

```
1: import 'dotenv/config';
2:
3: export default {
4:   expo: {
5:     name: 'Merzheial',
6:     plugins: [
7:       [
8:         "expo-image-picker",
9:         {
10:           photosPermission: "Allow Merzheial to access your photos",
11:           cameraPermission: "Allow Merzheial to access your camera"
12:         }
13:       ]
14:     ],
15:     slug: 'merzheial',
16:     privacy: 'public',
17:     platforms: ['ios', 'android'],
18:     version: '0.15.0',
19:     orientation: 'portrait',
20:     icon: './assets/icon.png',
21:     splash: {
22:       image: './assets/icon.png',
23:       resizeMode: 'cover',
24:       backgroundColor: '#F57C00'
25:     },
26:     updates: {
27:       fallbackToCacheTimeout: 0
28:     },
29:     assetBundlePatterns: ['**/*'],
30:     ios: {
31:       supportsTablet: true
32:     },
33:     extra: {
34:       apiKey: process.env.API_KEY,
35:       authDomain: process.env.AUTH_DOMAIN,
36:       projectId: process.env.PROJECT_ID,
37:       storageBucket: process.env.STORAGE_BUCKET,
38:       messagingSenderId: process.env.MESSAGING_SENDER_ID,
39:       appId: process.env.APP_ID
40:     }
41:   }
42: };
```



```
1: import { Images } from './images';  
2: import { Colors } from './theme';  
3: import { auth, db, storage } from './firebase';  
4:  
5: export { Images, Colors, auth, db, storage };
```


./config/theme.js

Mon Oct 25 15:08:00 2021

1

```
1: export const Colors = {  
2:   orange: '#f57c00',  
3:   blue: '#039be5',  
4:   black: '#222222',  
5:   white: '#ffffff',  
6:   mediumGray: '#6e6869',  
7:   red: '#fc5c65'  
8: };
```



```
1: import { initializeApp } from 'firebase/app';
2: import { getAuth } from 'firebase/auth';
3: import { initializeFirestore } from "firebase/firestore";
4: import { getStorage } from '@firebase/storage';
5: import Constants from 'expo-constants';
6:
7:
8: // add firebase config
9: const firebaseConfig = {
10:   apiKey: Constants.manifest.extra.apiKey,
11:   authDomain: Constants.manifest.extra.authDomain,
12:   projectId: Constants.manifest.extra.projectId,
13:   storageBucket: Constants.manifest.extra.storageBucket,
14:   messagingSenderId: Constants.manifest.extra.messagingSenderId,
15:   appId: Constants.manifest.extra.appId
16: };
17:
18: // initialize firebase
19: const firebaseApp = initializeApp(firebaseConfig);
20:
21: // initialize auth
22: const auth = getAuth();
23:
24: // bug with react-native work only react
25: // const db = getFirestore();
26:
27: // fix https://github.com/firebase/firebase-js-sdk/issues/5667#issuecomment-95207960
0
28: const db = initializeFirestore(firebaseApp, {useFetchStreams: false})
29:
30: const storage = getStorage(firebaseApp);
31:
32: export { auth, db, storage };
```



```
1: export const Images = {  
2:   logo: require('../assets/icon.png'),  
3: };  
4:
```



```
1: import React, { useState, createContext } from 'react';
2:
3: export const AuthenticatedUserContext = createContext({});
4:
5: export const AuthenticatedUserProvider = ({ children }) => {
6:   const [user, setUser] = useState(null);
7:
8:   return (
9:     <AuthenticatedUserContext.Provider value={{ user, setUser }}>
10:       {children}
11:     </AuthenticatedUserContext.Provider>
12:   );
13: };
14:
```



```
1: import {
2:   AuthenticatedUserContext,
3:   AuthenticatedUserProvider
4: } from './AuthenticatedUserProvider';
5:
6: export { AuthenticatedUserContext, AuthenticatedUserProvider };
```



```
1: import React from 'react'
2: import { createStackNavigator } from '@react-navigation/stack'
3:
4: import { DoctorHomeScreen } from '../screens'
5: import PatientInfo from '../screens/doctor/patientData/PatientInfo'
6: import DoctorMeetItem from '../screens/doctor/patientData/DoctorMeetItem'
7: import PatientTests from '../screens/doctor/patientData/PatientTests'
8: import TestInfo from '../screens/doctor/patientData/TestInfo'
9:
10: const Stack = createStackNavigator()
11:
12: export default function DoctorHomeStack() {
13:   return (
14:     <Stack.Navigator screenOptions={{ headerShown: false }}>
15:       <Stack.Screen name='Home' component={DoctorHomeScreen} />
16:       <Stack.Screen name='Info' component={PatientInfo} />
17:       <Stack.Screen name='MeetInfo' component={DoctorMeetItem} />
18:       <Stack.Screen name='Tests' component={PatientTests} />
19:       <Stack.Screen name='TestInfo' component={TestInfo} />
20:     </Stack.Navigator>
21:   )
22: }
```



```
1: import React from 'react'
2: import { createStackNavigator } from '@react-navigation/stack'
3:
4: import ItemList from '../screens/patient/PatientListPage/ItemList'
5: import PatientList from '../screens/patient/PatientListPage/PatientList'
6: import CreateList from '../screens/patient/PatientListPage/CreateList'
7:
8: const Stack = createStackNavigator()
9:
10: export default function PatientListStack() {
11:   return (
12:     <Stack.Navigator screenOptions={{ headerShown: false }}>
13:       <Stack.Screen name='List' component={PatientList} />
14:       <Stack.Screen name='Item' component={ItemList} />
15:       <Stack.Screen name='Create_new' component={CreateList} />
16:     </Stack.Navigator>
17:   )
18: }
```



```
1: import React from 'react'
2: import { createStackNavigator } from '@react-navigation/stack'
3:
4: import PatientMeetItem from '../screens/patient/PatientMeetItem'
5: import { PatientStatScreen } from '../screens'
6:
7: const Stack = createStackNavigator()
8:
9: export default function PatientStatStack() {
10:   return (
11:     <Stack.Navigator screenOptions={{ headerShown: false }}>
12:       <Stack.Screen name='Stat' component={PatientStatScreen} />
13:       <Stack.Screen name='MeetDocDesc' component={PatientMeetItem} />
14:     </Stack.Navigator>
15:   )
16: }
```



```
1: import React, { useState, useContext, useEffect } from 'react';
2: import { NavigationContainer } from '@react-navigation/native';
3: import { onAuthStateChanged } from 'firebase/auth';
4:
5: import { AuthStack } from '../AuthStack';
6: import { AppStack } from '../AppStack';
7: import { AuthenticatedUserContext } from '../providers';
8: import { LoadingIndicator } from '../components';
9: import { auth } from '../config';
10:
11: export const RootNavigator = () => {
12:   const { user, setUser } = useContext(AuthenticatedUserContext);
13:   const [isLoading, setIsLoading] = useState(true);
14:
15:   useEffect(() => {
16:     // onAuthStateChanged returns an unsubscribe
17:     const unsubscribeAuthStateChanged = onAuthStateChanged(
18:       auth,
19:       authenticatedUser => {
20:         // cannot use recoil with onAuthStateChanged
21:         authenticatedUser ? setUser(authenticatedUser) : setUser(null);
22:         setIsLoading(false);
23:       }
24:     );
25:
26:     // unsubscribe auth listener on unmount
27:     return () => {
28:       unsubscribeAuthStateChanged();
29:     }
30:   }, [user]);
31:
32:   if (isLoading) {
33:     return <LoadingIndicator />;
34:   }
35:
36:   return (
37:     <NavigationContainer>
38:       {user ? <AppStack /> : <AuthStack />}
39:     </NavigationContainer>
40:   );
41: };
```



```

1: import React, { useEffect, useState } from 'react';
2: import { createBottomTabNavigator } from '@react-navigation/bottom-tabs';
3: import { Ionicons } from '@expo/vector-icons';
4: import { UserStack } from './UserStack';
5:
6: // caretaker screen
7: import CareTakerHomeStack from './CareTakerHomeStack';
8:
9: // doctor screen
10: import DoctorHomeStack from './DoctorHomeStack';
11:
12: // patient screen
13: import Game from '../screens/patient/game/picture_puzzle/Game';
14: import { PatientSocialScreen } from '../screens';
15: import { PatientTest2 } from '../screens/patient/Test2Page/PatientTest2';
16: import PatientListStack from './PatientListStack';
17: import PatientStatStack from './PatiantStatStack';
18: import { FamilyStack } from './PatientFamilyStack';
19:
20: import { userInfoAtom, dayAtom } from '../store';
21: import { useRecoilState } from 'recoil';
22: import { doc, onSnapshot } from "firebase/firestore";
23: import { auth, db } from '../config';
24: import { LoadingIndicator, View } from '../components';
25: import { Text } from 'react-native';
26:
27: const Tab = createBottomTabNavigator();
28:
29: export const AppStack = () => {
30:   const [userInfo, setUserInfo] = useRecoilState(userInfoAtom);
31:   const [curDay, setDay] = useRecoilState(dayAtom);
32:   const [isLoading, setIsLoading] = useState(true);
33:
34:   useEffect(() => {
35:     const unsubscribeUserInfo = onSnapshot(doc(db, "users", auth.currentUser.uid), (
doc) => {
36:       console.log("user Info data: ", doc.data());
37:       setUserInfo(doc.data());
38:       setIsLoading(false);
39:     }, (error) => {
40:       console.log("error user info", error);
41:     });
42:
43:     setDay(() => {
44:       var d = new Date()
45:       d.setHours(0)
46:       d.setMinutes(0)
47:       d.setSeconds(0)
48:       return d.getTime()
49:     })
50:
51:     var a = curDay
52:     console.log("curDay ==>", a.toString());
53:
54:     return () => {
55:       unsubscribeUserInfo();
56:     }
57:   }, [])
58:
59:   const renderIcons = ({ focused, color, size, route }) => {
60:     let iconName;
61:
62:     if (route.name === 'à,«à,\231à¹\211à,²à¹\201à,£à,\201') {
63:       iconName = focused ? 'ios-home' : 'ios-home-outline';
64:     } else if (route.name === 'à,²à,\226à,\225à,¹') {
65:       iconName = focused ? 'ios-today' : 'ios-today-outline';

