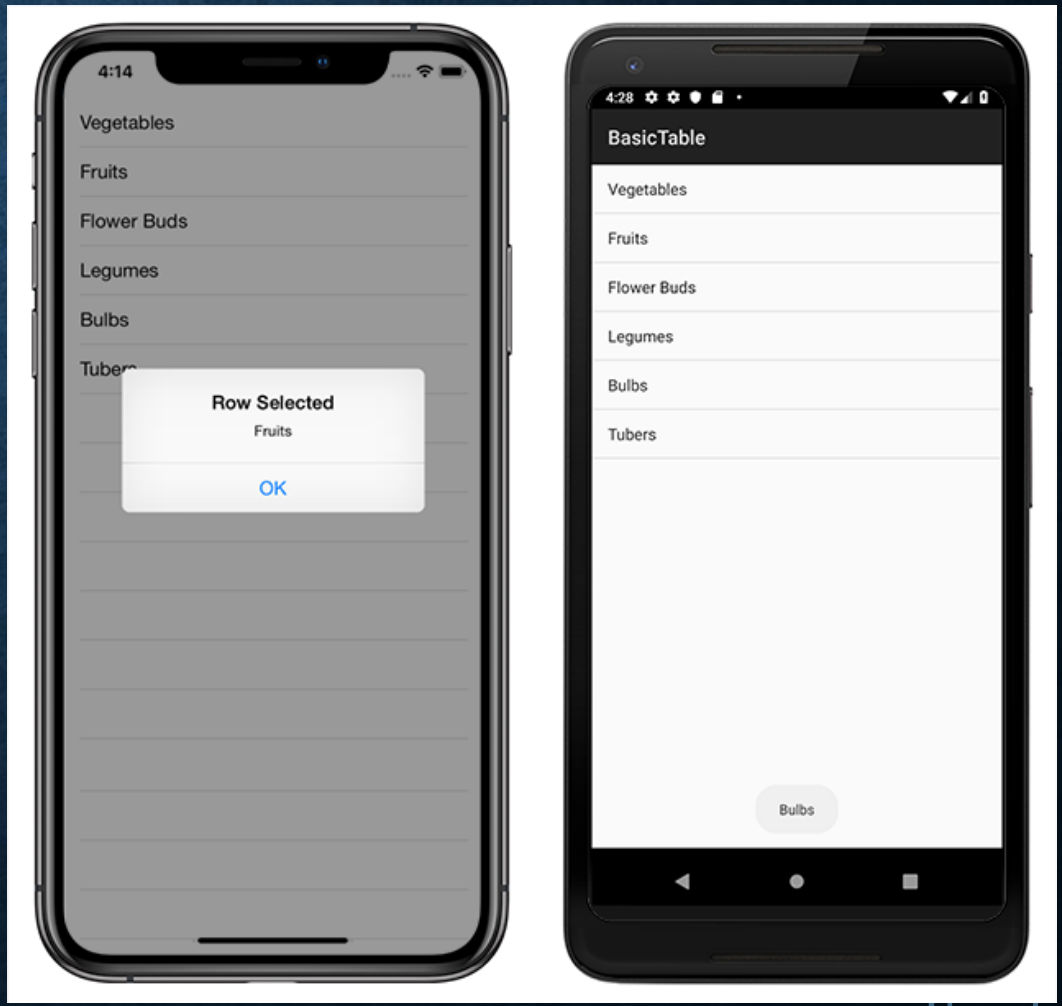
CORDOVA

- Originally Developed by Nitobi -> PhoneGap -> Apache Cordova
- Open Source
- Supports: Android, IOS, Windows Mobile
- Initial release: 2009
- Written in: HTML, CSS, JS, C#
- Generally used by quite a few other frameworks as plugins
- Based on Node
- https://cordova.apache.org/



XAMARIN

- Developed by Xamarin (A Microsoft Subsidiary)
- Supports: Android, IOS, Windows Mobile
- Initial Release: May 2011
- Written in: C# and .NET
- https://dotnet.microsoft.com/apps/xamarin



