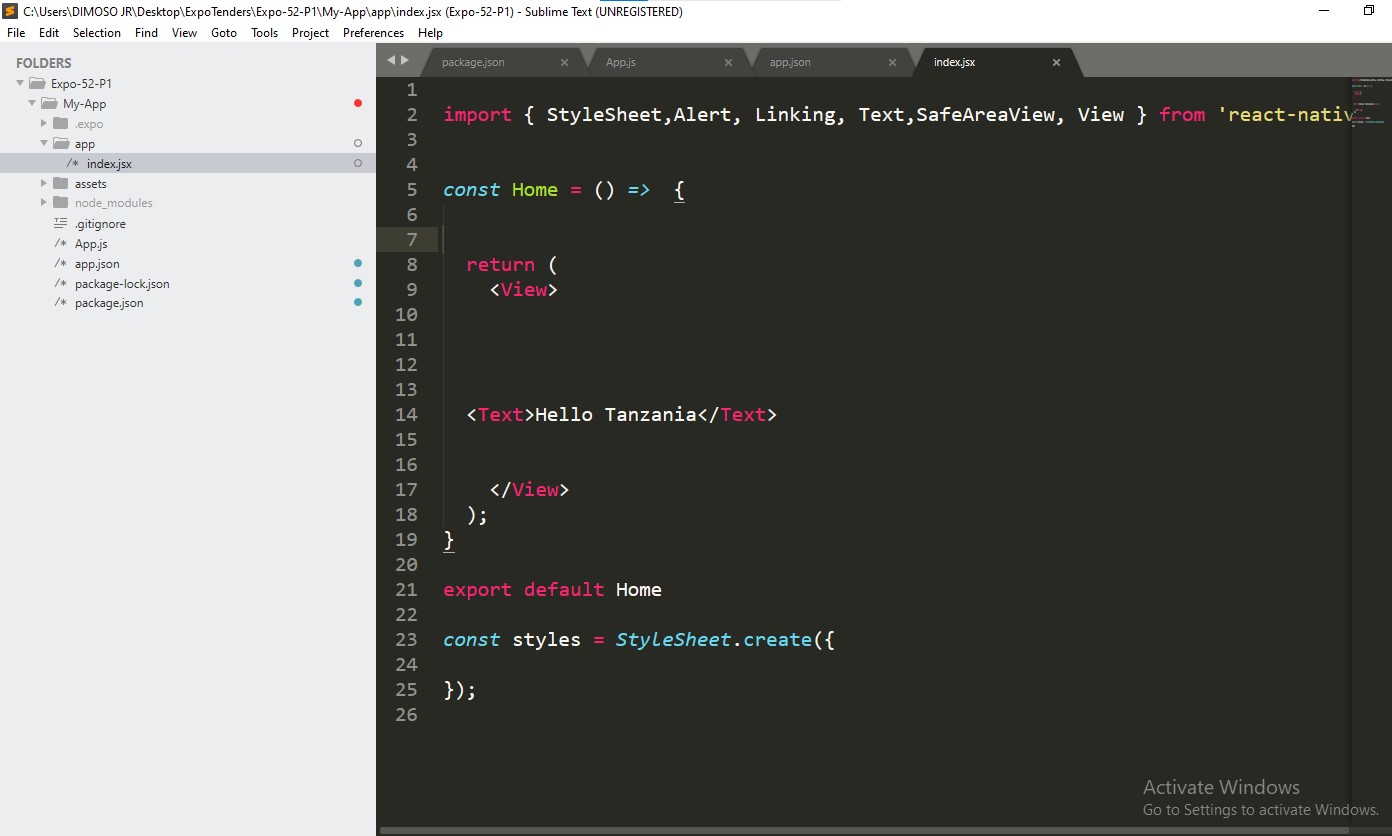
**TO SETUP NEW EXPO PROJECT SDK 52 USING EXPO ROUTER**

* **npm install –g expo-cli**
* npx create-expo-app@latest My-App –t expo-template-blank@sdk-52
* cd My-App
* npx expo install expo-router react-native-safe-area-context react-native-screens expo-linking expo-constants
* npm start

****

Hivi naona kila tutorial za expo react native SDK 52 ni tofauti na zile za SDK 51 hata format ni tofauti naona sijui kwenye SDK 52 kuna mambo ya expo router, stack ni tofauti na 51, yani vitu viko tofauti kabisa. Sasa project iliyotengenezwa kwa expo react native SDK 51 huwezi kuitumia kwenye 52, maana naona huku kwenye 52 mafaili wanatumia jsx mimi kwenye 51 nilikuwa natumia js, kwahiyo huku siwezi kutumia mafaili ya js. Yani sielewi hebu nipe full explanations ya haya mabadiliko

**ChatGPT said:**

Umeuliza swali zuri sana 👍.  
Ni kweli kabisa **Expo SDK 52** inaleta mabadiliko makubwa ukilinganisha na **SDK 51** na versions zilizopita. Hii ndio sababu unaona tutorials za zamani haziendani na zile mpya. Hebu nikuchambulie hatua kwa hatua ili uelewe vizuri:

**🔑 1. Expo Router na App Structure**

* **SDK 51 na nyuma yake**: Mara nyingi tulikuwa tunatumia App.js kama entry point, kisha tukajipangia navigation kwa kutumia react-navigation.
* **SDK 52**: Sasa default inakuja na **Expo Router** ambayo imechukua nafasi ya react-navigation kwenye project mpya.
  + Hii inamaanisha navigation sasa inategemea **file-system based routing** (mfano app/index.jsx, app/profile/[id].jsx, nk).
  + Badala ya kuandaa navigator kwa mkono, router inachukua majina ya files na folders kama route zako.

**🔑 2. File Extensions (.js vs .jsx)**

* **SDK 51**: Wengi walitumia .js hata kwa components zenye JSX. Hii haikuwa shida.
* **SDK 52**: Convention mpya inapendekeza kutumia .jsx au .tsx (kama unatumia TypeScript) kwa files zenye JSX.
  + Kwa nini? Ili editor (VSCode, WebStorm nk) na bundler zitambue wazi kuwa file lina JSX.
  + Hata hivyo, **bado unaweza kutumia .js** — sio kosa. Ni kwamba tutorials mpya zinahimiza .jsx kwa clarity na standard mpya.