```

```

66:   } else if (route.name === 'Social') {
67:     iconName = focused ? 'ios-chatbubble-ellipses' : 'ios-chatbubble-ellipses-outl
ine';
68:   } else if (route.name === 'à,\215à,²à,\225à,¹') {
69:     iconName = focused ? 'ios-people' : 'ios-people-outline';
70:   } else if (route.name === 'à,\234à,¹à¹\211à¹\203à,\212à¹\211') {
71:     iconName = focused ? 'ios-person' : 'ios-person-outline';
72:   } else if (route.name === 'Game') {
73:     iconName = focused ? 'game-controller' : 'game-controller-outline';
74:   }
75:
76:   // You can return any component that you like here!
77:   return <Ionicons name={iconName} size={size} color={color} />;
78: }
79:
80: if (isLoading) {
81:   return <LoadingIndicator />
82: }
83:
84: if (userInfo.person_type === 'doctor') {
85:   return (
86:     <Tab.Navigator
87:       screenOptions={({ route }) => ({
88:         tabBarIcon: ({ focused, color, size }) => {
89:           return renderIcons({ focused, color, size, route })
90:         },
91:         tabBarActiveTintColor: 'tomato',
92:         tabBarInactiveTintColor: 'gray',
93:         headerShown: false
94:       })
95:     >
96:     <Tab.Screen name='à,«à,\231à¹\211à,²à¹\201à,£à,\201' component={DoctorHomeSt
ack} />
97:     <Tab.Screen name='à,\234à,¹à¹\211à¹\203à,\212à¹\211' component={UserStack} /
>
98:   </Tab.Navigator>
99: );
100: } else if (userInfo.person_type === 'caretaker') {
101:   return (
102:     <Tab.Navigator
103:       screenOptions={({ route }) => ({
104:         tabBarIcon: ({ focused, color, size }) => {
105:           return renderIcons({ focused, color, size, route })
106:         },
107:         tabBarActiveTintColor: 'tomato',
108:         tabBarInactiveTintColor: 'gray',
109:         headerShown: false
110:       })
111:     >
112:     <Tab.Screen name='à,«à,\231à¹\211à,²à¹\201à,£à,\201' component={CareTakerHom
eStack} />
113:     <Tab.Screen name='à,\234à,¹à¹\211à¹\203à,\212à¹\211' component={UserStack} /
>
114:   </Tab.Navigator>
115: );
116: } else if (userInfo.person_type === 'patient' && !userInfo.isTest) {
117:   return (
118:     <PatientTest2 />
119:   )
120: } else if (userInfo.person_type === 'patient' && userInfo.isTest) {
121:   return (
122:     <Tab.Navigator
123:       initialRouteName="à,«à,\231à¹\211à,²à¹\201à,£à,\201"
124:       screenOptions={({ route }) => ({
125:         tabBarIcon: ({ focused, color, size }) => {
126:           return renderIcons({ focused, color, size, route })

```

```
127:         },
128:         tabBarActiveTintColor: 'tomato',
129:         tabBarInactiveTintColor: 'gray',
130:         headerShown: false
131:     )})
132: >
133:     {/* todo
134:      create web game and like to it
135:     */}
136:     <Tab.Screen name='Social' component={PatientSocialScreen}/>
137:     <Tab.Screen name='Game' component={Game}/>
138:     <Tab.Screen name='à,²à,\226à,´à,\225à,´' component={PatientStatStack}/>
139:     <Tab.Screen name='à,«à,\231à¹\211à,²à¹\201à,£à,\201' component={PatientListS
tack}/>
140:     <Tab.Screen name='à,\215à,²à,\225à,´' component={FamilyStack}/>
141:     <Tab.Screen name='à,\234à,¹à¹\211à¹\203à,\212à¹\211' component={UserStack}/>
142:   </Tab.Navigator>
143: );
144: }
145:
146: return (
147:   <View isSafe>
148:     <Text>Person type invalid!</Text>
149:   </View>
150: )
151: };
```



```
1: import React from 'react';
2: import { createStackNavigator } from '@react-navigation/stack';
3:
4: import PatientFamily from '../screens/patient/PatientFamily';
5: import PatientNewFP from '../screens/patient/PatientNewFP';
6:
7: const Stack = createStackNavigator();
8:
9: export const FamilyStack = () => {
10:   return (
11:     <>
12:       <Stack.Navigator screenOptions={{ headerShown: false }}>
13:         <Stack.Screen name='Family-lists' component={PatientFamily} />
14:         <Stack.Screen name='New-person' component={PatientNewFP} />
15:       </Stack.Navigator>
16:     </>
17:   );
18: };
```


./navigation/index.js

Mon Oct 25 15:08:00 2021

1

```
1: import { RootNavigator } from './RootNavigator';  
2:  
3: export { RootNavigator };
```



```
1: import * as React from 'react';
2: import { createStackNavigator } from '@react-navigation/stack';
3:
4: import { LoginScreen, SignupScreen, ForgotPasswordScreen } from '../screens';
5:
6: const Stack = createStackNavigator();
7:
8: export const AuthStack = () => {
9:   return (
10:     <Stack.Navigator
11:       initialRouteName='Login'
12:       screenOptions={{ headerShown: false }}
13:     >
14:       <Stack.Screen name='Login' component={LoginScreen} />
15:       <Stack.Screen name='Signup' component={SignupScreen} />
16:       <Stack.Screen name='ForgotPassword' component={ForgotPasswordScreen} />
17:     </Stack.Navigator>
18:   );
19: };
```



```
1: import React from 'react';
2: import { View } from '../components';
3: import { Text } from 'react-native';
4:
5: import { createStackNavigator } from '@react-navigation/stack';
6:
7: import { UserScreen, UpdateUser } from '../screens';
8: import { PatientTest2 } from '../screens/patient/Test2Page/PatientTest2';
9: import { PatientSimpleTest } from '../screens/patient/PatientSimpleTest';
10:
11: import { userInfoAtom } from '../store';
12: import { useRecoilValue } from 'recoil';
13:
14: const Stack = createStackNavigator();
15:
16: export const UserStack = () => {
17:   const userInfo = useRecoilValue(userInfoAtom);
18:
19:   return (
20:     <>
21:       {userInfo.person_type === 'patient' ?
22:         <Stack.Navigator screenOptions={{ headerShown: false }}>
23:           <Stack.Screen name='UserInfo' component={UserScreen} />
24:           <Stack.Screen name='UpdateUser' component={UpdateUser} />
25:           <Stack.Screen name='DoTestMOCA' component={PatientTest2} />
26:           <Stack.Screen name='DoSimpleTest' component={PatientSimpleTest} />
27:         </Stack.Navigator> :
28:         <Stack.Navigator screenOptions={{ headerShown: false }}>
29:           <Stack.Screen name='UserInfo' component={UserScreen} />
30:           <Stack.Screen name='UpdateUser' component={UpdateUser} />
31:         </Stack.Navigator>
32:       </>
33:     );
34:   };
```



```
1: import React from 'react'
2: import { createStackNavigator } from '@react-navigation/stack'
3:
4: import { CaretakerHomeScreen } from '../screens'
5: import CTakerPatientInfo from '../screens/caretaker/patientData/CTakerPatientInfo'
6: import CTakerMeetItem from '../screens/caretaker/patientData/CTakerMeetItem'
7: import PatientTests from '../screens/doctor/patientData/PatientTests'
8: import TestInfo from '../screens/doctor/patientData/TestInfo'
9: import CTakerPatientList from '../screens/caretaker/patientData/CTakerPatientList'
10: import CTakerItemList from '../screens/caretaker/patientData/CTakerItemList'
11:
12: const Stack = createStackNavigator()
13:
14: export default function CareTakerHomeStack() {
15:   return (
16:     <Stack.Navigator screenOptions={{ headerShown: false }}>
17:       <Stack.Screen name='Home' component={CaretakerHomeScreen} />
18:       <Stack.Screen name='Info' component={CTakerPatientInfo} />
19:       <Stack.Screen name='MeetInfo' component={CTakerMeetItem} />
20:
21:       {/* code same as doctor */}
22:       <Stack.Screen name='Tests' component={PatientTests} />
23:       <Stack.Screen name='TestInfo' component={TestInfo} />
24:
25:       {/* patient list */}
26:       <Stack.Screen name='List' component={CTakerPatientList} />
27:       <Stack.Screen name='Item' component={CTakerItemList} />
28:     </Stack.Navigator>
29:   )
30: }
```



```
1: import * as Yup from 'yup';
2:
3: export const loginValidationSchema = Yup.object().shape({
4:   email: Yup.string().required().email().label('Email'),
5:   password: Yup.string().required().min(6).label('Password')
6: });
7:
8: export const signupValidationSchema = Yup.object().shape({
9:   email: Yup.string().required().email().label('Email'),
10:  password: Yup.string().required().min(6).label('Password'),
11:  confirmPassword: Yup.string()
12:    .oneOf([Yup.ref('password')], 'Confirm Password must match password.')
13:    .required('Confirm Password is required.'),
14:  age: Yup.number().required(),
15:  person_type: Yup.string().required(),
16:  sex_type: Yup.string().required(),
17:  name: Yup.string().required(),
18:  lastname: Yup.string().required()
19: });
20:
21: export const updateUserSchema = Yup.object().shape({
22:   age: Yup.number().required(),
23:   sex_type: Yup.string().required(),
24:   name: Yup.string().required(),
25:   lastname: Yup.string().required()
26: });
27:
28: export const passwordResetSchema = Yup.object().shape({
29:   email: Yup.string()
30:     .required('Please enter a registered email')
31:     .label('Email')
32:     .email('Enter a valid email')
33: });
```


1

```

64: const patientsRef = collection(db, "users", auth.currentUser.uid, "all-patients")
65: const allpatients = await getDocs(patientsRef);
66: allpatients.docs.map(async (item) => {
67:   const patient_uid = item.data().uid
68:   const patientRef = doc(db, "users", patient_uid)
69:   const docsnap = await getDoc(patientRef)
70:   // console.log('docsnap', docsnap.data());
71:   setAllPatients((prev) => [...prev, {...docsnap.data()}]);
72: })
73: // console.log('refetch patients', allPatients);
74: setLoadingPatients(false)
75: }
76:
77: const loadOptions = async (inputValue) => {
78:   setLoading(true);
79:   setSearchPatient([]);
80:   // console.log('click query');
81:   const userRef = collection(db, "users");
82:   const q = query(userRef, orderBy("uid"), startAt(inputValue), endAt(inputValue+"
\u00ff"));
83:   const querySnapshot = await getDocs(q);
84:   const users = querySnapshot.docs.filter(doc => doc.data().person_type == "patient");
85:   // console.log('found user', users.map((doc) => doc.data()));
86:   setSearchPatient(users.map((doc) => {
87:     return {
88:       id: doc.id,
89:       ...doc.data()
90:     }
91:   }));
92:   setLoading(false);
93: }
94:
95: const checkSelectPatientInPatients = (uid) => {
96:   if (allPatients.length === 0) {
97:     return true
98:   }
99:
100:   const findPatient = allPatients.filter(e => e.uid === uid);
101:   // console.log("find patient ==>", findPatient);
102:
103:   if (findPatient.length === 0) {
104:     return true
105:   } else {
106:     return false
107:   }
108: }
109:
110: const addPatient = async (patient) => {
111:   // check patient that is not added in lists
112:   console.log('clicked patient ==> ', patient.firstname, '\n      uid ==> ', patient.uid);
113:   const r = checkSelectPatientInPatients(patient.uid)
114:   if (r === true) {
115:     const docRef = doc(db, "users", auth.currentUser.uid, "all-patients", patient.uid);
116:     await setDoc(docRef, {uid: patient.uid}, {merge: true}).then(() => {
117:       alertMessage("\u00a1\u200a,\u00c0,\u00e0\u207a,\u210a,\u00c0,\u00c0\u211a,\u231", "\u00c0,\u234a,\u00c0\u211a,\u233a\u210a,\u23a,\u226a,\u00c0,\u201a\u210a,\u200a,\u236a,\u00c0\u210a,\u00c0\u201a,\u00c0\u211a,\u231a,\u231a,\u00c0,\u231a,\u00c0\u211a");
118:     });
119:   } else {
120:     alertMessage("\u00c0,\u204a,\u00c0\u210a,\u00c0,\u00c0,\u00c0,\u226a\u200a,\u236a,\u00c0\u210a,\u00c0,\u234a,\u00c0\u211a,\u233a\u210a,\u00c0,\u00c0\u204a,\u224a\u211", "\u00c0,\u234a,\u00c0\u211a,\u233a\u210a,\u00c0,\u00c0,\u201a,\u201a\u210a,\u236a,\u00c0\u210a,\u00c0\u204a,\u233a\u210a,\u00c0,\u00c0\u211a,\u00c0,\u201a\u210a,\u20c0,\u231a,\u00c0,\u231a,\u00c0\u211a");

