**For Create space in mac:**go terminal and run these commands  
cd /Users/rabia/Library/developer/Xcode/DerivedData  
rm –rf \*  
**Download XCODE:**  
<https://stackoverflow.com/questions/10335747/how-to-download-xcode-dmg-or-xip-file>

**Linking**:  
is used for opne hyper link dialer ,mailer etc  
<https://reactnative.dev/docs/linking>  
**Deep Linking:**<https://medium.com/react-native-training/deep-linking-your-react-native-app-d87c39a1ad5e>  
https://www.youtube.com/watch?v=2kuvbfFuVmo

**Share Anything on social media like on what’s app etc:**<https://reactnative.dev/docs/share>

**Blur:**  
<https://github.com/react-native-community/react-native-blur>

**Shadow:**  
<https://github.com/879479119/react-native-shadow>

**Height and Wifth of React-native:**  
<https://dev.to/skptricks/react-native-get-device-height-width-on-button-click-19ld>

**url validation javascrip:**<https://stackoverflow.com/questions/5717093/check-if-a-javascript-string-is-a-url>  
  
**open of google map with direction:**<https://stackoverflow.com/questions/43214062/open-maps-google-maps-in-react-native>

**create url of map with direction origin and destination:**

* Open google map<https://developers.google.com/maps/documentation/urls/ios-urlscheme>
* <https://stackoverflow.com/questions/45116011/generate-a-google-map-link-with-directions-using-latitude-and-longitude>
* <https://developers.google.com/maps/documentation/urls/ios-urlscheme>

**Scalling Drawer:**<https://www.npmjs.com/package/react-native-scaling-drawer>

**Video player:**<https://github.com/uit2712/Play-video-in-React-Native>  
<https://aboutreact.com/react-native-video/>

<https://www.youtube.com/watch?v=swigQ10SL_w>

**Height and width of screen:**<https://dev.to/skptricks/react-native-get-device-height-width-on-button-click-19ld>

get height and width of screen and apply this while in percentage its good practice when add shadow.  
<https://dev.to/skptricks/react-native-get-device-height-width-on-button-click-19ld>

const screenWidth = Math.round(Dimensions.get('window').width);

width=(screenWidth/100)\*10  
this is formula is we give 10 width

**Star Rating Library:**<https://www.npmjs.com/package/react-native-star-rating>

**Change name of app:**

Change name into app.json  
change name into ios info.plist  
change name into android resourse string.  
  
///////////////  
**Drwer**:  
make saperate fie in this file we write code for drawer and this drawer will be implement on any button or event at any screen.  
then import Drawer.

import SideBar from '../../Drawer/Drawer'

write both of these function where drawer implement

closeDrawer = () => {

    this.drawer.\_root.close()

}

openDrawer = () => {

    // console.log("i am")

    this.drawer.\_root.open()

};

We also require

<Drawer  ref={(ref) => { this.drawer = ref; }}

                openDrawerOffset={0.05}

                content={<SideBar close={() => this.closeDrawer()} />}>  
      <Icon name='menu' onPress={()=>this.openDrawer()}  type='Entypo' style={{ margin: 10, fontSize: 28, }} /> //It will open Drawer

code of second screen</Drawer>

//**Navigation**:  
Make Navigation Folder and then file Navigation.js and import all files which u want to navigate  
install and Import this

 import { Router, Scene, Stack } from 'react-native-router-flux';

We create Scene for every imported file which will be navigate.

 import Login from '../Modules/Singup/Login'

 <Router>

         <Stack key="root" hideNavBar>

 <Scene key="login" component={Login} title="personal\_information"/>

            <Scene key="qualification" component={Qualification} title="qualification"/>

</Stack>

       </Router>

We use key for navigation in another file for example we want to navigate at loging screen  
onPress={() => Actions.Choice()}

**State:**State is access able in the class. Mostly variables are store into the state

State={  
a=0,  
b=’samran}  
state can also be define into the constructor

constructor(props) {

    super(props);

    this.state = {

      tab\_value: 0,

      load: true

    };

  }

**Props:**  
Props are work like a friend function if want to share our function or variable with another class then we create props.

First import class from which you want to share function and make function. Here personal\_information is class from which we share function updatetabe and we access with name updateTabe and v is parameter which is passed.

