Date .....

React Native

1. To setup
   npm install -g expo-cli

2. expo init ./
3. choose first option
4. npm start
5. click on localhost 19002
6. on side bar there are options to run on different devices
7. Copy package.json & paste it in your root folder
8. npm install

   View → same as div
   text → same as paragraph
9. remove everything inside ~~navigator~~ return
10. Below return there are styles remove that also
11. copy import from App.js & paste

Navigation → There are multiple screens in almost all the apps; you'll barely find an app with just single screen. Managing the presentation of screens and transitions between them is handled by navigator.

So if you have more than a couple of screens, you need to define routing & navigation that is scalable & easy to maintain

Date .....

If you are beginner, it is recommended to use react navigation.

☆ Navigation Container

The navigation container is responsible for managing your app state & linking your top level navigator to the app environment.

It performs following functionality :-

(i) Deep link integration with linking prop

(ii) Notify ~~changes~~ state changes for screen tracking, screen persistence, etc.

(iii) Handle system back button on android by using Back handler api from react native.

☆ ~~Nav~~ Stack Navigator

It provides a way for your app to transition b/w screens where each new screen is placed on top of a stack.

```
const Stack = createStackNavigator();

return(
  <Stack.Navigator>
    <Stack.Screen name="Home" component={Home}/>
  </Stack.Navigator>
```

creating a screen

✱ Themes

Themes allow you to change the colors of various components provided by react navigation. you can use themes to:-

- customize the colors match your brand
- provide light & dark theme

Creating a custom theme

```
const MyTheme = {
    ...DefaultTheme,
    colors: {
      ...DefaultTheme.colors,
      primary: 'rgb(0,0,0)',
    },
};

<NavigationContainer theme={MyTheme}>
</NavigationContainer>
```

A theme is a js object containing colors to use. properties:-

dark (boolean): whether dark or light
colors ⇒ primary → primary color of app
background → color of various backgrounds
card → background color of card like components
text → text color
border → color of borders
notification ⇒ color of tab navigator badge

Date .....

★ Stack. Navigator component accepts following props:-

1. id → optional unique ID for the navigator. This can be used with navigation.getParent to refer to this navigator in a child navigator.
2. initialRouteName → name of route to render on first load of navigator.
3. screenOptions → Default options to use for the screens in navigator.
4. detachInactiveScreens → Boolean used to indicate whether inactive screens should be detached from the view hierarchy to save memory. Defaults to true
5. keyboardHandlingEnabled → If false, the keyboard will not automatically dismiss when navigating to new screen. Defaults to true.

★ To create a new screen:-

make a new folder screens in root folder inside that folder create Details.js & Home.js inside any of it use rafce & create your component

★ Flatlist

displays the similar structured data in a scrollable list. It works well for large lists of data where the number of list items might change over time. It only shows those elements which are currently displaying on screen, not all elements at once. can simply be considered as map fn

★ Safeareg → to render components within safearea boundaries of a device.

Spiral

Date .....

Copy & paste constants folder

→ Create components folder
in Focused StatusBar use rnfe

★ StatusBar

Component to control app's status bar. The status bar is the zone, typically at the top of the screen, that displays current time, wifi and cellular network information, battery level and/or other status signals.

★ useIsFocused

we might want to render different content based on the current focus state of the screen. This hook triggers a re-render for the screen when it changes focus.

★ useNavigation

useNavigation is a hook which gives access to navigation object. It is useful when you cannot pass the navigation prop into components directly, or don't want to pass it in case of a deeply nested child.
It returns the navigation prop of the screen it is inside.

! When we have export default we can only return one component but when we use 'export const function' you can return multiple components