**🔑 3. Navigation Format (Stack, Tabs, Drawer)**

* **SDK 51**: Ili kuunda navigation ulitumia createStackNavigator, createBottomTabNavigator, nk kutoka @react-navigation/\*.
* **SDK 52**: Sasa unatumia components za Expo Router kama:
  + <Stack />
  + <Tabs />
  + <Drawer />  
    Zimejengwa ndani ya Expo Router, na hazihitaji configuration ngumu kama zamani.

Mfano:

// SDK 52 structure

import { Stack } from "expo-router";

export default function Layout() {

return (

<Stack>

<Stack.Screen name="index" />

<Stack.Screen name="profile/[id]" />

</Stack>

);

}

**🔑 4. Project Structure**

* **SDK 51**:
* App.js
* screens/
* components/
* navigation/
* **SDK 52** (default Expo Router):
* app/
* \_layout.jsx // layout ya navigation
* index.jsx // home screen
* profile/
* [id].jsx // dynamic route
* components/

👉 Hii ndio inakufanya uone tutorials mpya ziko tofauti kabisa na za zamani.

**🔑 5. Migration**

* Project ya **SDK 51 haiwezi ku-run moja kwa moja kwenye 52** bila adjustments kwa sababu ya hizi tofauti kwenye structure na navigation.
* Lakini unaweza kuendelea kutumia:
  + **React Navigation kwenye SDK 52** kama hutaki Expo Router (inahitaji ku-install kwa mkono).
  + Au kubadilisha project yako kuendana na Expo Router.

**🔑 6. Kwa Nini Wamebadili?**

* Expo Router inaleta experience karibu sana na frameworks kama **Next.js** (file-based routing).
* Inarahisisha mambo kama:
  + Dynamic routes (/profile/[id])
  + Nested layouts
  + Automatic linking
* Ni future direction ya Expo, ndio maana tutorials mpya zote ziko hivyo.

**✅ Conclusion**

* SDK 52 **ina support ya .js bado**, ila convention mpya inasisitiza .jsx / .tsx.
* Mabadiliko makubwa yapo kwenye **navigation** (Expo Router badala ya React Navigation).
* Project ya SDK 51 unaweza ku-update, ila itahitaji adjustments (hasa kwenye routing).

👉 Swali: Unapenda nikuandikie **njia ya kubadilisha project yako ya SDK 51 kwenda SDK 52** bila kuanzisha from scratch (migration guide), au ungependa nikupe **mfano mpya wa project ya SDK 52** ili uone structure yake vizuri?

**You said:**

Naomba niandikie njia ya kubadilisha project ya SDK 51 kwenda SDK 52 bila kuanzisha from scratch

**ChatGPT said:**

Vizuri sana 👌 umeuliza pointi ya msingi.  
Kubadilisha project kutoka **Expo SDK 51 → SDK 52** bila kuanza upya inaitwa **migration**. Hapa nitakupa hatua zote unazopaswa kufanya.

**🚀 Hatua za Kubadilisha Project SDK 51 → SDK 52**

**1️⃣ Update Expo SDK**

Kwanza hakikisha uko kwenye Expo SDK 52.  
Kwenye terminal ya project yako:

npx expo install expo@^52.0.0

Kisha update dependencies zingine:

npx expo install

**2️⃣ Badilisha package.json**

Angalia package.json yako, hakikisha inaonyesha:

"expo": "^52.0.0"

Halafu fanya install upya dependencies:

npm install

au

yarn install

**3️⃣ Kuandaa Expo Router**

Kwa sababu Expo Router sasa ndio default navigation, install:

npx expo install expo-router

Kisha fungua **package.json** na ongeza script hii kama haipo:

"main": "expo-router/entry"

**4️⃣ Badilisha Structure ya Files**

SDK 51 ulikuwa na App.js kama entry point.  
SDK 52 inatumia **folder app/**.

➡️ Hatua ya kufanya:

1. Tengeneza folder jipya:
2. app/
3. Hamisha App.js → ndani ya app/ na rename kuwa index.jsx

Mfano:

app/

index.jsx // ilikuwa App.js zamani

components/

⚠️ Usijali kuhusu extension .js — unaweza kubaki nayo, ila inapendekezwa index.jsx.

**5️⃣ Tengeneza Layout kwa Navigation**

Expo Router inahitaji \_layout.jsx kama navigation container.  
Tengeneza file jipya:

app/\_layout.jsx:

import { Stack } from "expo-router";

export default function Layout() {

return (

<Stack>

<Stack.Screen name="index" options={{ title: "Home" }} />

</Stack>

);

}

Hii inachukua nafasi ya NavigationContainer uliyoitumia kwenye React Navigation.

**6️⃣ Kubadilisha Screens**

Kama ulikuwa na folder screens/ kwa SDK 51, sasa hamisha kila screen kwenye app/.

**Mfano wa zamani (SDK 51):**

screens/

HomeScreen.js

ProfileScreen.js

**Badilika (SDK 52):**

app/

index.jsx // HomeScreen

profile.jsx // ProfileScreen

👉 Ukihitaji dynamic routes (mfano profile na id):

app/

profile/

[id].jsx

**7️⃣ Kusafisha React Navigation**

Kama hutaki kubaki na React Navigation (kwa sababu Expo Router tayari inashughulikia navigation), toa packages zake:

npm uninstall @react-navigation/native @react-navigation/stack

Lakini kama unataka kuendelea nazo, unaweza — Expo Router ni optional, ila recommended.

**8️⃣ Test Project**

Sasa run project yako:

npx expo start

Ikiwa structure iko sahihi, utaona app yako inafunguka bila shida, ila navigation itakuwa kupitia Expo Router.

**🔑 Mfano: Migration ya Haraka**

**Kabla (SDK 51)**

App.js:

import React from "react";

import { NavigationContainer } from "@react-navigation/native";

import { createStackNavigator } from "@react-navigation/stack";

import HomeScreen from "./screens/HomeScreen";

import ProfileScreen from "./screens/ProfileScreen";

const Stack = createStackNavigator();

export default function App() {

return (

<NavigationContainer>

<Stack.Navigator>

<Stack.Screen name="Home" component={HomeScreen} />

<Stack.Screen name="Profile" component={ProfileScreen} />

</Stack.Navigator>

</NavigationContainer>

);

}

**Baada (SDK 52)**

app/\_layout.jsx:

import { Stack } from "expo-router";

export default function Layout() {

return (

<Stack>

<Stack.Screen name="index" options={{ title: "Home" }} />

<Stack.Screen name="profile" options={{ title: "Profile" }} />

</Stack>

);

}

app/index.jsx:

import { View, Text, Button } from "react-native";

import { Link } from "expo-router";

export default function HomeScreen() {

return (

<View>

<Text>Home Screen</Text>

<Link href="/profile">Go to Profile</Link>

</View>

);

}

app/profile.jsx:

import { View, Text } from "react-native";

export default function ProfileScreen() {

return (

<View>

<Text>Profile Screen</Text>

</View>

);

}

✅ Hapo sasa project yako ya SDK 51 imehamishwa SDK 52 bila kuanza from scratch.