```

```
./screens/caretaker/CaretakerHomeScreen.js
```

Wed Jan 19 22:11:33 2022

2

[illegible]

```

181:         loadOptions(search)
182:     })
183: }
184:     <Text style={styles.textBtn}>à,\204à¹\211à,\231à,«à,²</Text>
185: </Pressable>
186: </View>
187: {isLoading ? <View style={styles.scrollView}>
188:     {searchPatient.length == 0 ? <Text>à¹\204à,;à¹\210à,\236à,\232à,\234à,¹à¹
\211à,\233à¹\210à,$à,çà,\227à,µà¹\210 uid à,\202à,¶à¹\211à,\231à,\225à¹\211à,\231à,\224à¹
\211à,$à,ç {search}</Text>
189:     : (
190:         <ScrollView>
191:             <View style={{height: 4, width: width*0.8, backgroundColor: '#f0f0f0'}}
192:             {searchPatient.map(patient => {
193:                 return <View
194:                     key={patient.uid}
195:                     style={[styles.item, styles.itemSearch]}
196:                 >
197:                     <Image
198:                         style={[styles.image, styles.imageSearch]}
199:                         source={{uri: patient.urlImage}}
200:                     />
201:                     <View style={styles.itemViewText}>
202:                         <Text style={styles.itemTitle}>{patient.firstname}</Text>
203:                         <Text style={styles.itemTitle}>{patient.lastname}</Text>
204:                     </View>
205:                     {checkSelectPatientInPatients(patient.uid) ?
206:                         <Pressable
207:                             style={[styles.buttonOpen, styles.button]}
208:                             onPress={() => {
209:                                 addPatient(patient)
210:                             }}
211:                         >
212:                             <Text style={styles.textBtn}>à¹\200à,\236à,¹à¹\210à,;à,
\204à,\231à¹\204à,\202à¹\211</Text>
213:                         </Pressable>
214:                         :
215:                         <Pressable
216:                             style={[styles.button, {backgroundColor: "#00af00"}]}
217:                             onPress={() => {
218:                                 // navigate to patientInfo
219:                                 gotoPatientInfo(patient)
220:                             }}
221:                         >
222:                             <Text style={styles.textBtn}>à¹\204à,\233à,\227à,µà¹\210à
,\202à¹\211à,-à,;à,¹à,¥</Text>
223:                         </Pressable>
224:                     </View>
225:                 }
226:             )
227:         }
228:     </ScrollView>
229: )
230: }
231: </View>: <View style={styles.scrollView}><LoadingIndicator/></View>
232: <View style={{height: 5, width: width, backgroundColor: '#1597e5'}}</View>
233: {!loadingPatients ?
234: <ScrollView
235:     refreshControl={
236:         <RefreshControl
237:             refreshing={refreshing}
238:             onRefresh={onRefresh}
239:         />
240:     }
241:     >

```

```
242:         <View style={{height: 4, width: width, backgroundColor: '#f0f0f0'}}></View>
>
243:         <RenderPatients/>
244:         </ScrollView>) : <LoadingIndicator/>
245:     </View>
246: );
247: };
248:
249: const styles = StyleSheet.create({
250:   container: {
251:     flex: 1,
252:     alignItems: 'center',
253:     backgroundColor: 'white',
254:     height: height
255:     // justifyContent: 'center'
256:   },
257:   scrollView: {
258:     width: width*0.8,
259:     minHeight: 70,
260:     maxHeight: 200,
261:     alignItems: 'center',
262:     justifyContent: "center",
263:     borderRadius: 5,
264:     backgroundColor: "#efefef",
265:     marginVertical: 12,
266:     // shadow
267:     shadowRadius: 2.62,
268:     shadowOpacity: 0.23,
269:     shadowColor: "#000",
270:     shadowOffset: {
271:       width: 0,
272:       height: 2
273:     },
274:     elevation: 4,
275:   },
276:   button: {
277:     justifyContent: 'center',
278:     width: 'auto',
279:     height: 50,
280:     borderRadius: 5,
281:     paddingHorizontal: 10,
282:   },
283:   buttonOpen: {
284:     backgroundColor: Colors.orange,
285:   },
286:   buttonSummit: {
287:     backgroundColor: '#1597e5',
288:   },
289:   textBtn: {
290:     alignSelf: 'center',
291:     fontWeight: 'bold',
292:     color: 'white'
293:   },
294:   textHeader: {
295:     fontWeight: 'bold',
296:     fontSize: 30,
297:     marginLeft: 20,
298:     marginBottom: 10
299:   },
300:   viewHeader: {
301:     width: width*0.9,
302:     flexDirection: 'row',
303:     justifyContent: 'space-evenly',
304:     alignItems: 'center',
305:     alignSelf: 'center'
306:   },
```

```
307:   titleInput:{
308:     height: 50,
309:     width: width*0.7,
310:     padding: 10,
311:     borderRadius: 5,
312:     fontSize: 20,
313:     backgroundColor: "#f7f7f7",
314:   },
315:   itemSearch:{
316:     width: width*0.8,
317:     justifyContent: 'space-evenly',
318:     alignItems: 'center',
319:   },
320:   itemPatient:{
321:     width: width,
322:   },
323:   item:{
324:     height: 'auto',
325:     paddingHorizontal: width * 0.05,
326:     borderBottomWidth: 4,
327:     borderColor: '#f0f0f0',
328:     paddingBottom: 5,
329:     flexDirection: 'row',
330:   },
331:   imageSearch: {
332:     height: 50,
333:     width: 50,
334:   },
335:   imagePatient: {
336:     height: 70,
337:     width: 70,
338:   },
339:   image: {
340:     borderRadius: 5,
341:     marginVertical: 12,
342:     alignSelf: 'flex-start'
343:   },
344:   itemTitle:{
345:     fontSize: 20,
346:     fontWeight: '600',
347:   },
348:   itemViewText: {
349:     alignSelf: 'center',
350:     paddingLeft: 10
351:   },
352: })
353:
```



```

60: const styles = StyleSheet.create({
61:   container: {
62:     flex: 1,
63:     alignItems: 'center',
64:     backgroundColor: 'white',
65:     height: height
66:     // justifyContent: 'center'
67:   },
68:   content: {
69:     width: width*0.8,
70:     alignItems: 'center'
71:   },
72:   button: {
73:     justifyContent: 'center',
74:     width: width*0.9,
75:     height: 50,
76:     borderRadius: 5,
77:     marginVertical: 10
78:   },
79:   buttonOpen: {
80:     backgroundColor: Colors.orange,
81:   },
82:   buttonSummit: {
83:     backgroundColor: '#1597e5',
84:   },
85:   textStyle: {
86:     fontWeight: "bold",
87:     fontSize: 20,
88:     textAlign: "left",
89:     alignSelf: 'flex-start'
90:   },
91:   textHeader: {
92:     fontWeight: 'bold',
93:     fontSize: 30,
94:     marginLeft: 20,
95:     marginBottom: 10
96:   },
97:   viewHeader: {
98:     width: width*0.9,
99:     flexDirection: 'row',
100:    justifyContent: 'flex-start',
101:    alignItems: 'center',
102:    alignSelf: 'center'
103:  },
104: })

```



```

1: import React from 'react'
2: import { StyleSheet, Text, Dimensions, ScrollView, Pressable } from 'react-native'
3: import { View } from '.././../components'
4: import { Colors } from '.././../config';
5: import { Ionicons } from '@expo/vector-icons';
6: const {width, height} = Dimensions.get('window');
7:
8: const CTakerMeetItem = ({navigation, route}) => {
9:
10:   const { data } = route.params
11:
12:   return (
13:     <ScrollView>
14:       <View isSafe style={styles.container}>
15:         <View style={styles.viewHeader}>
16:           <Pressable onPress={() => navigation.goBack()}>
17:             <Ionicons name='arrow-back-circle' size={30} color={Colors
.blue} />
18:           </Pressable>
19:           <Text style={styles.textHeader}>{data.title}</Text>
20:         </View>
21:         <View style={styles.content}>
22:           <Text style={[styles.textStyle, {fontWeight: '300'}]}>{(new Intl
.DateTimeFormat("th-TH",{ dateStyle: 'full', timeStyle: 'short' })).format((new Date(data.tim
e.seconds * 1000)))}.toString())</Text>
23:           <Text style={styles.textStyle}>à,£à,²à,çà,¥à,°à¹\200à,-à,µà,çà,
\224: {data.description}</Text>
24:           <Text style={[styles.textStyle, {color : (data.note.length > 0)
? "black" : "red"}]}>note à,\202à,-à,\207à¹\201à,\236à,\227à,çà¹\214: {(data.note.length > 0
) ? data.note : "à,£à,-à¹\201à,\236à,\227à,çà¹\214à,\233à,£à,°à¹\200à,¡à,´à,\223à,«à,¥à,±à,
\207à,\231à,±à,\224à,\236à,\232"}</Text>
25:         </View>
26:       </View>
27:     </ScrollView>
28:   )
29: }
30:
31: export default CTakerMeetItem
32:
33: const styles = StyleSheet.create({
34:   container: {
35:     flex: 1,
36:     alignItems: 'center',
37:     backgroundColor: 'white',
38:     minHeight: height
39:     // justifyContent: 'center'
40:   },
41:   content: {
42:     width: width*0.8,
43:     alignItems: 'center'
44:   },
45:   textStyle: {
46:     fontWeight: "bold",
47:     fontSize: 20,
48:     alignSelf: 'flex-start'
49:   },
50:   textHeader: {
51:     fontWeight: 'bold',
52:     fontSize: 30,
53:     marginLeft: 20,
54:     marginBottom: 10
55:   },
56:   viewHeader: {
57:     width: width*0.9,
58:     flexDirection: 'row',
59:     justifyContent: 'flex-start',
60:     alignItems: 'center',
61:     alignSelf: 'center'
62:   },
63: })
64:

```



```

1: import React, {useState, useEffect} from 'react'
2: import { StyleSheet, Text, Image, Dimensions, ScrollView, Pressable } from 'react-native'
3: import { View, Button } from '.././../components'
4: import { Colors, db } from '.././../config';
5: import { collection, query, where, onSnapshot, doc } from 'firebase/firestore';
6: import { useRecoilValue, useRecoilState } from 'recoil';
7: import { sortcurrentPatientMeetDocs, currentPatientMeetDocs } from '.././../store';
8: import { Ionicons } from '@expo/vector-icons';
9: const {width, height} = Dimensions.get('window');
10:
11: const CTakerPatientInfo = ({navigation, route}) => {
12:   const { patientInfo } = route.params;
13:
14:   const [e,setCurrentPatientMeet] = useRecoilState(currentPatientMeetDocs);
15:   const currentPatientMeet = useRecoilValue(sortcurrentPatientMeetDocs);
16:
17:   const [patientFireInfo, setPatientFireInfo] = useState();
18:
19:   useEffect(() => {
20:     // effect
21:
22:     const docRef = doc(db, "users", patientInfo.uid);
23:
24:     const unsubscribeUserRef = onSnapshot(docRef, (doc) => {
25:       // console.log("dddddd",**/({id: doc.id, ...doc.data()}))/**/);
26:       setPatientFireInfo(**/({id: doc.id, ...doc.data()}))/**;
27:     });
28:
29:     const collectionMeetRef = collection(db, "meet_doctor");
30:     const q = query(collectionMeetRef, where("uid_patient", "==", patientInfo.uid))
31:
32:     const unsubscribe = onSnapshot(q, (querySnapshot) => {
33:       setCurrentPatientMeet(querySnapshot.docs.map((doc) => **/({id: doc.id, ...doc.data()}))/**/));
34:     });
35:     return () => {
36:       // cleanup
37:       unsubscribe()
38:       unsubscribeUserRef()
39:     }, [] )
40:
41:     return (
42:       <View isSafe style={styles.container}>
43:         <View style={styles.viewHeader}>
44:           <Pressable onPress={() => navigation.goBack()}>
45:             <Ionicons name='arrow-back-circle' size={30} color={Colors.blue}/>
46:           </Pressable>
47:           <View style={{flex: 1, flexDirection: 'row', justifyContent: 'flex-end'}}>
48:             <Button
49:               style={{paddingRight: 5}}
50:               title='à,\234à,\201à,\232à,\227à,\224à,*à,-à,\232\'
51:               borderless
52:               onPress={() => navigation.navigate("Tests", { patientInfo: patientInfo })}
53:             />
54:             <Button
55:               style={{paddingRight: 5}}
56:               title='à,\201à,\210à,\201à,èà,èà,i\'
57:               borderless
58:               onPress={() => navigation.navigate("List", { patientInfo: patientInfo })}
59:             />

```

```

6:      </View>
7:    </View>
8:  </View>
9:  <View style={styles.content}>
10:    <Text style={{[styles.textHeader, {alignSelf: 'flex-start'}]}}>à,\204à
11:    ,\231à¹\204à,\202à¹\211</Text>
12:    {patientFireInfo && <View style={{[styles.item, styles.itemPatient, {
13:    maxHeight: height * 0.2}]}>
14:      {patientFireInfo.urlImage ?
15:        <Image
16:          style={{[styles.image, styles.imagePatient]}}
17:          source={require('../assets/avatar.webp')}
18:        /> :
19:        <Image
20:          style={{[styles.image, styles.imagePatient]}}
21:          source={{uri: patientFireInfo.urlImage}}
22:        />
23:      <View style={styles.itemViewText}>
24:        <ScrollView>
25:          <Text style={styles.itemTitle}>à,\212à,²à¹\210à,-: {pati
26:          entFireInfo.firstname}</Text>
27:          <Text style={styles.itemTitle}>à,\231à,²à,jà,²à,\201à,²à
28:          ,¥: {patientFireInfo.lastname}</Text>
29:          <Text style={styles.itemTitle}>à,-à,²à,çà,,: {patientFir
30:          eInfo.age}à,\233à,µ à¹\200à,\236à,": {patientFireInfo.sex_type === "female" ? "à,«à,\215à,´à
31:          ,\207" : "à,\212à,²à,çà"}</Text>
32:          <Text style={{[styles.itemTitle, {fontSize: 14, paddingTo
33:          p: 3}]}>à,²à,´à¹\210à,\207à,\227à,µà¹\210à,\212à,-à,\232 {patientFireInfo.like}</Text>
34:          <Text style={{[styles.itemTitle, {fontSize: 14, paddingTo
35:          p: 3}]}>à,²à,´à¹\210à,\207à,\227à,µà¹\210à¹\204à,jà¹\210à,\212à,-à,\232 {patientFireInfo.unli
36:          ke}</Text>
37:          <Text style={{[styles.itemTitle, {fontSize: 14, paddingTo
38:          p: 3}]}>à,²à,´à¹\210à,\207à,\227à,µà¹\210à¹\201à,\236à¹\211 {patientFireInfo.allergy}</Text>
39:          <Text style={{[styles.itemTitle, {fontSize: 14, paddingTo
40:          p: 3}]}>à,\227à,µà¹\210à,-à,çà,¹à¹\210 {patientFireInfo.address}</Text>
41:          <Text style={{[styles.itemTitle, {fontSize: 14, paddingTo
42:          p: 3}]}>uid: {patientFireInfo.uid}</Text>
43:          <Text style={{[styles.itemTitle, {fontSize: 14, paddingTo
44:          p: 3, color: {patientFireInfo.alzheimer_lv !== ""} ? "black" : "red"}]}>à,²à,²à,²à,²à,²à,-à,²à,
45:          \201à,²à,£: {patientFireInfo.alzheimer_lv !== "" ? patientFireInfo.alzheimer_lv : "à¹\203à,«à
46:          à¹\211à¹\201à,\236à,\227à,çà¹\214à,\233à,²à,²à¹\200à,jà,²à,\223"}</Text>
47:          <Text style={{[styles.itemTitle, {fontSize: 14, paddingTo
48:          p: 3, color: {patientFireInfo.medicine !== ""} ? "black" : "red"}]}>à,²à,²à,²à,²à,²à,-à,²à,\201
49:          à,²à,£: {patientFireInfo.medicine !== "" ? patientFireInfo.medicine : "à¹\203à,«à¹\211à¹\201
50:          à,\236à,\227à,çà¹\214à,\233à,²à,²à¹\200à,jà,²à,\223"}</Text>
51:        </ScrollView>
52:      </View>
53:    </View>
54:  </View>
55:  <Text style={styles.textStyle}>à,²à,²à,²à,\201à,²à,²à,\231à,²à,\224à,«à,
56:  à,²à,çà</Text>
57:  <ScrollView style={{height: 'auto', width: width}}>
58:    <View style={{height: 4, width: width, backgroundColor: '#f0f0f0'}}>
59:      <View>
60:        {currentPatientMeet.map((data) => {
61:          return <Pressable
62:            key={data.id}
63:            style={{
64:              height: 'auto',
65:              width: width,
66:              paddingHorizontal: width * 0.05,
67:              borderBottomWidth: 4,
68:              borderColor: '#f0f0f0',
69:              paddingBottom: 5,
70:            }}
71:            onPress={() => navigation.navigate("MeetInfo", { data: d

```

```
ata }}}
106:         >
107:         <Text style={styles.itemTitle}>{data.title}</Text>
108:         { /* (new Intl.DateTimeFormat("th-TH",{ dateStyle: 'full'
, timeStyle: 'short' })).format((new Date(data.item.time.seconds * 1000))).toString() */}
109:         <Text style={styles.itemTime}>{(new Intl.DateTimeFormat(
"th-TH",{ dateStyle: 'full', timeStyle: 'short' })).format((new Date(data.time.seconds * 1000
))).toString())}</Text>
110:         </Pressable>
111:         }}}
112:     </ScrollView>
113: </View>
114: )
115: }
116:
117: export default CTakerPatientInfo
118:
119: const styles = StyleSheet.create({
120:     container: {
121:         flex: 1,
122:         alignItems: 'center',
123:         backgroundColor: 'white',
124:         height: height,
125:         // justifyContent: 'center'
126:     },
127:     content: {
128:         width: width*0.8,
129:         alignItems: 'center'
130:     },
131:     textStyle: {
132:         fontWeight: "bold",
133:         fontSize: 20,
134:     },
135:     textHeader: {
136:         fontWeight: 'bold',
137:         fontSize: 30,
138:         marginLeft: 20,
139:         marginBottom: 10
140:     },
141:     viewHeader: {
142:         width: width*0.9,
143:         flexDirection: 'row',
144:         justifyContent: 'flex-start',
145:         alignItems: 'center',
146:         alignSelf: 'center'
147:     },
148:     itemPatient:{
149:         width: width,
150:     },
151:     item:{
152:         height: 'auto',
153:         paddingHorizontal: width * 0.05,
154:         borderBottomWidth: 4,
155:         borderColor: '#f0f0f0',
156:         paddingBottom: 5,
157:         flexDirection: 'row',
158:     },
159:     imagePatient: {
160:         height: 70,
161:         width: 70,
162:     },
163:     image: {
164:         borderRadius: 5,
165:         marginVertical: 12,
166:         alignSelf: 'flex-start'
167:     },
168: }
```

```
168:     itemTitle:{
169:         fontSize: 20,
170:         fontWeight: '600',
171:     },
172:     itemTime: {
173:         fontSize: 16,
174:         fontWeight: '400'
175:     },
176:     itemViewText: {
177:         alignSelf: 'center',
178:         paddingLeft: 10
179:     },
180: }
```

```

60:     }
61:   }
62:   const renderItemNotDone = () => {
63:     return notDoneLists.map(item => {
64:       return <TouchableOpacity
65:         key={item.id}
66:         style={styles.item}
67:         onPress={() => navigation.navigate("Item", { data: item })}
68:       >
69:         <Text style={styles.itemTitle}>{item.title}</Text>
70:         /* <Text style={styles.itemTime}>{(new Date(item.day.seconds
s * 1000)).toString()}</Text> */
71:         <Text style={styles.itemTime}>{item.day_string}</Text>
72:       </TouchableOpacity>
73:     )})
74:   }
75:
76:   const renderItemAll = () => {
77:     return sortLists.map(item => {
78:       return <TouchableOpacity
79:         key={item.id}
80:         style={styles.item}
81:         onPress={() => navigation.navigate("Item", { data: item })}
82:       >
83:         <Text style={styles.itemTitle}>{item.title}</Text>
84:         /* show status */
85:         <Text style={[styles.itemTitle, {color: (item.isDone) ? "#00
bf0b" : ((curDay - (item.day.seconds * 1000) > 0) ? "red" : "#fbbf00")}]>{(item.isDone) ? "
à,\227à,\3à\201à,\Yà\211à,$" : ((curDay - (item.day.seconds * 1000)) >= 0) ? "à,\Yà,\à,\jà,\227
à,\3" : "à,\çà,\±à,\207à\204à,\jà\210à\204à,\224à\211à,\227à,\3"}</Text>
86:         <Text style={styles.itemTime}>{item.day_string}</Text>
87:       </TouchableOpacity>
88:     )})
89:   }
90:
91:   const SwitchRender = () => {
92:     switch (buttonsSelect[indexSelect]) {
93:       case 'à,\234à,\1à\211à,\233à\210à,$à,\çà,\227à,\3à\200à,\à,\èà\207à,\210
à\201à,\Yà\211à,$':
94:         return <RenderItemDone/>
95:       case 'à,\234à,\1à\211à,\233à\210à,$à,\çà,\çà,\±à,\207à\204à,\jà\210à\204
à,\224à\211à,\227à,\3':
96:         return <RenderItemNotDone/>
97:       case 'à,\èà,$à,\jà,\227à,\±à\211à,\207à,\«à,\jà,\224à,\225à,\±à\211à,\207à\
201à,\225à\210à,\225à\211à,\231à,\210à,\231à,\226à,\¶à,\207à,\«à,\Yà,\±à,\207 7 à,$à,\±à,\231à
\200à,\233à\207à,\231à,\225à\211à,\231à\204à,\233':
98:         return <RenderItemAll/>
99:       default:
100:         break;
101:     }
102:   }
103:
104:   return (
105:     <View isSafe style={styles.container}>
106:       <View style={styles.viewHeader}>
107:         <Pressable onPress={() => navigation.goBack()}>
108:           <Ionicons name='arrow-back-circle' size={30} color={Colors.blue} />
109:         </Pressable>
110:         <Text style={styles.textHeader}>>\à,\`à\210à,\207à,\227à,\µà\210à,\
225à\211à,-à,\207à,\227à,\3à,\202à,-à,\207 {patientInfo.firstname} {patientInfo.lastname}</
Text>
111:       </View>
112:       <ButtonGroup
113:         onPress={onSelectIndex}
114:         selectedIndex={indexSelect}