IONIC

- Developed by Drifty Co
- Open Source
- Supports: Android, IOS, Windows
- Initial Release: 2013
- Built on top of Angular and Cordova
- Written in: JavaScript, HTML, CSS, Angular, React or Vue
- We will focus on learning Ionic
- https://ionicframework.com/

IONIC

- There are 3 options to code using the Ionic Framework:
  - Angular
  - React
  - Vue

- We will discuss the Angular version since it is the easiest and more similar to what you are familiar with ☺

IONIC - INSTALLATION

- Firstly to install Ionic you need NodeJS:

```
npm install -g @ionic/cli
```

- To create an app

```
ionic start myApp sidemenu
```

IONIC

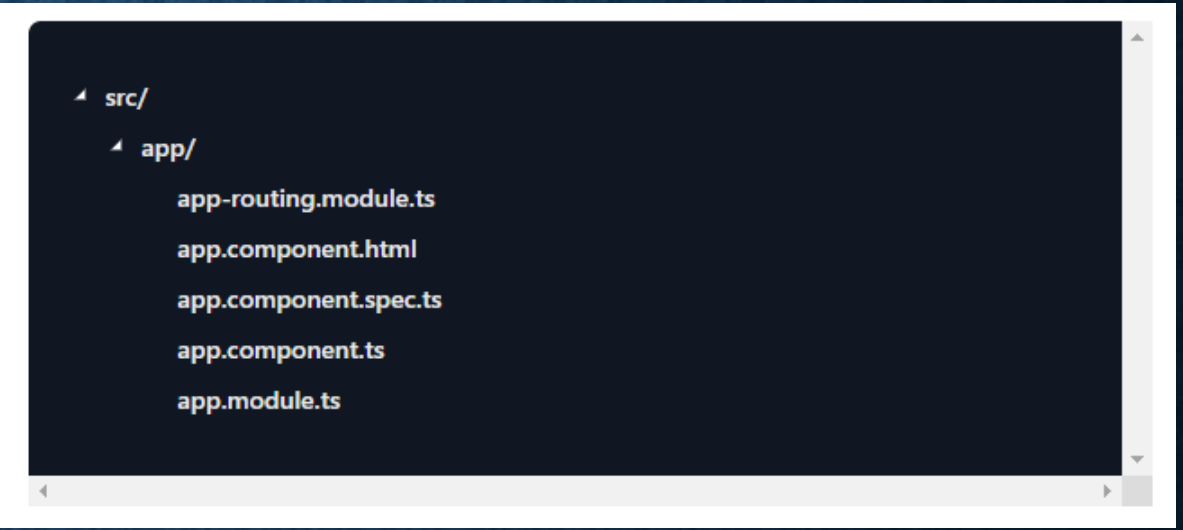
- There after it will take some time to install a bunch of libraries
- To run the app use: ionic serve
- This will spin up a webserver and serve the mobile application in your web browser.
- If you wish to see multiple layouts at once use: ionic serve --lab
- The nice thing about Ionic is that it supports live reload ☺