Je unataka nikuandikie **migration guide** ya special case 👉 kama unataka **kuendelea kutumia React Navigation kwenye SDK 52 bila Expo Router** (kwa sababu hiyo pia inawezekana), au unataka tuendelee na Expo Router kama default?

Hivi naona kila tutorial za expo react native SDK 52 ni tofauti na zile za SDK 51 hata format ni tofauti naona sijui kwenye SDK 52 kuna mambo ya expo router, stack ni tofauti na 51, yani vitu viko tofauti kabisa. Sasa project iliyotengenezwa kwa expo react native SDK 51 huwezi kuitumia kwenye 52, maana naona huku kwenye 52 mafaili wanatumia jsx mimi kwenye 51 nilikuwa natumia js, kwahiyo huku siwezi kutumia mafaili ya js. Yani sielewi hebu nipe full explanations ya haya mabadiliko

**ChatGPT said:**

Umeuliza swali zuri sana 👍.  
Ni kweli kabisa **Expo SDK 52** inaleta mabadiliko makubwa ukilinganisha na **SDK 51** na versions zilizopita. Hii ndio sababu unaona tutorials za zamani haziendani na zile mpya. Hebu nikuchambulie hatua kwa hatua ili uelewe vizuri:

**🔑 1. Expo Router na App Structure**

* **SDK 51 na nyuma yake**: Mara nyingi tulikuwa tunatumia App.js kama entry point, kisha tukajipangia navigation kwa kutumia react-navigation.
* **SDK 52**: Sasa default inakuja na **Expo Router** ambayo imechukua nafasi ya react-navigation kwenye project mpya.
  + Hii inamaanisha navigation sasa inategemea **file-system based routing** (mfano app/index.jsx, app/profile/[id].jsx, nk).
  + Badala ya kuandaa navigator kwa mkono, router inachukua majina ya files na folders kama route zako.

**🔑 2. File Extensions (.js vs .jsx)**

* **SDK 51**: Wengi walitumia .js hata kwa components zenye JSX. Hii haikuwa shida.
* **SDK 52**: Convention mpya inapendekeza kutumia .jsx au .tsx (kama unatumia TypeScript) kwa files zenye JSX.
  + Kwa nini? Ili editor (VSCode, WebStorm nk) na bundler zitambue wazi kuwa file lina JSX.
  + Hata hivyo, **bado unaweza kutumia .js** — sio kosa. Ni kwamba tutorials mpya zinahimiza .jsx kwa clarity na standard mpya.

**🔑 3. Navigation Format (Stack, Tabs, Drawer)**

* **SDK 51**: Ili kuunda navigation ulitumia createStackNavigator, createBottomTabNavigator, nk kutoka @react-navigation/\*.
* **SDK 52**: Sasa unatumia components za Expo Router kama:
  + <Stack />
  + <Tabs />
  + <Drawer />  
    Zimejengwa ndani ya Expo Router, na hazihitaji configuration ngumu kama zamani.

Mfano:

// SDK 52 structure

import { Stack } from "expo-router";

export default function Layout() {

return (

<Stack>

<Stack.Screen name="index" />

<Stack.Screen name="profile/[id]" />

</Stack>

);

}

**🔑 4. Project Structure**

* **SDK 51**:
* App.js
* screens/
* components/
* navigation/
* **SDK 52** (default Expo Router):
* app/
* \_layout.jsx // layout ya navigation
* index.jsx // home screen
* profile/
* [id].jsx // dynamic route
* components/

👉 Hii ndio inakufanya uone tutorials mpya ziko tofauti kabisa na za zamani.

**🔑 5. Migration**

* Project ya **SDK 51 haiwezi ku-run moja kwa moja kwenye 52** bila adjustments kwa sababu ya hizi tofauti kwenye structure na navigation.
* Lakini unaweza kuendelea kutumia:
  + **React Navigation kwenye SDK 52** kama hutaki Expo Router (inahitaji ku-install kwa mkono).
  + Au kubadilisha project yako kuendana na Expo Router.

**🔑 6. Kwa Nini Wamebadili?**

* Expo Router inaleta experience karibu sana na frameworks kama **Next.js** (file-based routing).
* Inarahisisha mambo kama:
  + Dynamic routes (/profile/[id])
  + Nested layouts
  + Automatic linking
* Ni future direction ya Expo, ndio maana tutorials mpya zote ziko hivyo.

**✅ Conclusion**

* SDK 52 **ina support ya .js bado**, ila convention mpya inasisitiza .jsx / .tsx.
* Mabadiliko makubwa yapo kwenye **navigation** (Expo Router badala ya React Navigation).
* Project ya SDK 51 unaweza ku-update, ila itahitaji adjustments (hasa kwenye routing).

👉 Swali: Unapenda nikuandikie **njia ya kubadilisha project yako ya SDK 51 kwenda SDK 52** bila kuanzisha from scratch (migration guide), au ungependa nikupe **mfano mpya wa project ya SDK 52** ili uone structure yake vizuri?

**You said:**

Naomba niandikie njia ya kubadilisha project ya SDK 51 kwenda SDK 52 bila kuanzisha from scratch

**ChatGPT said:**

Vizuri sana 👌 umeuliza pointi ya msingi.  
Kubadilisha project kutoka **Expo SDK 51 → SDK 52** bila kuanza upya inaitwa **migration**. Hapa nitakupa hatua zote unazopaswa kufanya.

**🚀 Hatua za Kubadilisha Project SDK 51 → SDK 52**

**1️⃣ Update Expo SDK**

Kwanza hakikisha uko kwenye Expo SDK 52.  
Kwenye terminal ya project yako:

npx expo install expo@^52.0.0

Kisha update dependencies zingine:

npx expo install

**2️⃣ Badilisha package.json**

Angalia package.json yako, hakikisha inaonyesha:

"expo": "^52.0.0"

Halafu fanya install upya dependencies:

npm install

au

yarn install

**3️⃣ Kuandaa Expo Router**

Kwa sababu Expo Router sasa ndio default navigation, install:

npx expo install expo-router

Kisha fungua **package.json** na ongeza script hii kama haipo:

"main": "expo-router/entry"

**4️⃣ Badilisha Structure ya Files**

SDK 51 ulikuwa na App.js kama entry point.  
SDK 52 inatumia **folder app/**.