```

```
115:         buttons={buttonsSelect}
116:         containerStyle={{height: 70, marginBottom: 12, padding: 5}}
117:       />
118:       {!!loading ? (<ScrollView>
119:         <View style={{height: 4, width: width, backgroundColor: '#f0f0f0'}}>
</View>
120:         <SwitchRender/>
121:       </ScrollView>) : <LoadingIndicator/>}
122:     </View>
123:   )
124: }
125:
126: const styles = StyleSheet.create({
127:   container: {
128:     flex: 1,
129:     alignItems: 'center',
130:     backgroundColor: 'white',
131:     // justifyContent: 'center'
132:   },
133:   textHeader: {
134:     fontWeight: 'bold',
135:     fontSize: 20,
136:     marginLeft: 20,
137:   },
138:   btn: {
139:     width: width*0.9,
140:     height: 50,
141:     borderRadius: 5,
142:     backgroundColor: Colors.orange,
143:     alignItems: 'center',
144:     justifyContent: 'center',
145:     marginVertical: 12
146:   },
147:   item:{
148:     width: width,
149:     height: 70,
150:     paddingHorizontal: width * 0.05,
151:     borderBottomWidth: 4,
152:     borderColor: '#f0f0f0',
153:     justifyContent: 'center'
154:   },
155:   itemTitle:{
156:     fontSize: 20,
157:     fontWeight: '600'
158:   },
159:   itemTime: {
160:     fontSize: 16,
161:     fontWeight: '400'
162:   },
163:   viewHeader: {
164:     width: width*0.9,
165:     flexDirection: 'row',
166:     justifyContent: 'flex-start',
167:     alignItems: 'center',
168:     alignSelf: 'center'
169:   },
170: })
171:
```



```

66:     </Formik>
67:     {/* Button to navigate to Login screen */}
68:     <Button
69:       style={styles.borderlessButtonContainer}
70:       borderless
71:       title={'à, ¥à, \207à, \212à, •à¹\210à, -à¹\200à, \202à¹\211à, ºà¹\203à, \212à¹\211'}
72:       onPress={() => navigation.navigate('Login')}
73:     />
74:   </View>
75: );
76: };
77:
78: const styles = StyleSheet.create({
79:   container: {
80:     flex: 1,
81:     backgroundColor: Colors.white,
82:     paddingHorizontal: 12
83:   },
84:   innercontainer: {
85:     alignItems: 'center'
86:   },
87:   screenTitle: {
88:     fontSize: 32,
89:     fontWeight: '700',
90:     color: Colors.black,
91:     paddingTop: 20
92:   },
93:   button: {
94:     width: '100%',
95:     justifyContent: 'center',
96:     alignItems: 'center',
97:     marginTop: 8,
98:     backgroundColor: Colors.orange,
99:     padding: 10,
100:    borderRadius: 8
101:  },
102:  buttonText: {
103:    fontSize: 20,
104:    color: Colors.white,
105:    fontWeight: '700'
106:  },
107:  borderlessButtonContainer: {
108:    marginTop: 16,
109:    alignItems: 'center',
110:    justifyContent: 'center'
111:  }
112: });
113:

```



```

1: import React, { useState, useEffect } from 'react';
2: import { Text, StyleSheet, Alert, Platform, Image, Pressable, Dimensions } from 'react-native';
3: import { Formik } from 'formik';
4: import { KeyboardAwareScrollView } from 'react-native-keyboard-aware-scroll-view';
5: import { View, TextInput, Logo, FormErrorMessage, Button } from '../components';
6: import { DropDownList } from 'react-native-ultimate-modal-picker';
7: import * as ImagePicker from 'expo-image-picker';
8: import { Ionicons } from '@expo/vector-icons';
9:
10: import { updateUserSchema } from '../utils';
11: import { doc, setDoc } from "firebase/firestore";
12: import { Images, Colors, auth, db, storage } from '../config';
13: import { ref, uploadBytesResumable, getDownloadURL } from "firebase/storage";
14: import { useRecoilValue } from 'recoil';
15: import { userInfoAtom } from '../store';
16:
17: const {width} = Dimensions.get('window');
18:
19: export const UpdateUser = ({ navigation }) => {
20:
21:   const userInfo = useRecoilValue(userInfoAtom)
22:
23:   const [errorState, setErrorState] = useState('');
24:   const [image, setImage] = useState(null);
25:   const [btnEnable, setBtnEnable] = useState(true);
26:   const [uploadProgress, setUploadProgress] = useState(0);
27:   const [allergy, setAllergy] = useState('');
28:   const [address, setAddress] = useState('');
29:   const [like, setLike] = useState('');
30:   const [unlike, setUnlike] = useState('');
31:
32:   useEffect(() => {
33:     (async () => {
34:       if (Platform.OS !== 'web') {
35:         const { status } = await ImagePicker.requestMediaLibraryPermissionsAsync();
36:         if (status !== 'granted') {
37:           Alert.alert(
38:             "à,\201à,\2à,\200à,\202à\211à,\2à,\226à,\207à,\2à,\233à,\236",
39:             "Sorry, we need camera roll permissions to make pick the photo",
40:             [
41:               { text: "OK", onPress: () => console.log("OK Pressed") }
42:             ]
43:           )
44:         }
45:       })();
46:     }, []);
47:
48:     const pickImage = async () => {
49:       let result = await ImagePicker.launchImageLibraryAsync({
50:         mediaTypes: ImagePicker.MediaTypeOptions.Images,
51:         quality: 1,
52:       });
53:       console.log(result);
54:       if (!result.cancelled) {
55:         setImage(result.uri);
56:       }
57:     };
58:
59:     const uploadImageAsync = async (uri, callBack) => {
60:       if (uri) {
61:         console.log('image ok', uri);
62:         const imageRef = ref(storage, 'images-user/' + auth.currentUser.uid);
63:         await fetch(uri)
64:           .then(response => response.blob())

```

```

65:
66:       .then(blob => {
67:         const uploadTask = uploadBytesResumable(imageRef, blob);
68:         uploadTask.on('state_changed', (snapshot) => {
69:           const progress = (snapshot.bytesTransferred / snapshot.totalBytes) * 100;
70:           console.log('Upload is ' + progress + '% done');
71:           setUploadProgress(progress.toFixed(2));
72:         },
73:         (error) => {
74:           setErrorState(error.message);
75:         },
76:         () => {
77:           getDownloadURL(uploadTask.snapshot.ref).then((downloadURL) => {
78:             console.log('File available at', downloadURL);
79:             callBack(downloadURL);
80:           });
81:         });
82:       } else {
83:         callBack('');
84:       }
85:     }
86:
87:     const updateUser = async values => {
88:       setBtnEnable(false);
89:       const { age, sex_type, name, lastname } = values;
90:       const docRef = doc(db, "users", auth.currentUser.uid)
91:       console.log('click updateUser');
92:       await uploadImageAsync(image, async (result) => {
93:         if (result === '') {
94:           const imageRef = ref(storage, 'images-user/' + auth.currentUser.uid);
95:           // user did not pick image but have already upload image before
96:           getDownloadURL(imageRef).then(async (downloadURL) => {
97:             console.log('File available at', downloadURL);
98:             await setDoc(docRef, {urlImage: downloadURL}, { merge: true })
99:           })
100:         } else {
101:           await setDoc(docRef, {urlImage: result}, { merge: true })
102:         }
103:       const payload = {
104:         firstname: name,
105:         lastname: lastname,
106:         age: age,
107:         sex_type: sex_type,
108:       };
109:       console.log(payload);
110:
111:       if (userInfo.person_type === 'patient') {
112:         const payload = {
113:           allergy: (allergy !== '') ? allergy : userInfo.allergy,
114:           address: (address !== '') ? address : userInfo.address,
115:           like: (like !== '') ? like : userInfo.like,
116:           unlike: (unlike !== '') ? unlike : userInfo.unlike,
117:         };
118:         await setDoc(docRef, payload, { merge: true })
119:       }
120:
121:       // bug
122:       // fix https://github.com/firebase/firebase-js-sdk/issues/5667#issuecomment-95
123:       // fix bug with const db = initializeFirestore(firebaseApp, {useFetchStreams:
124:       false})
125:       await setDoc(docRef, payload, { merge: true }).then(() => {
126:         setBtnEnable(true);
127:         Alert.alert(
128:           "à\200à,\233à,\2à,\210à,\2à,\231à\201à,\233à,\207à,\202à\211à,-à,\2à,\2à,\2",

