**React Native Application**

1. **Using EXPO CLI**

**Open Command Prompt**

* Need an account on expo for apk building and need expo app in mobile to run application in debug mode.
* Expo CLI install (**npm install -g expo-cli**)
* Expo init Comunev( New project named comunev initialized)
* Change icon and splash screen icon in assets folder
* All the react dependencies are there in the package.json file which can be installed through

**npm install name\_of\_dependency**

* The default function is App in the App.js file.
* All other screens and features of the application are there in separate js files in the component folder which are imported where needed.

**To run app on emulator=> expo start**

* Then a QR code will open which can be scanned through expo app in mobile and app will run in mobile in debug mode and all logs can be seen on command prompt.

**To build apk file=> expo build:android**

* Enter your expo account credentials on cmd
* Click on apk bundle and then open dashboard of expo account
* There apk building will be in progress and can be downloaded.

1. **Using React-Native CLI**

* **Environment Set-Up**

Detailed and latest steps available on official site.

<https://reactnative.dev/docs/environment-setup>

Python, JDK, Android Studio and react-native CLI

* For new project

On cmd type: react-native init Name

Then cd Name

* To clone our existing Project

Open Gitbash or sourcetree

git clone link\_of\_repository

* First open command Prompt inside folder where app.js file is present and run : **npm install**

This will install all dependencies from package.json

* **To run in debug mode:**

npm run android

**Note:**

**It will give an error of razorpay, then go to build.gradle file inside android folder.**

**Change : minSdkVersion to 16**

**After that again run using npm run android.**

**It will again give error. Then again change minSdkVersion to 19 and run.**

**Now it will run without error.**

buildscript {

    ext {

        buildToolsVersion = "29.0.2"

        minSdkVersion = 19

        compileSdkVersion = 29

        targetSdkVersion = 29

    }

if on running above command, our port is already in use and we get error then run following:

<PORT> is the port number

<PID> is the process id we get after running command 1.

1. netstat -ano | findstr :<PORT>
2. taskkill /PID <PID> /F

**Structure of Existing Project**

1. Signin Screen -> SignUp Screen-> Interest Select

Screen (Sign Up Successful)

-> About us

-> Contact Us

-> Become venue partner

-> Sponsor us

-> Become a speaker

-> Partner with us

-> Event-> Event display->

Register-> Payment

2. After Login -> Profile Screen(Profile)

-> Matchmaking page(Add)

-> Event Screen(Event) -> Event

display-> Register-> Payment

-> Chat Screen (chat)

-> About us

-> Contact Us

-> Become venue partner

-> Sponsor us

-> Become a speaker

-> Partner with us

* App.js is the main file where function App( ) is called by default.
* First Retrieve\_Token method is run which checks whether a user is already logged in or not from async storage(Local storage for phones )
* If token is found, User Profile screen is displayed else Signin Screen is displayed.
* All the Screen designs such as about us, contact us, chat etc. are there in the component folder.
* A folder named assets is there in :

C:\Users\HP\Documents\Abhyudaya\Internship\mobilerepo\Comunev\android\app\src\main\assets

All pictures are in this folder. Png format is must as other format are not accepted.

* React navigation drawer, tab ,stack and login idea is explained in detail in this small series.

[**https://www.youtube.com/playlist?list=PLQWFhX-gwJbmmqcP-9zMXBaxQbGKfIJY2**](https://www.youtube.com/playlist?list=PLQWFhX-gwJbmmqcP-9zMXBaxQbGKfIJY2)

**To connect with backend with get or post request:**

1. Get Request

fetch(“API link for backend/Function\_name”,{

      method:"get",

      headers:{

        'Content-Type':'application/json'

      },

    })

    .then(res=>res.json()).then(data=>{

    //All information from backend received in data variable.

    })

1. Post Request

fetch(“API link for backend/method\_name”,{

      method:"post",

      headers:{

        'Content-Type':'application/json'

      },

      body:JSON.stringify({

       variable1:value1,

variable2: value2 (and so on)

      })

    })

    .then(res=>res.json()).then(data=>{

    //All information from backend received in data variable.

    })

In post request name of variable 1 and variable 2 etc. are same as used in backend and value value 2 etc are the values we pass from our side like email, password etc.

**To make fetch( ) part working in apk.**

If we run application on emulator, fetch part works fine but on running on apk, it does not work due to mismatch of http and https issue.

So we need to add this line below in AndroidManifest.xml file inside the

C:\Users\HP\Documents\Abhyudaya\Internship\mobilerepo\Comunev\android\app\src\main

Folder

 android:usesCleartextTraffic="true"

**To Change Application Icon**

* If we need to remove default icon and setup our own icon, we need to generate our icon in specified format from **Android Asset Studio.**

Place the downloaded icons in

C:\Users\HP\Documents\Abhyudaya\Internship\mobilerepo\Comunev\android\app\src\main\res

Folder where each folder must have 2 icons. For second icon, copy the first icon in existing folder and rename it as ic-launcher-round.

* Next step is to open command prompt in android folder and run **gradlew clean**
* Then go back to main folder where App.js present and run

**npm run android**

The icon is changed!!

**To Generate APK file**

1. In the main folder where App.js is present, open command prompt and run

**keytool -genkey -v -keystore my-release-key.keystore -alias my-key-alias -keyalg RSA -keysize 2048 -validity 10000**

**It will ask to set new password and name details on command prompt itself, then a new file named:**

MYAPP\_RELEASE\_STORE\_FILE

**Is generated which is cut and pasted in android/app folder**

* **Now paste below 4 lines in** ~/.gradle/gradle.properties
* MYAPP\_RELEASE\_STORE\_FILE=my-release-key.keystore
* MYAPP\_RELEASE\_KEY\_ALIAS=my-key-alias
* MYAPP\_RELEASE\_STORE\_PASSWORD=\*\*\*\*(Your password set on cmd)
* MYAPP\_RELEASE\_KEY\_PASSWORD=\*\*\*\*(same password)

1. **Now inside android folder run gradlew assembleRelease**

**Which will generate the apk file in**

**C:\Users\HP\Documents\Abhyudaya\Internship\mobilerepo\Comunev\android\app\build\outputs\apk\release**

**Folder.**

**Note: Step 1 need to be done first time only, after that only step 2.**

**Reduce SIZE of APK File**

android.enableR8=true

org.gradle.daemon=true

org.gradle.jvmargs=-Xmx2560m

* Add the above three lines in gradle.properties file inside android folder.
* Inside C:\Users\HP\Documents\Abhyudaya\mobilerepo\Comunev\android\app\build.gradle file , Set these 2 as true.
* def enableSeparateBuildPerCPUArchitecture = true
* /\*\*
* \* Run Proguard to shrink the Java bytecode in release builds.
* \*/
* def enableProguardInReleaseBuilds = true

and in the same file below, set universal apk to true.

 splits {

        abi {

            reset()

            enable enableSeparateBuildPerCPUArchitecture

            universalApk true  // If true, also generate a universal APK

            include "armeabi-v7a", "x86", "arm64-v8a", "x86\_64"

        }

**CREATING CUSTOM ICONS OTHER THEN VECTOR ICONS**

https://www.reactnative.guide/12-svg-icons-using-react-native-vector-icons/12.1-creating-custom-iconset.html

Author

Aditi Tiwari

Abhyudaya Mittal