➡️ Hatua ya kufanya:

1. Tengeneza folder jipya:
2. app/
3. Hamisha App.js → ndani ya app/ na rename kuwa index.jsx

Mfano:

app/

index.jsx // ilikuwa App.js zamani

components/

⚠️ Usijali kuhusu extension .js — unaweza kubaki nayo, ila inapendekezwa index.jsx.

**5️⃣ Tengeneza Layout kwa Navigation**

Expo Router inahitaji \_layout.jsx kama navigation container.  
Tengeneza file jipya:

app/\_layout.jsx:

import { Stack } from "expo-router";

export default function Layout() {

return (

<Stack>

<Stack.Screen name="index" options={{ title: "Home" }} />

</Stack>

);

}

Hii inachukua nafasi ya NavigationContainer uliyoitumia kwenye React Navigation.

**6️⃣ Kubadilisha Screens**

Kama ulikuwa na folder screens/ kwa SDK 51, sasa hamisha kila screen kwenye app/.

**Mfano wa zamani (SDK 51):**

screens/

HomeScreen.js

ProfileScreen.js

**Badilika (SDK 52):**

app/

index.jsx // HomeScreen

profile.jsx // ProfileScreen

👉 Ukihitaji dynamic routes (mfano profile na id):

app/

profile/

[id].jsx

**7️⃣ Kusafisha React Navigation**

Kama hutaki kubaki na React Navigation (kwa sababu Expo Router tayari inashughulikia navigation), toa packages zake:

npm uninstall @react-navigation/native @react-navigation/stack

Lakini kama unataka kuendelea nazo, unaweza — Expo Router ni optional, ila recommended.

**8️⃣ Test Project**

Sasa run project yako:

npx expo start

Ikiwa structure iko sahihi, utaona app yako inafunguka bila shida, ila navigation itakuwa kupitia Expo Router.

**🔑 Mfano: Migration ya Haraka**

**Kabla (SDK 51)**

App.js:

import React from "react";

import { NavigationContainer } from "@react-navigation/native";

import { createStackNavigator } from "@react-navigation/stack";

import HomeScreen from "./screens/HomeScreen";

import ProfileScreen from "./screens/ProfileScreen";

const Stack = createStackNavigator();

export default function App() {

return (

<NavigationContainer>

<Stack.Navigator>

<Stack.Screen name="Home" component={HomeScreen} />

<Stack.Screen name="Profile" component={ProfileScreen} />

</Stack.Navigator>

</NavigationContainer>

);

}

**Baada (SDK 52)**

app/\_layout.jsx:

import { Stack } from "expo-router";

export default function Layout() {

return (

<Stack>

<Stack.Screen name="index" options={{ title: "Home" }} />

<Stack.Screen name="profile" options={{ title: "Profile" }} />

</Stack>

);

}

app/index.jsx:

import { View, Text, Button } from "react-native";

import { Link } from "expo-router";

export default function HomeScreen() {

return (

<View>

<Text>Home Screen</Text>

<Link href="/profile">Go to Profile</Link>

</View>

);

}

app/profile.jsx:

import { View, Text } from "react-native";

export default function ProfileScreen() {

return (

<View>

<Text>Profile Screen</Text>

</View>

);

}

✅ Hapo sasa project yako ya SDK 51 imehamishwa SDK 52 bila kuanza from scratch.

Je unataka nikuandikie **migration guide** ya special case 👉 kama unataka **kuendelea kutumia React Navigation kwenye SDK 52 bila Expo Router** (kwa sababu hiyo pia inawezekana), au unataka tuendelee na Expo Router kama default?

Pole sana kwa kutoelewana awali—umenielekeza wazi unataka **Expo Router (SDK 52)**, si React Navigation. Sawa, twende hatua kwa hatua kubadili **structure yako ya zamani (App → Drawer → Stack → Screens)** kwenda **Expo Router** huku ukihifadhi “feel” ile ile uliyokuwa nayo. Nakuandikia **maelezo + full codes** za kuanzia—ukibandika hizi zita-run bila kulazimika kubadili kila screen mara moja.

**1) Install & setup Expo Router**

npx expo install expo-router

Kisha kwenye **package.json** hakikisha:

{

"main": "expo-router/entry"

}

Usitumie tena NavigationContainer wala @react-navigation/\* moja kwa moja. Expo Router hutengeneza hiyo chini ya kapeti.

**2) Badili project structure kuwa file-based routing**

Unda structure hii (unaweza kuhamisha/kubandika existing screens zako polepole):

app/

\_layout.jsx // Drawer (root)

payment.jsx // item ya Drawer (Payment)

(main)/ // kundi la screens zako zote

\_layout.jsx // Stack ya (main)

index.jsx // Home (badala ya HomeStack)

welcome-screen.jsx // Welcome one-time

signup-stack.jsx

signin-stack.jsx

send-sms.jsx

wateja-wote.jsx

view-mteja.jsx

change-password.jsx

get-all-products.jsx

add-new-product.jsx

all-chatting.jsx

view-product.jsx

payment-info.jsx

components/

Header.jsx // Header yako ya zamani (imeboreshwa kwa Router)

UserContext.js // kama ulivyokuwa nao

Constant/links.js // kama ulivyokuwa nao

**3) Drawer (root) – app/\_layout.jsx**

Hii inachukua nafasi ya App.js + MyDrawer. Pia hapa tunaweka **UserProvider** ili context ibaki global kama zamani.

// app/\_layout.jsx

import React from 'react';

import { Slot } from 'expo-router';

import { Drawer } from 'expo-router/drawer';

import { UserProvider } from '../UserContext';

export default function RootLayout() {

return (

<UserProvider>

<Drawer

screenOptions={{

headerShown: false,

drawerStyle: { backgroundColor: '#11998e', width: 300 },

drawerLabelStyle: { color: 'white' },

overlayColor: '#1f1f1f',

}}

>

{/\* Kundi kuu la app yako (lenye Stack) \*/}

<Drawer.Screen

name="(main)"

options={{

drawerLabel: 'Mwanzo',

title: 'Mwanzo',

}}

/>

{/\* Payment kama Drawer item tofauti (kama zamani) \*/}

<Drawer.Screen

name="payment"

options={{

drawerLabel: 'Payment',

title: 'Payment',

}}

/>

</Drawer>

{/\* Slot ni sehemu ya kurender child routes kama payment.jsx \*/}

<Slot />

</UserProvider>

);

}

**4) Stack ya (main) – app/(main)/\_layout.jsx**

Hapa ndipo tunarudia “Stack Navigator” yako ya zamani, lakini kwa **expo-router/Stack**. Pia tunashughulikia **Welcome Screen one-time** (AsyncStorage) kwa redirect ya haraka.