```

```
128:      "à¹\200à,\233à,¥à,µà¹\210à,çà,\231à¹\201à,\233à,¥à,\207à,\202à¹\211à,-à,jà,¹
à,¥à,\234à,¹à¹\211à¹\203à,\212à¹\211à,ªà,³à¹\200à,£à¹\207à,\210",
129:      [
130:        { text: "OK", onPress: () => {
131:          console.log("OK Pressed");
132:          navigation.goBack();
133:        }
134:      ]
135:      console.log('create user success');
136:    }).catch(error => setErrorState(error.message));
137:  });
138: };
139:
140: return (
141:   <View isSafe style={styles.container}>
142:     <View style={styles.viewHeader}>
143:       <Pressable onPress={() => navigation.goBack()} disabled={!btnEnable}>
144:         <Icons name={'arrow-back-circle'} size={30} color={Colors.blue} />
145:       </Pressable>
146:       <Text style={styles.textHeader}>à,-à,±à,\236à¹\200à,\224à,\225à,\202à¹\211à,
-à,jà,¹à,¥à,\234à,¹à¹\211à¹\203à,\212à¹\211</Text>
147:     </View>
148:     <KeyboardAwareScrollView enableOnAndroid={true}>
149:       {/* LogoContainer: consits app logo and screen title */}
150:       <View style={styles.logoContainer}>
151:         {image ? <Image source={{ uri: image }} style={{ width: 200, height: 200,
borderRadius: 100 }} /> : <Logo uri={Images.logo} />}
152:         <Text style={styles.screenTitle}>à¹\200à,\233à,¥à,µà¹\210à,çà,\231à¹\201à,
\233à,¥à,\207à,\202à¹\211à,-à,jà,¹à,¥à,\234à,¹à¹\211à¹\203à,\212à¹\211</Text>
153:         <Button style={styles.button} onPress={pickImage} disabled={!btnEnable}>
154:           <Text style={styles.buttonText}>à¹\200à,¥à,ªà,-à,\201à,£à,¹à,\233à, à,²à
,\236à¹\202à,\233à,£à¹\204à,\237à,¥à¹\214</Text>
155:         </Button>
156:         {!btnEnable ? <Text style={{paddingVertical: 16}}>à,-à,±à,\236à¹\202à,«à,¥
à,\224à,£à,¹à,\233à, à,²à,\236à¹\201à,¥à¹\211à,$ {uploadProgress} %</Text> : null}
157:       </View>
158:       {/* Formik Wrapper */}
159:       <Formik
160:         initialValues={{
161:           age: '',
162:           sex_type: '',
163:           name: '',
164:           lastname: ''
165:         }}
166:         validationSchema={updateUserSchema}
167:         onSubmit={values => updateUser(values)}
168:       >
169:         {({
170:           values,
171:           touched,
172:           errors,
173:           handleChange,
174:           handleSubmit,
175:           handleBlur,
176:           isSubmitting,
177:         }) => (
178:           <>
179:             {/* name */}
180:             <TextInput
181:               name='name_lastname'
182:               placeholder='à,\201à,£à,,à,\223à,²à,£à,°à,\232à,,à,\212à,ªà¹\210à,-'
183:               autoCapitalize='none'
184:               autoFocus={true}
185:               value={values.name}
186:               onChangeText={handleChange('name')}
187:               onBlur={handleBlur('name')}
188:             />
189:             <FormErrorMessage
190:               error={errors.name}
191:               visible={touched.name}
192:             />
193:             <TextInput
194:               name='lastname'
195:               placeholder='à,\201à,£à,,à,\223à,²à,£à,°à,\232à,,à,\231à,²à,jà,ªà,
\201à,,à,¥'
196:               autoCapitalize='none'
197:               autoFocus={true}
198:               value={values.lastname}
199:               onChangeText={handleChange('lastname')}
200:               onBlur={handleBlur('lastname')}
201:             />
202:             <FormErrorMessage
203:               error={errors.lastname}
204:               visible={touched.lastname}
205:             />
206:             {/* age */}
207:             <TextInput
208:               name='age'
209:               placeholder='à¹\203à,ªà¹\210à,-à,²à,çà,,à,\202à,-à,\207à,\204à,,à,
\223'
210:               autoCapitalize='none'
211:               autoFocus={true}
212:               value={values.age}
213:               onChangeText={handleChange('age')}
214:               onBlur={handleBlur('age')}
215:             />
216:             <FormErrorMessage
217:               error={errors.age}
218:               visible={touched.age}
219:             />
220:             {/* patient */}
221:             {userInfo.person_type === 'patient' ?
222:               <>
223:                 <TextInput
224:                   name='à,ªà,¹à¹\210à,\207à,\227à,µà¹\210à¹\201à,\236à¹\211'
225:                   placeholder='à,\201à,£à,,à,\223à,²à,£à,°à,\232à,,à,ªà,¹à¹\210à,
\207à,\227à,µà¹\210à¹\201à,\236à¹\211'
226:                   autoCapitalize='none'
227:                   autoFocus={true}
228:                   value={allergy}
229:                   onChangeText={setAllergy}
230:                 />
231:                 <TextInput
232:                   name='à,ªà,¹à¹\210à,\207à,\227à,µà¹\210à,\212à,-à,\232'
233:                   placeholder='à,\201à,£à,,à,\223à,²à,£à,°à,\232à,,à,ªà,¹à¹\210à,
\207à,\227à,µà¹\210à,\212à,-à,\232'
234:                   autoCapitalize='none'
235:                   autoFocus={true}
236:                   value={like}
237:                   onChangeText={setLike}
238:                 />
239:                 <TextInput
240:                   name='à,ªà,¹à¹\210à,\207à,\227à,µà¹\210à¹\204à,jà¹\210à,\212à,-à,
\232'
241:                   placeholder='à,\201à,£à,,à,\223à,²à,£à,°à,\232à,,à,ªà,¹à¹\210à,
\207à,\227à,µà¹\210à¹\204à,jà¹\210à,\212à,-à,\232'
242:                   autoCapitalize='none'
243:                   autoFocus={true}
244:                   value={unlike}
245:                   onChangeText={setUnlike}
246:                 />
247:               </>
248:             }
249:           </>
250:         )
251:       }
252:     )
253:   )
254: )
255: )
256: )
257: )
258: )
259: )
260: )
261: )
262: )
263: )
264: )
265: )
266: )
267: )
268: )
269: )
270: )
271: )
272: )
273: )
274: )
275: )
276: )
277: )
278: )
279: )
280: )
281: )
282: )
283: )
284: )
285: )
286: )
287: )
288: )
289: )
290: )
291: )
292: )
293: )
294: )
295: )
296: )
297: )
298: )
299: )
300: )
301: )
302: )
303: )
304: )
305: )
306: )
307: )
308: )
309: )
310: )
311: )
312: )
313: )
314: )
315: )
316: )
317: )
318: )
319: )
320: )
321: )
322: )
323: )
324: )
325: )
326: )
327: )
328: )
329: )
330: )
331: )
332: )
333: )
334: )
335: )
336: )
337: )
338: )
339: )
340: )
341: )
342: )
343: )
344: )
345: )
346: )
347: )
348: )
349: )
350: )
351: )
352: )
353: )
354: )
355: )
356: )
357: )
358: )
359: )
360: )
361: )
362: )
363: )
364: )
365: )
366: )
367: )
368: )
369: )
370: )
371: )
372: )
373: )
374: )
375: )
376: )
377: )
378: )
379: )
380: )
381: )
382: )
383: )
384: )
385: )
386: )
387: )
388: )
389: )
390: )
391: )
392: )
393: )
394: )
395: )
396: )
397: )
398: )
399: )
400: )
401: )
402: )
403: )
404: )
405: )
406: )
407: )
408: )
409: )
410: )
411: )
412: )
413: )
414: )
415: )
416: )
417: )
418: )
419: )
420: )
421: )
422: )
423: )
424: )
425: )
426: )
427: )
428: )
429: )
430: )
431: )
432: )
433: )
434: )
435: )
436: )
437: )
438: )
439: )
440: )
441: )
442: )
443: )
444: )
445: )
446: )
447: )
448: )
449: )
450: )
451: )
452: )
453: )
454: )
455: )
456: )
457: )
458: )
459: )
460: )
461: )
462: )
463: )
464: )
465: )
466: )
467: )
468: )
469: )
470: )
471: )
472: )
473: )
474: )
475: )
476: )
477: )
478: )
479: )
480: )
481: )
482: )
483: )
484: )
485: )
486: )
487: )
488: )
489: )
490: )
491: )
492: )
493: )
494: )
495: )
496: )
497: )
498: )
499: )
500: )
501: )
502: )
503: )
504: )
505: )
506: )
507: )
508: )
509: )
510: )
511: )
512: )
513: )
514: )
515: )
516: )
517: )
518: )
519: )
520: )
521: )
522: )
523: )
524: )
525: )
526: )
527: )
528: )
529: )
530: )
531: )
532: )
533: )
534: )
535: )
536: )
537: )
538: )
539: )
540: )
541: )
542: )
543: )
544: )
545: )
546: )
547: )
548: )
549: )
550: )
551: )
552: )
553: )
554: )
555: )
556: )
557: )
558: )
559: )
560: )
561: )
562: )
563: )
564: )
565: )
566: )
567: )
568: )
569: )
570: )
571: )
572: )
573: )
574: )
575: )
576: )
577: )
578: )
579: )
580: )
581: )
582: )
583: )
584: )
585: )
586: )
587: )
588: )
589: )
590: )
591: )
592: )
593: )
594: )
595: )
596: )
597: )
598: )
599: )
600: )
601: )
602: )
603: )
604: )
605: )
606: )
607: )
608: )
609: )
610: )
611: )
612: )
613: )
614: )
615: )
616: )
617: )
618: )
619: )
620: )
621: )
622: )
623: )
624: )
625: )
626: )
627: )
628: )
629: )
630: )
631: )
632: )
633: )
634: )
635: )
636: )
637: )
638: )
639: )
640: )
641: )
642: )
643: )
644: )
645: )
646: )
647: )
648: )
649: )
650: )
651: )
652: )
653: )
654: )
655: )
656: )
657: )
658: )
659: )
660: )
661: )
662: )
663: )
664: )
665: )
666: )
667: )
668: )
669: )
670: )
671: )
672: )
673: )
674: )
675: )
676: )
677: )
678: )
679: )
680: )
681: )
682: )
683: )
684: )
685: )
686: )
687: )
688: )
689: )
690: )
691: )
692: )
693: )
694: )
695: )
696: )
697: )
698: )
699: )
700: )
701: )
702: )
703: )
704: )
705: )
706: )
707: )
708: )
709: )
710: )
711: )
712: )
713: )
714: )
715: )
716: )
717: )
718: )
719: )
720: )
721: )
722: )
723: )
724: )
725: )
726: )
727: )
728: )
729: )
730: )
731: )
732: )
733: )
734: )
735: )
736: )
737: )
738: )
739: )
740: )
741: )
742: )
743: )
744: )
745: )
746: )
747: )
748: )
749: )
750: )
751: )
752: )
753: )
754: )
755: )
756: )
757: )
758: )
759: )
760: )
761: )
762: )
763: )
764: )
765: )
766: )
767: )
768: )
769: )
770: )
771: )
772: )
773: )
774: )
775: )
776: )
777: )
778: )
779: )
780: )
781: )
782: )
783: )
784: )
785: )
786: )
787: )
788: )
789: )
790: )
791: )
792: )
793: )
794: )
795: )
796: )
797: )
798: )
799: )
800: )
801: )
802: )
803: )
804: )
805: )
806: )
807: )
808: )
809: )
810: )
811: )
812: )
813: )
814: )
815: )
816: )
817: )
818: )
819: )
820: )
821: )
822: )
823: )
824: )
825: )
826: )
827: )
828: )
829: )
830: )
831: )
832: )
833: )
834: )
835: )
836: )
837: )
838: )
839: )
840: )
841: )
842: )
843: )
844: )
845: )
846: )
847: )
848: )
849: )
850: )
851: )
852: )
853: )
854: )
855: )
856: )
857: )
858: )
859: )
860: )
861: )
862: )
863: )
864: )
865: )
866: )
867: )
868: )
869: )
870: )
871: )
872: )
873: )
874: )
875: )
876: )
877: )
878: )
879: )
880: )
881: )
882: )
883: )
884: )
885: )
886: )
887: )
888: )
889: )
890: )
891: )
892: )
893: )
894: )
895: )
896: )
897: )
898: )
899: )
900: )
901: )
902: )
903: )
904: )
905: )
906: )
907: )
908: )
909: )
910: )
911: )
912: )
913: )
914: )
915: )
916: )
917: )
918: )
919: )
920: )
921: )
922: )
923: )
924: )
925: )
926: )
927: )
928: )
929: )
930: )
931: )
932: )
933: )
934: )
935: )
936: )
937: )
938: )
939: )
940: )
941: )
942: )
943: )
944: )
945: )
946: )
947: )
948: )
949: )
950: )
951: )
952: )
953: )
954: )
955: )
956: )
957: )
958: )
959: )
960: )
961: )
962: )
963: )
964: )
965: )
966: )
967: )
968: )
969: )
970: )
971: )
972: )
973: )
974: )
975: )
976: )
977: )
978: )
979: )
980: )
981: )
982: )
983: )
984: )
985: )
986: )
987: )
988: )
989: )
990: )
991: )
992: )
993: )
994: )
995: )
996: )
997: )
998: )
999: )
1000: )
```

```
188: />
189: <FormErrorMessage
190:   error={errors.name}
191:   visible={touched.name}
192: />
193: <TextInput
194:   name='lastname'
195:   placeholder='à,\201à,£à,,à,\223à,²à,£à,°à,\232à,,à,\231à,²à,jà,ªà,
\201à,,à,¥'
196:   autoCapitalize='none'
197:   autoFocus={true}
198:   value={values.lastname}
199:   onChangeText={handleChange('lastname')}
200:   onBlur={handleBlur('lastname')}
201: />
202: <FormErrorMessage
203:   error={errors.lastname}
204:   visible={touched.lastname}
205: />
206: {/* age */}
207: <TextInput
208:   name='age'
209:   placeholder='à¹\203à,ªà¹\210à,-à,²à,çà,,à,\202à,-à,\207à,\204à,,à,
\223'
210:   autoCapitalize='none'
211:   autoFocus={true}
212:   value={values.age}
213:   onChangeText={handleChange('age')}
214:   onBlur={handleBlur('age')}
215: />
216: <FormErrorMessage
217:   error={errors.age}
218:   visible={touched.age}
219: />
220: {/* patient */}
221: {userInfo.person_type === 'patient' ?
222:   <>
223:     <TextInput
224:       name='à,ªà,¹à¹\210à,\207à,\227à,µà¹\210à¹\201à,\236à¹\211'
225:       placeholder='à,\201à,£à,,à,\223à,²à,£à,°à,\232à,,à,ªà,¹à¹\210à,
\207à,\227à,µà¹\210à¹\201à,\236à¹\211'
226:       autoCapitalize='none'
227:       autoFocus={true}
228:       value={allergy}
229:       onChangeText={setAllergy}
230:     />
231:     <TextInput
232:       name='à,ªà,¹à¹\210à,\207à,\227à,µà¹\210à,\212à,-à,\232'
233:       placeholder='à,\201à,£à,,à,\223à,²à,£à,°à,\232à,,à,ªà,¹à¹\210à,
\207à,\227à,µà¹\210à,\212à,-à,\232'
234:       autoCapitalize='none'
235:       autoFocus={true}
236:       value={like}
237:       onChangeText={setLike}
238:     />
239:     <TextInput
240:       name='à,ªà,¹à¹\210à,\207à,\227à,µà¹\210à¹\204à,jà¹\210à,\212à,-à,
\232'
241:       placeholder='à,\201à,£à,,à,\223à,²à,£à,°à,\232à,,à,ªà,¹à¹\210à,
\207à,\227à,µà¹\210à¹\204à,jà¹\210à,\212à,-à,\232'
242:       autoCapitalize='none'
243:       autoFocus={true}
244:       value={unlike}
245:       onChangeText={setUnlike}
246:     />
247:   </>
248: }
249: }
250: )
251: )
252: )
253: )
254: )
255: )
256: )
257: )
258: )
259: )
260: )
261: )
262: )
263: )
264: )
265: )
266: )
267: )
268: )
269: )
270: )
271: )
272: )
273: )
274: )
275: )
276: )
277: )
278: )
279: )
280: )
281: )
282: )
283: )
284: )
285: )
286: )
287: )
288: )
289: )
290: )
291: )
292: )
293: )
294: )
295: )
296: )
297: )
298: )
299: )
300: )
301: )
302: )
303: )
304: )
305: )
306: )
307: )
308: )
309: )
310: )
311: )
312: )
313: )
314: )
315: )
316: )
317: )
318: )
319: )
320: )
321: )
322: )
323: )
324: )
325: )
326: )
327: )
328: )
329: )
330: )
331: )
332: )
333: )
334: )
335: )
336: )
337: )
338: )
339: )
340: )
341: )
342: )
343: )
344: )
345: )
346: )
347: )
348: )
349: )
350: )
351: )
352: )
353: )
354: )
355: )
356: )
357: )
358: )
359: )
360: )
361: )
362: )
363: )
364: )
365: )
366: )
367: )
368: )
369: )
370: )
371: )
372: )
373: )
374: )
375: )
376: )
377: )
378: )
379: )
380: )
381: )
382: )
383: )
384: )
385: )
386: )
387: )
388: )
389: )
390: )
391: )
392: )
393: )
394: )
395: )
396: )
397: )
398: )
399: )
400: )
401: )
402: )
403: )
404: )
405: )
406: )
407: )
408: )
409: )
410: )
411: )
412: )
413: )
414: )
415: )
416: )
417: )
418: )
419: )
420: )
421: )
422: )
423: )
424: )
425: )
426: )
427: )
428: )
429: )
430: )
431: )
432: )
433: )
434: )
435: )
436: )
437: )
438: )
439: )
440: )
441: )
442: )
443: )
444: )
445: )
446: )
447: )
448: )
449: )
450: )
451: )
452: )
453: )
454: )
455: )
456: )
457: )
458: )
459: )
460: )
461: )
462: )
463: )
464: )
465: )
466: )
467: )
468: )
469: )
470: )
471: )
472: )
473: )
474: )
475: )
476: )
477: )
478: )
479: )
480: )
481: )
482: )
483: )
484: )
485: )
486: )
487: )
488: )
489: )
490: )
491: )
492: )
493: )
494: )
495: )
496: )
497: )
498: )
499: )
500: )
501: )
502: )
503: )
504: )
505: )
506: )
507: )
508: )
509: )
510: )
511: )
512: )
513: )
514: )
515: )
516: )
517: )
518: )
519: )
520: )
521: )
522: )
523: )
524: )
525: )
526: )
527: )
528: )
529: )
530: )
531: )
532: )
533: )
534: )
535: )
536: )
537: )
538: )
539: )
540: )
541: )
542: )
543: )
544: )
545: )
546: )
547: )
548: )
549: )
550: )
551: )
552: )
553: )
554: )
555: )
556: )
557: )
558: )
559: )
560: )
561: )
562: )
563: )
564: )
565: )
566: )
567: )
568: )
569: )
570: )
571: )
572: )
573: )
574: )
575: )
576: )
577: )
578: )
579: )
580: )
581: )
582: )
583: )
584: )
585: )
586: )
587: )
588: )
589: )
590: )
591: )
592: )
593: )
594: )
595: )
596: )
597: )
598: )
599: )
600: )
601: )
602: )
603: )
604: )
605: )
606: )
607: )
608: )
609: )
610: )
611: )
612: )
613: )
614: )
615: )
616: )
617: )
618: )
619: )
620: )
621: )
622: )
623: )
624: )
625: )
626: )
627: )
628: )
629: )
630: )
631: )
632: )
633: )
634: )
635: )
636: )
637: )
638: )
639: )
640: )
641: )
642: )
643: )
644: )
645: )
646: )
647: )
648: )
649: )
650: )
651: )
652: )
653: )
654: )
655: )
656: )
657: )
658: )
659: )
660: )
661: )
662: )
663: )
664: )
665: )
666: )
667: )
668: )
669: )
670: )
671: )
672: )
673: )
674: )
675: )
676: )
677: )
678: )
679: )
680: )
681: )
682: )
683: )
684: )
685: )
686: )
687: )
688: )
689: )
690: )
691: )
692: )
693: )
694: )
695: )
696: )
697: )
698: )
699: )
700: )
701: )
702: )
703: )
704: )
705: )
706: )
707: )
708: )
709: )
710: )
711: )
712: )
713: )
714: )
715: )
716: )
717: )
718: )
719: )
720: )
721: )
722: )
723: )
724: )
725: )
726: )
727: )
728: )
729: )
730: )
731: )
732: )
733: )
734: )
735: )
736: )
737: )
738: )
739: )
740: )
741: )
742: )
743: )
744: )
745: )
746: )
747: )
748: )
749: )
750: )
751: )
752: )
753: )
754: )
755: )
756: )
757: )
758: )
759: )
760: )
761: )
762: )
763: )
764: )
765: )
766: )
767: )
768: )
769: )
770: )
771: )
772: )
773: )
774: )
775: )
776: )
777: )
778: )
779: )
780: )
781: )
782: )
783: )
784: )
785: )
786: )
787: )
788: )
789: )
790: )
791: )
792: )
793: )
794: )
795: )
796: )
797: )
798: )
799: )
800: )
801: )
802: )
803: )
804: )
805: )
806: )
807: )
808: )
809: )
810: )
811: )
812: )
813: )
814: )
815: )
816: )
817: )
818: )
819: )
820: )
821: )
822: )
823: )
824: )
825: )
826: )
827: )
828: )
829: )
830: )
831: )
832: )
833: )
834: )
835: )
836: )
837: )
838: )
839: )
840: )
841: )
842: )
843: )
844: )
845: )
846: )
847: )
848: )
849: )
850: )
851: )
852: )
853: )
854: )
855: )
856: )
857: )
858: )
859: )
860: )
861: )
862: )
863: )
864: )
865: )
866: )
867: )
868: )
869: )
870: )
871: )
872: )
873: )
874: )
875: )
876: )
877: )
878: )
879: )
880: )
881: )
882: )
883: )
884: )
885: )
886: )
887: )
888: )
889: )
890: )
891: )
892: )
893: )
894: )
895: )
896: )
897: )
898: )
899: )
900: )
901: )
902: )
903: )
904: )
905: )
906: )
907: )
908: )
909: )
910: )
911: )
912: )
913: )
914: )
915: )
916: )
917: )
918: )
919: )
920: )
921: )
922: )
923: )
924: )
925: )
926: )
927: )