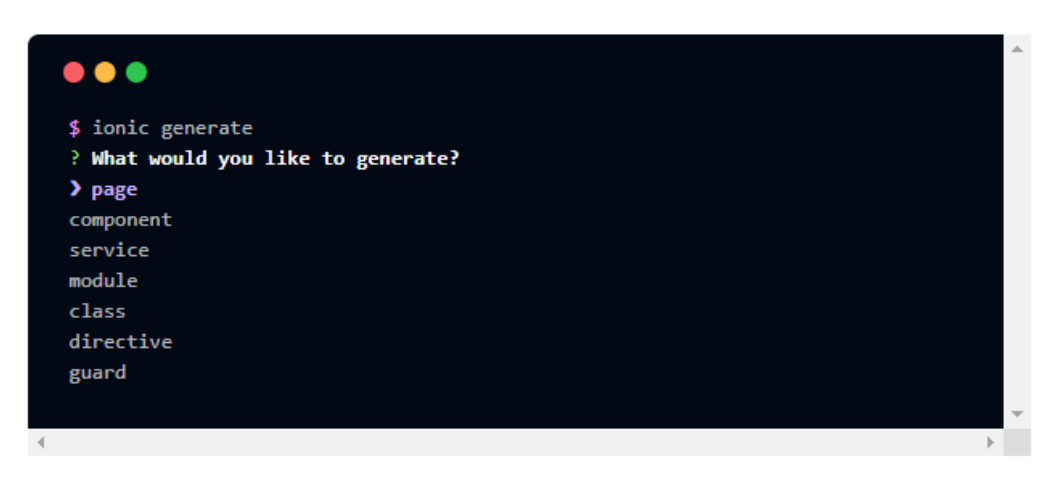
IONIC - STRUCTURE

- Thereafter, if you see your project structure it follows the same principles of what we discussed in the Angular Lecture.