// app/(main)/\_layout.jsx

import React, { useEffect, useState } from 'react';

import { Stack, useRouter } from 'expo-router';

import AsyncStorage from '@react-native-async-storage/async-storage';

import { ActivityIndicator, View } from 'react-native';

export default function MainLayout() {

const router = useRouter();

const [checking, setChecking] = useState(true);

useEffect(() => {

(async () => {

try {

const seen = await AsyncStorage.getItem('isAppFirstLaunched');

if (seen == null) {

await AsyncStorage.setItem('isAppFirstLaunched', 'false');

// on first launch nenda Welcome

router.replace('/(main)/welcome-screen');

}

} finally {

setChecking(false);

}

})();

}, []);

if (checking) {

return (

<View style={{ flex: 1, alignItems: 'center', justifyContent: 'center' }}>

<ActivityIndicator />

</View>

);

}

return (

<Stack

screenOptions={{

headerShown: false, // tunatumia Header yetu custom

}}

>

{/\* Hapa hutaji kutaja kila screen; files kwenye (main) zinakuwa routes automatically \*/}

<Stack.Screen name="index" />

<Stack.Screen name="welcome-screen" />

<Stack.Screen name="signup-stack" />

<Stack.Screen name="signin-stack" />

<Stack.Screen name="send-sms" />

<Stack.Screen name="wateja-wote" />

<Stack.Screen name="view-mteja" />

<Stack.Screen name="change-password" />

<Stack.Screen name="get-all-products" />

<Stack.Screen name="add-new-product" />

<Stack.Screen name="all-chatting" />

<Stack.Screen name="view-product" />

<Stack.Screen name="payment-info" />

</Stack>

);

}

Kumbuka: **jina la file = route name**. Mfano get-all-products.jsx route yake ni /(main)/get-all-products.

**5) Header yako – components/Header.jsx**

* Kufungua drawer: **dispatch DrawerActions.openDrawer()**
* Kwenda Add New Product: **router.push('/(main)/add-new-product')**

// components/Header.jsx

import React from 'react';

import { View, Text, TouchableOpacity } from 'react-native';

import { useRouter } from 'expo-router';

import { useNavigation, DrawerActions } from '@react-navigation/native';

import { FontAwesome } from '@expo/vector-icons';

export default function Header({ title }) {

const router = useRouter();

const navigation = useNavigation();

const openMenu = () => navigation.dispatch(DrawerActions.openDrawer());

const GoHome = () => router.push('/(main)/add-new-product');

return (

<View style={{ backgroundColor: '#015d68', marginBottom: 10 }}>

<View

style={{

paddingHorizontal: 12,

paddingVertical: 10,

flexDirection: 'row',

alignItems: 'center',

justifyContent: 'space-between',

}}

>

<TouchableOpacity onPress={openMenu}>

<View>

<View style={{ width: 20, height: 3, marginVertical: 5, backgroundColor: 'white' }} />

<View style={{ width: 15, height: 3, backgroundColor: 'white' }} />

<View style={{ width: 10, height: 3, marginVertical: 5, backgroundColor: 'white' }} />

</View>

</TouchableOpacity>

<Text style={{ fontSize: 16, color: 'white' }}>{title || 'TW 👩 CC'}</Text>

<TouchableOpacity onPress={GoHome}>

<FontAwesome name="calendar-plus-o" size={26} color="white" />

</TouchableOpacity>

</View>

</View>

);

}

**6) Home (index) – app/(main)/index.jsx**

Badala ya “Home Stack” ya zamani, **index.jsx** ni route ya /.

// app/(main)/index.jsx

import React from 'react';

import { View, Text } from 'react-native';

import Header from '../../components/Header';

export default function HomeScreen() {

return (

<View style={{ flex: 1 }}>

<Header title="Home" />

<View style={{ padding: 16 }}>

<Text>Karibu Home (Expo Router)</Text>

</View>

</View>

);

}

**7) Welcome (one-time) – app/(main)/welcome-screen.jsx**

// app/(main)/welcome-screen.jsx

import React from 'react';

import { View, Text, Button } from 'react-native';

import Header from '../../components/Header';

import { useRouter } from 'expo-router';

export default function WelcomeScreen() {

const router = useRouter();

return (

<View style={{ flex: 1 }}>

<Header title="Welcome" />

<View style={{ padding: 16 }}>

<Text>Karibu! (itaonekana mara ya kwanza tu)</Text>

<Button title="Endelea" onPress={() => router.replace('/(main)')} />

</View>

</View>

);

}

**8) Get All Products – app/(main)/get-all-products.jsx**

**JINSI YA KUNAVIGATE “exactly kama zamani”**:  
Badala ya navigation.navigate('View Product', item) tumia **router.push({ pathname, params })**.  
Kisha upande wa “View Product” **useLocalSearchParams()** kusoma params.

// app/(main)/get-all-products.jsx

import React, { useEffect, useState, useCallback } from 'react';

import { View, Text, FlatList, TouchableOpacity, ActivityIndicator, StyleSheet } from 'react-native';

import Header from '../../components/Header';

import { useRouter } from 'expo-router';

import { EndPoint } from '../../Constant/links';

export default function GetAllProducts() {

const router = useRouter();

const [queryset, setQueryset] = useState([]);

const [loading, setLoading] = useState(false);

const [endReached, setEndReached] = useState(false);

const [current\_page, setCurrentPage] = useState(1);

const getItems = useCallback(() => {

if (endReached || loading) return;

setLoading(true);

const url = `${EndPoint}/GetAllDukaLakoView/?page=${current\_page}&page\_size=50`;

fetch(url)

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

.then((data) => {

const items = data?.queryset ?? [];

if (items.length > 0) {

setQueryset((prev) => [...prev, ...items]);

setCurrentPage((p) => p + 1);

} else {

setEndReached(true);

}

})

.finally(() => setLoading(false));

}, [endReached, loading, current\_page]);

useEffect(() => {

getItems();

}, []); // load once

const CartCard = ({ item }) => (

<View style={styles.card}>

<TouchableOpacity

style={styles.button}

onPress={() =>

router.push({

pathname: '/(main)/view-product',

// ⚠️ Params hupitia kwenye URL; kama object ni kubwa sana, tumia { id: item.id } kisha ufetche tena

params: { ...item },

})