                  <Personal\_Information updateTab={(v) => this.updatetab(v)} />

**Array and map arry:**Describe arrays in state   
one is simple and one is json type

country: ['Pakistan', 'India', 'UK', 'UAE'],

      applying:[{key:'Key1',name:'School'},

      {key:'Key0',name:'College'}, ]

We can map array in this way

{

              this.state.country.map((data, index) => (

                  <Picker.Item label={data} value={index} />

              ))

            }

Data contain data of array and index tell us index of array.it will work like loop

**Tabs :**

 <Tabs onChangeTab={(c)=>this.updatetabb(c.i)} initialPage={this.state.tab\_value} tabBarUnderlineStyle={{ borderBottomColor: "red", borderBottomWidth: 4 }} renderTabBar={() => <ScrollableTab />}>

                <Tab heading="Personal Information" tabStyle={{ backgroundColor: '#f6f7f9' }} activeTabStyle={{ backgroundColor: '#f6f7f9' }} textStyle={{ color: 'black', fontFamily: 'cretype - Artico ExtraCond Light' }} activeTextStyle={{ color: 'red', fontWeight: 'normal', fontFamily: 'cretype - Artico ExtraCond Light' }}>

                  <Personal\_Information updateTab={(v) => this.updatetab(v)} />

                </Tab>

                <Tab heading="Qualification" tabStyle={{ backgroundColor: '#f6f7f9' }} activeTabStyle={{ backgroundColor: '#f6f7f9' }} textStyle={{ color: 'black', fontFamily: 'cretype - Artico ExtraCond Light' }} activeTextStyle={{ color: 'red', fontWeight: 'normal', fontFamily: 'cretype - Artico ExtraCond Light' }}>

                  <Qualification updateTab={(v) => this.updatetab(v)} />

                </Tab>

              </Tabs>

onChangeTab is used to call any function or event handler on tab change.initialPage={0} it will open first tab  
initialPage={1} it will open second page

updatetab = (v) => {

  //  alert('calling')

    this.setState({ load: false })

    this.setState({ tab\_value: v })

    setTimeout(() => {

      this.setState({ load: true })

    }, 0.0001);

  }

We make this function as prop and use in file of tab and send value which set in initialPage={0} and load variable will be used for refresh the particular compone**nt.  
Make APK For Android:**Go to project and then android and open CMD and run command (gradlew assemblerelease)  
we found APK and this path H:\Office\_Work\Enroll\_project\Mobile\enroll1\android\app\build\outputs\apk\release  
**StyleSheet:**write all outline stying here for View,Text,Button etc

import { StyleSheet, Dimensions } from 'react-native';

import { Fonts } from '../../src/utils/Fonts'

export default StyleSheet.create({

staric:{marginTop:25,color:'#f23d3d'},

input\_view:{flexDirection:'row'}

})

this will make text size constant in all size of screens text size will remain same

<Text allowFontScaling={false}>SAMRAN</Text>

**ProgressBar:**

              <Progress.Bar progress={this.state.barprogres} height={6} width={280} color='#f23d23' unfilledColor='white' style={{ width: '90%' }} />

Progress bar will fill according to value into progress

**StatusBar:**most Upper part of application

 <StatusBar hidden={false} backgroundColor='#f23d3d' barStyle="light-content" />

<https://www.youtube.com/watch?v=CpXWG4ePFdM&list=PLQMSKd-xIgDSGlb-V4gu1j7kI8lWSLDxN>

**Distance Calculate Formula from Latitude and longitude :**

distance=(lat1, lon1, lat2, lon2, unit)=> {

if ((lat1 == lat2) && (lon1 == lon2)) {

alert(0)

}

else {

var radlat1 = Math.PI \* lat1/180;

var radlat2 = Math.PI \* lat2/180;

var theta = lon1-lon2;

var radtheta = Math.PI \* theta/180;

var dist = Math.sin(radlat1) \* Math.sin(radlat2) + Math.cos(radlat1) \* Math.cos(radlat2) \* Math.cos(radtheta);

if (dist > 1) {

dist = 1;

}

dist = Math.acos(dist);

dist = dist \* 180/Math.PI;

dist = dist \* 60 \* 1.1515;

if (unit=="K") { dist = dist \* 1.609344 }

if (unit=="N") { dist = dist \* 0.8684 }

// return dist;

// alert(dist.toFixed(6))

this.setState({

distance:dist.toFixed(6)

})

}

}