IONIC - COMPONENTS

- Just like the Angular Lecture we can also generate components

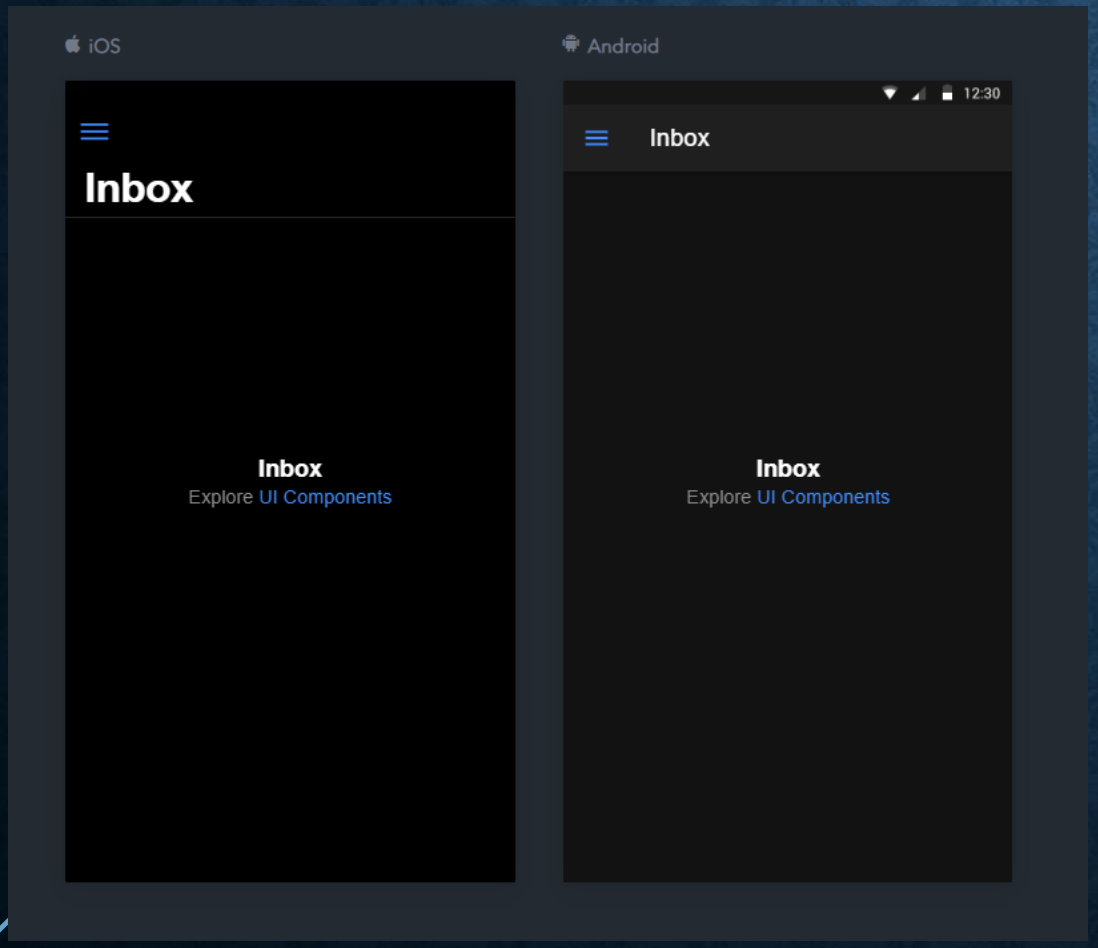


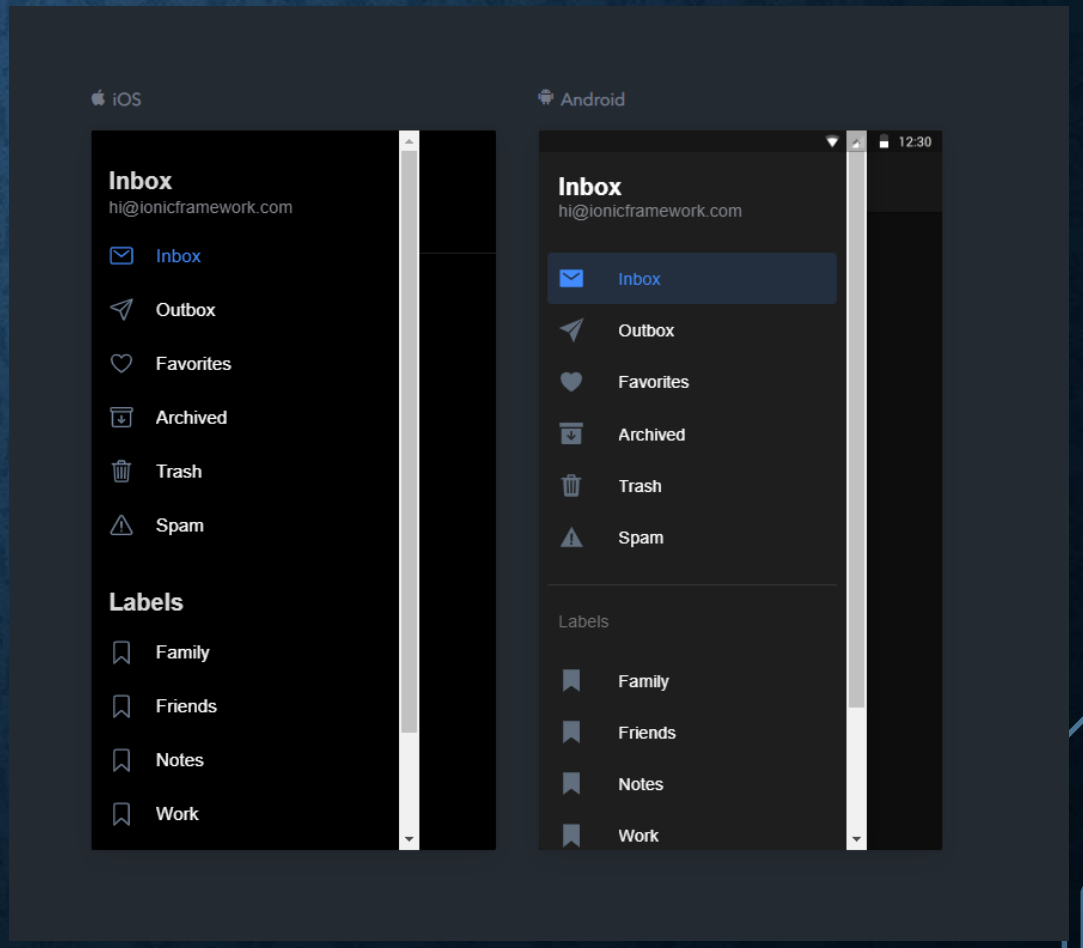


IONIC – ANDROID SDK

- With all hybrid frameworks, you still need to have the Android SDK and build tools installed.
- If you have not done so already, please install Android studio, it is available through Google Drive if you need it Zero-rated.
- However, this is only needed if you generating an APK file.