}

>

<Text style={styles.buttonText}>Tazama</Text>

</TouchableOpacity>

</View>

);

return (

<View style={{ flex: 1 }}>

<Header title="Historia Zako" />

{queryset.length > 0 ? (

<FlatList

data={queryset}

renderItem={CartCard}

keyExtractor={(it, idx) => String(it?.id ?? idx)}

showsVerticalScrollIndicator={false}

ListFooterComponent={loading ? <ActivityIndicator /> : null}

onEndReached={getItems}

onEndReachedThreshold={0.5}

/>

) : loading ? (

<ActivityIndicator />

) : (

<Text style={{ textAlign: 'center', marginTop: 20 }}>Hakuna data</Text>

)}

</View>

);

}

const styles = StyleSheet.create({

card: {

padding: 10,

marginHorizontal: 12,

marginVertical: 8,

backgroundColor: '#fff',

borderRadius: 8,

elevation: 2,

},

button: {

backgroundColor: 'blue',

padding: 10,

alignItems: 'center',

borderRadius: 6,

},

buttonText: { color: 'white' },

});

**9) View Product – app/(main)/view-product.jsx**

Soma params kwa useLocalSearchParams(). Ukipeleka id tu, unaweza kufetcha details hapa.

// app/(main)/view-product.jsx

import React from 'react';

import { View, Text } from 'react-native';

import Header from '../../components/Header';

import { useLocalSearchParams } from 'expo-router';

export default function ViewProduct() {

const params = useLocalSearchParams(); // hapa utapata fields ulizotuma

// const { id, Title } = params; // mfano ukituma id/title tu

return (

<View style={{ flex: 1 }}>

<Header title="View Product" />

<View style={{ padding: 16 }}>

<Text style={{ fontSize: 18, marginBottom: 8 }}>Product Details</Text>

<Text>ID: {String(params?.id ?? '-')}</Text>

<Text>Title: {params?.Title ?? '—'}</Text>

{/\* Ongeza fields nyingine kulingana na ulivyotuma \*/}

</View>

</View>

);

}

**10) Payment (Drawer item) – app/payment.jsx**

// app/payment.jsx

import React from 'react';

import { View, Text } from 'react-native';

import Header from '../components/Header';

export default function PaymentScreen() {

return (

<View style={{ flex: 1 }}>

<Header title="Payment" />

<View style={{ padding: 16 }}>

<Text>Payment Screen (Drawer item)</Text>

</View>

</View>

);

}

**11) Mapping ya majina ya zamani → routes mpya**

| **Zamani (React Navigation name)** | **Sasa (Expo Router pathname)** |
| --- | --- |
| "Welcome Screen" | /(main)/welcome-screen |
| "Signup Stack" | /(main)/signup-stack |
| "Signin Stack" | /(main)/signin-stack |
| "Send SMS" | /(main)/send-sms |
| "Wateja Wote" | /(main)/wateja-wote |
| "View Mteja" | /(main)/view-mteja |
| "Badili Password" | /(main)/change-password |
| "Home Stack" | /(main)/index |
| "Get All Products" | /(main)/get-all-products |
| "Add New Product" | /(main)/add-new-product |
| "All Chatting" | /(main)/all-chatting |
| "View Product" | /(main)/view-product |
| "Payment Screen" (Drawer) | /payment |
| "Payment Info" | /(main)/payment-info |

**Navigate “kama zamani”:**

* Zamani: navigation.navigate('View Product', item)
* Sasa:
* import { useRouter } from 'expo-router';
* const router = useRouter();
* router.push({ pathname: '/(main)/view-product', params: { ...item } });
* Kusoma params upande wa pili: useLocalSearchParams()

**Tahadhari ndogo:** URL search params zina limit; kama item ni object kubwa, ni salama zaidi kutuma { id: item.id } halafu u-fetche details ndani ya view-product.

**12) Mambo muhimu ya kukumbuka**

* **Usiweke NavigationContainer** mahali popote. Expo Router inakushikia hilo.
* **Header ya React Navigation** tumeizima kwa headerShown:false; tumebakiza **Header yako custom** kila screen (kama zamani).
* **Kufungua Drawer** kutoka Header.jsx: tumetumia navigation.dispatch(DrawerActions.openDrawer()) – fully supported.
* **UserProvider** tumeiweka juu kabisa app/\_layout.jsx ili i-cover app yote kama zamani.
* **AsyncStorage first-launch** sasa inafanya **redirect** kwenda /(main)/welcome-screen mara ya kwanza tu.

Ikiwa unataka nikubandikie pia **stubs** za signin-stack.jsx, signup-stack.jsx, n.k. (zilizo tayari na <Header /> juu) nitaweka zote. Pia nikipew­a **names halisi za props kwenye item** ninakuandikia **view-product.jsx** ifanye fetching kwa {id} badala ya kubeba object lote kwenye URL.

Tena pole kwa kuto-elewana mwanzo. Hapo juu ni **mpango kamili wa migration→ Expo Router (SDK 52)** uki-preserve structure na UX uliyozoea.