```

```
248:         <TextInput
249:           name='à,\227à,\210à,-à,à,\210'
250:           placeholder='à,\201à,à,\223à,\2à,à,\232à,\227à,\210à,
-à,à,\210'
251:           autoCapitalize='none'
252:           autoFocus={true}
253:           value={address}
254:           onChangeText={setAddress}
255:         />
256:       </>
257:     : null)
258:
259:
260:     {/* picker sex type */}
261:     <DropDownList
262:       title="à\200à,\2à,\201à\200à,\236à,"
263:       items={[
264:         { label: 'à,\215à,\207', value: 'female' },
265:         { label: 'à,\212à,\2à,\207', value: 'male' },
266:       ]}
267:       onChange={handleChange('sex_type')}
268:     />
269:     <FormErrorMessage
270:       error={errors.sex_type}
271:       visible={touched.sex_type}
272:     />
273:
274:     {/* Display Screen Error Mesages */}
275:     {errorState !== '' ? (
276:       <FormErrorMessage error={errorState} visible={true} />
277:     ) : null}
278:     {/* Update button */}
279:     <Button style={styles.button} onPress={handleSubmit} disabled={!isSub
mitting == btnEnable) ? false : true}>
280:       <Text style={styles.buttonText}>à\200à,\233à,\210à,à,\231à\
201à,\233à,\207à,\202à\211à,-à,\234à,\211à\203à,\212à\211</Text>
281:     </Button>
282:   </>
283: )}
284: </Formik>
285: </KeyboardAwareScrollView>
286: </View>
287: );
288: };
289:
290: const styles = StyleSheet.create({
291:   container: {
292:     flex: 1,
293:     backgroundColor: Colors.white,
294:     paddingHorizontal: 12
295:   },
296:   logoContainer: {
297:     alignItems: 'center'
298:   },
299:   screenTitle: {
300:     fontSize: 32,
301:     fontWeight: '700',
302:     color: Colors.black,
303:     paddingTop: 20
304:   },
305:   button: {
306:     width: '100%',
307:     justifyContent: 'center',
308:     alignItems: 'center',
309:     marginTop: 8,
310:     backgroundColor: Colors.orange,
```

```
311:     padding: 10,
312:     borderRadius: 8
313:   },
314:   buttonText: {
315:     fontSize: 20,
316:     color: Colors.white,
317:     fontWeight: '700'
318:   },
319:   borderlessButtonContainer: {
320:     marginTop: 16,
321:     marginBottom: 32,
322:     alignItems: 'center',
323:     justifyContent: 'center'
324:   },
325:   textHeader: {
326:     fontWeight: 'bold',
327:     fontSize: 30,
328:     marginLeft: 20,
329:     marginBottom: 10
330:   },
331:   viewHeader: {
332:     width: width*0.9,
333:     flexDirection: 'row',
334:     justifyContent: 'flex-start',
335:     alignItems: 'center',
336:     alignSelf: 'center'
337:   },
338: });
```



```

211à,- 4 {data.choice4.title}</Text>
45:         <Image style={{width: width * 0.8, height: width * 0.8, borderRa
dius: 5, marginVertical: 5}} source={require("../.../assets/camel.jpg")}/>
46:         <Text style={[styles.textStyle, {fontWeight: '300'}}]>à¹\200à,
\211à,¥à,¢: {data.choice4.answer}</Text>
47:         <Text style={[styles.textStyle, {fontWeight: '300'}}]>à,\204à,³à
,\225à,-à,\232: {data.choice4.user_answer}</Text>
48:         <View style={{height: 4, width: 'auto', backgroundColor: '#f0f0f
0'}}></View>
49:
50:         {/* 5 */}
51:         <Text style={[styles.textStyle, {fontWeight: '300'}}]>à,\202à¹
\211à,- 5 {data.choice5.title}</Text>
52:         <Image style={{width: width * 0.8, height: width * 0.8, borderRa
dius: 5, marginVertical: 5}} source={{uri: data.choice5.url}}/>
53:         <View style={{height: 4, width: 'auto', backgroundColor: '#f0f0f
0'}}></View>
54:
55:         {/* 6 */}
56:         <Text style={[styles.textStyle, {fontWeight: '300'}}]>à,\202à¹
\211à,- 6 {data.choice6.title}</Text>
57:         <Image style={{width: width * 0.8, height: width * 0.8, borderRa
dius: 5, marginVertical: 5}} source={{uri: data.choice6.url}}/>
58:         </ScrollView>
59:     )
60:     } else if (data.type === "simple") {
61:         return <ScrollView style={{marginBottom: 100}}>
62:             <View style={{height: 4, width: 'auto', backgroundColor: '#f0f0f0'}}
></View>
63:             {data.choice.map((item) => {
64:                 if (item.isTest) {
65:                     return <View key={item.id}>
66:                         <Text style={[styles.textStyle, {fontWeight: "bold"}]}>à
,\202à¹\211à,-{item.id} {item.title}</Text>
67:                         <Text style={[styles.textStyle, {fontWeight: "normal"}]}
>à,\204à,³à,\225à,-à,\232: à¹\200à,¥à,•à,-à,\201à,\202à¹\211à,-{item.choose_num} {answer_typ
e[item.choose_num-1]}</Text>
68:                         <View style={{height: 4, width: 'auto', backgroundColor:
'#f0f0f0'}}></View>
69:                     </View>
70:                 }
71:             })}
72:         </ScrollView>
73:     } else {
74:         return <</>
75:     }
76: }
77:
78: return (
79:     <View isSafe style={styles.container}>
80:         <View style={styles.viewHeader}>
81:             <Pressable onPress={() => navigation.goBack()}>
82:                 <Icons name={'arrow-back-circle'} size={30} color={Colors.blue}
e) />
83:             </Pressable>
84:             <Text style={styles.textHeader}>à,\234à,¥à¹\201à,\232à,\232à,\227à,
\224à,àà,-à,\232</Text>
85:         </View>
86:         <View style={styles.content}>
87:             <Text style={[styles.textStyle, {fontWeight: '300'}}]>{(new Intl.Dat
eTimeFormat("th-TH",{ dateStyle: 'full', timeStyle: 'short' }).format((new Date(data.user_ti
me.seconds * 1000))).toString())}</Text>
88:             <RenderTest/>
89:         </View>
90:     </View>
91: )

