1. **Creating React Native Project**
   1. Installing Dependencies
      1. Xcode on Mac.
      2. Android Studio on Mac/Window.
      3. Node Js
      4. React Native CLI
   2. Creating Project Solution
      1. react-native init yourprojectname
      2. For iOS
         1. Run Xcode > Open Project > yourprojectsolution>ios>yourprojectname.xcodeproj.
         2. Make sure you have full privilege.
         3. Build your Project.
         4. Run your project from Xcode or by command line react-native run-ios on terminal.
      3. For Android
         1. Run Android Studio > Open Project > yourprojectsolution>android>build.gradle.
         2. Make sure you have full privilege.
         3. Build your project.
         4. Run android Emulator – Tool>AVD Manager> start emulator from list.
         5. Run you project by command line react-native run-android.
2. **App Icon & Launch Screen.**
   1. For iOS.
      1. For App Icon
         1. Xcode>yourproject>Images.xcassests.
         2. Click on App Icon option.
         3. Add Icon with required resolution.
      2. For App Launch Screen.
         1. Xcode>yourproject>LaunchScreen.xib.
         2. Choose an initial device view.
         3. Edit your project launch screen.
   2. For Android.
      1. Android Studio > yourprojectsolution>select app>right click> select image asset from menu.
      2. Configure Image Asset.
3. **Adding React Native Navigation.**
   1. iOS Installation
      1. Install react-native-navigation. Terminal- npm install react-native-navigation.
      2. In Xcode, in Project Navigator (left pane), right-click on the Libraries > Add files to [project name]. Add ./node\_modules/react-native-navigation/ios/ReactNativeNavigation.xcodeproj (screenshots)
      3. In Xcode, in Project Navigator (left pane), click on your project (top), then click on your target row (on the "project and targets list", which is on the left column of the right pane) and select the Build Phases tab (right pane). In the Link Binary With Libraries section add libReactNativeNavigation.a (screenshots)
      4. In Xcode, in Project Navigator (left pane), click on your project (top), then click on your project row (on the "project and targets list") and select the Build Settings tab (right pane). In the Header Search Paths section add $(SRCROOT)/../node\_modules/react-native-navigation/ios. Make sure on the right to mark this new path recursive (screenshots)
      5. In Xcode, you will need to edit this file: AppDelegate.m.
      6. Replace all of its code with this reference
      7. Replace @"index.ios" with @"index" if you are using index.js as your entry point instead of index.ios.js and index.android.js
      8. Add #import "RCCManager.h"
      9. Add [[RCCManager sharedIntance] initBridgeWithBundleURL:jsCodeLocation launchOptions:launchOptions];
   2. Android Installation
      1. Install react-native-navigation. Terminal - npm install react-native-navigation.
      2. Add the following in android/settings.gradle
         1. include ':react-native-navigation'
         2. project(':react-native-navigation').projectDir = new File(rootProject.projectDir, '../node\_modules/react-native-navigation/android/app/')
      3. Update project dependencies in android/app/build.gradle.
         1. dependencies {
         2. compile fileTree(dir: "libs", include: ["\*.jar"])
         3. compile "com.android.support:appcompat-v7:23.0.1"
         4. compile "com.facebook.react:react-native:+"
         5. compile project(':react-native-navigation')
         6. }
      4. In MainActivity.java it should extend com.reactnativenavigation.controllers.SplashActivity instead of ReactActivity.
         1. This file can be located in android/app/src/main/java/com/yourproject/.
         2. import com.reactnativenavigation.controllers.SplashActivity;
         3. public class MainActivity extends SplashActivity {
         4. }
      5. In MainApplication.java, add the following
         1. import com.reactnativenavigation.NavigationApplication;
         2. public class MainApplication extends NavigationApplication {
         3. @Override
         4. public boolean isDebug() {
         5. // Make sure you are using BuildConfig from your own application
         6. return BuildConfig.DEBUG;
         7. }
         8. protected List<ReactPackage> getPackages() {
         9. // Add additional packages you require here
         10. // No need to add RnnPackage and MainReactPackage
         11. return Arrays.<ReactPackage>asList(
         12. // eg. new VectorIconsPackage()
         13. );
         14. }
         15. @Override
         16. public List<ReactPackage> createAdditionalReactPackages() {
         17. return getPackages();
         18. }
         19. if you are using index.js as your entry point instead of index.ios.js and index.android.js. Add the following:
         20. @Override
         21. public String getJSMainModuleName() {
         22. return "index";
         23. }
4. **Adding Vector Icons.**
   1. iOS Installation.
      1. npm install react-native-vector-icons –save
      2. Browse to node\_modules/react-native-vector-icons and drag the folder Fonts (or just the ones you want) to your project in Xcode.
      3. Edit Info.plist and add a property called Fonts provided by application (or UIAppFonts if Xcode won't autocomplete/not using Xcode) and type in the files you just added.
      4. add RNVectorIcons.xcodeproj to Libraries
      5. add libRNVectorIcons.a to Link Binary With Libraries under Build Phases
   2. Android Installation.
      1. npm install react-native-vector-icons –save
      2. Edit android/app/build.gradle ( NOT android/build.gradle ) and add the following:
      3. apply from: "../../node\_modules/react-native-vector-icons/fonts.gradle"
5. **Adding Redux.**
   1. Installing Dependencies
      1. npm install redux
      2. npm install react-redux
      3. npm install redux-logger
      4. npm install redux-thunk
      5. npm install redux-persist