**You said:**

sasa kama hapa natumia hivi kunavigate 'import { useRouter } from 'expo-router'; const router = useRouter(); router.push({ pathname: '/(main)/view-product', params: { ...item } });', je vipi huku hii screen ya view-product? ambayo codes zake hapo mwanzo zilikuwa zipo hivi je itakuwaje kwa expo router?; const {width, height} = Dimensions.get('window'); export default function ViewProduct ({navigation, route}) { const { Title, id, Maelezo, LevelImage, phone, username, Location, Created, PichaYaPost, PichaYaPost2, PichaYaPost3, PichaYaPost4, PichaYaPost5, profile\_image, Likes, UserRole, TickStatus, email } = route.params return ( {/\*mwanzo wa Item View\*/} <View style={globalStyles.AppFlatListContainerYoutubeChannel} > <CustomCard > <View style={globalStyles.AppItemContainerYoutubeChannel} > <View style={{ //justifyContent:"space-between", }}> <Text style={[globalStyles.AppItemNameYoutubeChannel,{ color:'white', }]} >{Title}</Text> ); } const styles = StyleSheet.create({ }); na vipi kwenye hii MinorHeader.js kuna button ya user kugo back, je nafanyaje ili niweze kuendana na expo-router? MinorHeader codes are; import {View,StyleSheet,Image,Text,TouchableOpacity,FlatList,Dimensions} from 'react-native'; // import {WalletCoinCard} from './WalletCoinCard'; // import {CoinCard} from './CoinCard'; import { useNavigation } from '@react-navigation/native'; import {MaterialIcons,Entypo,MaterialCommunityIcons,FontAwesome5, Ionicons,Feather,AntDesign, FontAwesome} from '@expo/vector-icons'; import React, {useState, useEffect, useContext} from 'react'; import {globalStyles} from '../Styles/GlobalStyles'; import {useFonts} from 'expo-font'; export default function MinorHeader( {title} ) { let [fontsLoaded] = useFonts({ 'Bold': require('../assets/fonts/Poppins-Bold.ttf'), 'Medium': require('../assets/fonts/Poppins-Medium.ttf'), 'SemiBold': require('../assets/fonts/Poppins-SemiBold.ttf'), 'Regular': require('../assets/fonts/Poppins-Regular.ttf'), 'Thin': require('../assets/fonts/Poppins-Thin.ttf'), 'Light': require('../assets/fonts/Poppins-Light.ttf'), }); const navigation = useNavigation(); const goBackPage = () => { //navigation.navigate(screenName); navigation.goBack(); } const GoHome = () => { navigation.navigate('Get All Products'); } const [greeting, setGreeting] = useState(''); // Function to get the current time and set the greeting based on the time const setGreetingBasedOnTime = () => { const currentHour = new Date().getHours(); if (currentHour >= 5 && currentHour < 12) { setGreeting('GOOD MORNING'); } else if (currentHour >= 12 && currentHour <= 15) { setGreeting('GOOD AFTERNOON'); } else if (currentHour > 15 && currentHour <= 18) { setGreeting('GOOD EVENING'); } else { setGreeting('GOOD NIGHT'); } }; // Use useEffect to set the initial greeting and update it when needed useEffect(() => { setGreetingBasedOnTime(); }, []); return ( <>{!fontsLoaded ? (<View/>):( <View style={{ backgroundColor:'#015d68', marginBottom: 10, }}> <View style={styles.headerbar}> <TouchableOpacity onPress={goBackPage} > <Ionicons name='arrow-back' size={28} onPress={goBackPage} style={globalStyles.iconHeaderFile} color="white" /> </TouchableOpacity> <Text style={{fontSize:16, fontFamily:"Regular", color:'white'}}>TW 👩 CC</Text> <TouchableOpacity onPress={GoHome} > <FontAwesome name="home" size={26} color="white"/> </TouchableOpacity> </View> </View> )}</> ); }

**ChatGPT said:**

💯 umeuliza swali zuri sana — hii ndo sehemu ngumu kwa mtu anayetoka **React Navigation (SDK 51)** kwenda **Expo Router (SDK 52)**.  
Ngoja nikuonyeshe hatua kwa hatua jinsi ya kubadilisha:

**📌 1. Screen ya ViewProduct**

**Zamani (React Navigation / SDK 51)**

Ulikuwa unapata params kwa:

export default function ViewProduct({ navigation, route }) {

const { Title, id, ... } = route.params;

...

}

**Sasa (Expo Router / SDK 52)**

Expo Router haitumii tena navigation na route props.  
Badala yake inatumia **hooks** kutoka expo-router:

* useLocalSearchParams() 👉 badala ya route.params
* useRouter() 👉 badala ya navigation

import { View, Text, StyleSheet, Dimensions } from "react-native";

import { useLocalSearchParams } from "expo-router";

import { globalStyles } from "../Styles/GlobalStyles";

import CustomCard from "../components/CustomCard";

const { width, height } = Dimensions.get("window");

export default function ViewProduct() {

const {

Title,

id,

Maelezo,

LevelImage,

phone,

username,

Location,

Created,

PichaYaPost,

PichaYaPost2,

PichaYaPost3,

PichaYaPost4,

PichaYaPost5,

profile\_image,

Likes,

UserRole,

TickStatus,

email

} = useLocalSearchParams();

return (

<View style={globalStyles.AppFlatListContainerYoutubeChannel}>

<CustomCard>

<View style={globalStyles.AppItemContainerYoutubeChannel}>

<Text

style={[

globalStyles.AppItemNameYoutubeChannel,

{ color: "white" }

]}

>

{Title}

</Text>

</View>

</CustomCard>

</View>

);

}

const styles = StyleSheet.create({});

👉 Hapa sasa unapata params kupitia useLocalSearchParams moja kwa moja.

**📌 2. MinorHeader.js (Back Button + Home Button)**

**Zamani (React Navigation)**

Ulikuwa unatumia:

const navigation = useNavigation();

navigation.goBack();

navigation.navigate('Get All Products');

**Sasa (Expo Router)**

Badala yake, tunatumia **useRouter()**:

import { View, Text, TouchableOpacity, StyleSheet } from "react-native";

import { Ionicons, FontAwesome } from "@expo/vector-icons";

import { useRouter } from "expo-router";

import { useEffect, useState } from "react";

import { globalStyles } from "../Styles/GlobalStyles";

export default function MinorHeader({ title }) {

const router = useRouter();

const [greeting, setGreeting] = useState("");

const goBackPage = () => {

router.back(); // 🔑 badala ya navigation.goBack()

};

const GoHome = () => {

router.push("/"); // 🔑 badala ya navigation.navigate("Get All Products")

};

useEffect(() => {

const currentHour = new Date().getHours();

if (currentHour >= 5 && currentHour < 12) setGreeting("GOOD MORNING");

else if (currentHour >= 12 && currentHour <= 15) setGreeting("GOOD AFTERNOON");

else if (currentHour > 15 && currentHour <= 18) setGreeting("GOOD EVENING");

else setGreeting("GOOD NIGHT");

}, []);

return (

<View style={{ backgroundColor: "#015d68", marginBottom: 10 }}>

<View style={styles.headerbar}>

<TouchableOpacity onPress={goBackPage}>

<Ionicons

name="arrow-back"

size={28}

style={globalStyles.iconHeaderFile}

color="white"

/>

</TouchableOpacity>

<Text

style={{

fontSize: 16,

fontFamily: "Regular",

color: "white",

}}

>

{title ?? "TW 👩 CC"}

</Text>

<TouchableOpacity onPress={GoHome}>

<FontAwesome name="home" size={26} color="white" />

</TouchableOpacity>

</View>

</View>

);

}

const styles = StyleSheet.create({

headerbar: {

flexDirection: "row",

alignItems: "center",

justifyContent: "space-between",

padding: 10,

},

});

**🔑 Summary**

* **route.params → useLocalSearchParams()**
* **navigation.goBack() → router.back()**
* **navigation.navigate('Screen') → router.push('/screen')**

👉 Je unataka nikutengenezee **mfano wa folder structure (app/)** unaoonyesha jinsi view-product inavyokaa kwenye Expo Router (kwa mfano /app/(main)/view-product.jsx) ili uone full picture?