IONIC SIDE MENU APP





app.components.html

IONIC

```
<ion-app>
  <ion-split-pane contentId="main-content">
    <ion-menu contentId="main-content" type="overlay">
      <ion-content>
        <ion-list id="inbox-list">
          <ion-list-header>Inbox</ion-list-header>
          <ion-note>hi@ionicframework.com</ion-note>

          <ion-menu-toggle auto-hide="false" *ngFor="let p of appPages; let i = index">
            <ion-
item (click)="selectedIndex = i" routerDirection="root" [routerLink]="[p.url]" lines="none" detail="false" [class.selected]="selectedIndex == i">
              <ion-icon slot="start" [ios]="p.icon + '-outline'" [md]="p.icon + '-sharp'"></ion-icon>
              <ion-label>{{ p.title }}</ion-label>
            </ion-item>
          </ion-menu-toggle>
        </ion-list>

        <ion-list id="labels-list">
          <ion-list-header>Labels</ion-list-header>

          <ion-item *ngFor="let label of labels" lines="none">
            <ion-icon slot="start" ios="bookmark-outline" md="bookmark-sharp"></ion-icon>
            <ion-label>{{ label }}</ion-label>
          </ion-item>
        </ion-list>
      </ion-content>
    </ion-menu>
    <ion-router-outlet id="main-content"></ion-router-outlet>
  </ion-split-pane>
</ion-app>
```

app.components.ts

```
public appPages = [
  {
    title: 'Inbox',
    url: '/folder/Inbox',
    icon: 'mail'
  },
```

app.module.ts

IONIC

```
@NgModule({
  declarations: [AppComponent],
  entryComponents: [],
  imports: [
    BrowserModule,
    IonicModule.forRoot(),
    AppRoutingModule
  ],
  providers: [
    StatusBar,
    SplashScreen,
    { provide: RouteReuseStrategy, useClass: IonicRouteStrategy }
  ],
  bootstrap: [AppComponent]
})
export class AppModule {}
```

app.component.ts

IONIC

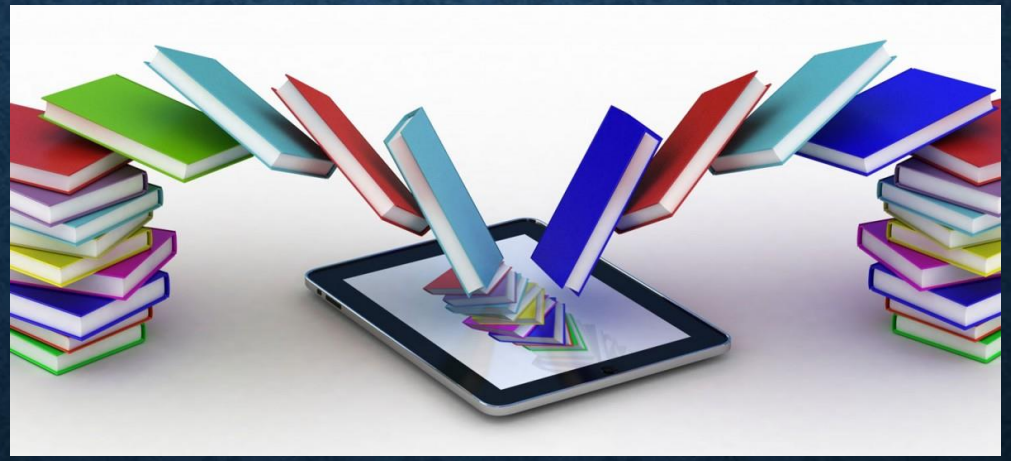
```
constructor(
  private platform: Platform,
  private splashScreen: SplashScreen,
  private statusBar: StatusBar
) {
  this.initializeApp();
}

initializeApp() {
  this.platform.ready().then(() => {
    this.statusBar.styleDefault();
    this.splashScreen.hide();
  });
}

ngOnInit() {
  const path = window.location.pathname.split('folder/')[1];
  if (path !== undefined) {
    this.selectedIndex = this.appPages.findIndex(page => page.title.toLowerCase() === path.toLowerCase());
  }
}
```

IONIC - DOCUMENTATION

- https://ionicframework.com/docs/




if (czy_zalogowa
przekieruj('logowanie','102
exit;
}
<scri
14
15
16
17
18
ret
func
false);