**You said:**

swali la mwisho sasa hii ndio ilikuwa packages.json file langu kwenye expo sdk 51 sasa sitak tena kutumia hiyo react navigation natumia expo-router kama nilivyosema, ondoa navigation packages zote ambazohazitumiki tena exposdk 52, then nipe fresh file ikiwa omeboresha nilichokwambia. my codes are; { "name": "agri-hub", "version": "1.0.3", "main": "expo/AppEntry.js", "scripts": { "start": "expo start", "android": "expo start --android", "ios": "expo start --ios", "web": "expo start --web" }, "dependencies": { "@react-native-async-storage/async-storage": "^2.0.0", "@react-native-community/masked-view": "^0.1.11", "@react-native-picker/picker": "^2.9.0", "@react-navigation/bottom-tabs": "^6.6.1", "@react-navigation/drawer": "^6.6.1", "@react-navigation/native": "^6.1.18", "@react-navigation/stack": "^6.4.1", "axios": "^1.7.7", "expo": "~51.0.28", "expo-app-loading": "^2.1.1", "expo-av": "^14.0.7", "expo-checkbox": "^3.0.0", "expo-constants": "^16.0.2", "expo-device": "^6.0.2", "expo-document-picker": "^12.0.2", "expo-file-system": "^17.0.1", "expo-font": "^12.0.10", "expo-image-picker": "^15.0.7", "expo-linear-gradient": "^13.0.2", "expo-location": "~17.0.1", "expo-notifications": "^0.28.19", "expo-print": "^13.0.1", "expo-sharing": "^12.0.1", "expo-splash-screen": "^0.27.6", "expo-status-bar": "~1.12.1", "lottie-react-native": "^7.0.0", "react": "18.2.0", "react-native": "0.74.5", "react-native-autocomplete-input": "^5.5.6", "react-native-awesome-alerts": "^2.0.0", "react-native-data-table": "^0.2.3", "react-native-date-picker": "^5.0.7", "react-native-event-listeners": "^1.0.7", "react-native-gesture-handler": "^2.20.2", "react-native-link": "^4.1.0", "react-native-maps": "1.14.0", "react-native-modern-datepicker": "^1.0.0-beta.91", "react-native-pdf": "^6.7.5", "react-native-reanimated": "^3.10.1", "react-native-render-html": "^6.3.4", "react-native-safe-area-context": "^4.13.1", "react-native-screens": "^3.35.0", "react-native-sweet-alert": "^3.5.0" }, "devDependencies": { "@babel/core": "^7.20.0" }, "private": true }

**ChatGPT said:**

👌 sawa nimekuelewa vizuri sana — unataka kuondoa **React Navigation** na mabaki yote yake (ambayo hayatumiki tena kwenye Expo SDK 52 kwa sababu unahamia kwenye expo-router) kisha nikutengenezee package.json mpya safi.

**📌 Hizi ndizo packages za React Navigation ambazo hazitakuhitajika tena:**

* @react-navigation/native
* @react-navigation/stack
* @react-navigation/bottom-tabs
* @react-navigation/drawer
* @react-native-community/masked-view (ilikuwa inahitajika kwa react-navigation)
* react-native-screens (hii inatumika zaidi na navigation)
* react-native-safe-area-context (pia navigation dependency)

👉 Kwa kuwa unatumia **Expo Router**, hizi zote tunaziondoa.

**📌 Fresh package.json (imeboreshwa kwa SDK 52 + Expo Router)**

{

"name": "agri-hub",

"version": "1.0.3",

"main": "expo/AppEntry.js",

"scripts": {

"start": "expo start",

"android": "expo start --android",

"ios": "expo start --ios",

"web": "expo start --web"

},

"dependencies": {

"@react-native-async-storage/async-storage": "^2.0.0",

"@react-native-picker/picker": "^2.9.0",

"axios": "^1.7.7",

"expo": "~52.0.0",

"expo-app-loading": "^2.1.1",

"expo-av": "^14.0.7",

"expo-checkbox": "^3.0.0",

"expo-constants": "^16.0.2",

"expo-device": "^6.0.2",

"expo-document-picker": "^12.0.2",

"expo-file-system": "^17.0.1",

"expo-font": "^12.0.10",

"expo-image-picker": "^15.0.7",

"expo-linear-gradient": "^13.0.2",

"expo-location": "~17.0.1",

"expo-notifications": "^0.28.19",

"expo-print": "^13.0.1",

"expo-router": "^3.0.0",

"expo-sharing": "^12.0.1",

"expo-splash-screen": "^0.27.6",

"expo-status-bar": "~1.12.1",

"lottie-react-native": "^7.0.0",

"react": "18.2.0",

"react-native": "0.74.5",

"react-native-autocomplete-input": "^5.5.6",

"react-native-awesome-alerts": "^2.0.0",

"react-native-data-table": "^0.2.3",

"react-native-date-picker": "^5.0.7",

"react-native-event-listeners": "^1.0.7",

"react-native-link": "^4.1.0",

"react-native-maps": "1.14.0",

"react-native-modern-datepicker": "^1.0.0-beta.91",

"react-native-pdf": "^6.7.5",

"react-native-reanimated": "^3.10.1",

"react-native-render-html": "^6.3.4",

"react-native-sweet-alert": "^3.5.0"

},

"devDependencies": {

"@babel/core": "^7.20.0"

},

"private": true

}

**🔑 Mambo ya Muhimu Uliyobadilishwa**

1. **Zimeondolewa kabisa**:
   * @react-navigation/\*
   * @react-native-community/masked-view
   * react-native-screens
   * react-native-safe-area-context
2. **Imeongezwa rasmi**:
   * "expo-router": "^3.0.0"
3. **Expo imepanda** kwenda ~52.0.0 (kwa sababu SDK mpya).

👉 Baada ya kubadilisha package.json, unafanya hivi:

rm -rf node\_modules package-lock.json

npm install

au kama unatumia yarn:

rm -rf node\_modules yarn.lock

yarn install

Unataka nikupangie pia **folder structure sahihi ya app/ kwa Expo Router SDK 52** (mfano (main)/view-product.jsx, (auth)/login.jsx) ili iwe rahisi ku-migrate screens zako zote?