```

```
92: }
93:
94: export default TestInfo
95:
96: const styles = StyleSheet.create({
97:   container: {
98:     flex: 1,
99:     alignItems: 'center',
100:    backgroundColor: 'white',
101:    height: height
102:    // justifyContent: 'center'
103:  },
104:  content: {
105:    width: width*0.8,
106:    alignItems: 'center'
107:  },
108:  button: {
109:    justifyContent: 'center',
110:    width: width*0.9,
111:    height: 50,
112:    borderRadius: 5,
113:    marginVertical: 10
114:  },
115:  buttonOpen: {
116:    backgroundColor: Colors.orange,
117:  },
118:  buttonSummit: {
119:    backgroundColor: '#1597e5',
120:  },
121:  textStyle: {
122:    fontWeight: "bold",
123:    fontSize: 20,
124:    textAlign: "left",
125:    alignSelf: 'flex-start'
126:  },
127:  textHeader: {
128:    fontWeight: 'bold',
129:    fontSize: 30,
130:    marginLeft: 20,
131:    marginBottom: 10
132:  },
133:  viewHeader: {
134:    width: width*0.9,
135:    flexDirection: 'row',
136:    justifyContent: 'flex-start',
137:    alignItems: 'center',
138:    alignSelf: 'center'
139:  },
140: })
```



```

1: import React, {useState} from 'react'
2: import { StyleSheet, Text, Dimensions, ScrollView, Modal, TextInput, Pressable, Alert } from 'react-native'
3: import { KeyboardAwareScrollView } from 'react-native-keyboard-aware-scroll-view';
4: import { View, Button } from '.././../components'
5: import { Colors, db } from '.././../config';
6: import { doc, setDoc } from 'firebase/firestore';
7: import { Ionicons } from '@expo/vector-icons';
8: const {width, height} = Dimensions.get('window');
9:
10: const DoctorMeetItem = ({navigation, route}) => {
11:
12:   const { data } = route.params
13:
14:   const [note, setNote] = useState('');
15:   const [modalVisible, setModalVisible] = useState(false);
16:   const [enable, setEnable] = useState(true)
17:
18:   const updateNote = async () => {
19:     if (note.length === 0) {
20:       Alert.alert("à,è,°à,\232à,\202à¹\211à,-à,\204à,$à,²à,i","à,è,°à,\232à,\202à¹\211à,-à,\204à,$à,²à,ià¹\203à,\231à,\212à¹\210à,-à,\207à,$à¹\210à,²à,\207")
21:       return
22:     }
23:     setEnable(false)
24:     await setDoc(doc(db, "meet_doctor", data.id), {note: note}, {merge: true})
25:     setNote('');
26:     setModalVisible(false);
27:     setEnable(true)
28:     navigation.goBack()
29:   }
30:
31:   return (
32:     <ScrollView>
33:       <View isSafe style={styles.container}>
34:         <View style={styles.viewHeader}>
35:           <Pressable onPress={() => navigation.goBack()}>
36:             <Ionicons name='arrow-back-circle' size={30} color={Colors.blue} />
37:           </Pressable>
38:           <Text style={styles.textHeader}><data.title></Text>
39:           <View style={{flex: 1, flexDirection: 'row', justifyContent: 'flex-end'}}>
40:             <Button
41:               style={{paddingRight: 5}}
42:               title='à,-à,±à,\236à¹\200à,\224à,\225 note'
43:               borderless
44:               onPress={() => setModalVisible(true)}
45:               // à,\227à,³ screen list all test
46:               // à,\227à,³ test info
47:             />
48:           </View>
49:         </View>
50:         <Modal
51:           animationType="slide"
52:           visible={modalVisible}
53:         >
54:           <KeyboardAwareScrollView enableOnAndroid={true}>
55:             <View isSafe style={[styles.container, {height: height, width: width}]>
56:               <View style={styles.viewHeader}>
57:                 <Pressable
58:                   onPress={() => {
59:                     setModalVisible(false)
60:                     setNote('');
61:                   }}

```

[illegible]

```

118:     backgroundColor: Colors.orange,
119:   },
120:   buttonSubmit: {
121:     backgroundColor: '#1597e5',
122:   },
123:   textStyle: {
124:     fontWeight: "bold",
125:     fontSize: 20,
126:     alignSelf: 'flex-start'
127:   },
128:   textHeader: {
129:     fontWeight: 'bold',
130:     fontSize: 30,
131:     marginLeft: 20,
132:     marginBottom: 10
133:   },
134:   viewHeader: {
135:     width: width*0.9,
136:     flexDirection: 'row',
137:     justifyContent: 'flex-start',
138:     alignItems: 'center',
139:     alignSelf: 'center'
140:   },
141:   itemPatient:{
142:     width: width,
143:   },
144:   item:{
145:     height: 'auto',
146:     paddingHorizontal: width * 0.05,
147:     borderBottomWidth: 4,
148:     borderColor: '#f0f0f0',
149:     paddingBottom: 5,
150:     flexDirection: 'row',
151:   },
152:   imagePatient: {
153:     height: 70,
154:     width: 70,
155:   },
156:   image: {
157:     borderRadius: 5,
158:     marginVertical: 12,
159:     alignSelf: 'flex-start'
160:   },
161:   itemTitle:{
162:     fontSize: 20,
163:     fontWeight: '600',
164:   },
165:   itemTime: {
166:     fontSize: 16,
167:     fontWeight: '400'
168:   },
169:   itemViewText: {
170:     alignSelf: 'center',
171:     paddingLeft: 10
172:   },
173:   picker: {
174:     width: 90,
175:     marginHorizontal: (width - 90)/2,
176:     marginVertical: 5
177:   },
178:   titleInput:{
179:     height: 50,
180:     width: width*0.9,
181:     margin: 12,
182:     padding: 10,
183:     borderRadius: 5,

```

```

184:     fontSize: 20,
185:     backgroundColor: "#f7f7f7",
186:   },
187:   multiInput: {
188:     minHeight: 100,
189:     maxHeight: height*0.5,
190:     width: width*0.9,
191:     margin: 12,
192:     padding: 10,
193:     borderRadius: 5,
194:     fontSize: 20,
195:     backgroundColor: "#f7f7f7",
196:   },
197: })
198:
199:

```

1

```

332à, ,à, \202à¹\211à, -à, \204à, $à, ºà, ;à¹\203à, \231à, \212à¹\210à, -à, \207à, $à¹\210à, ºà, \207")
63:         return
64:     }
65:     const docRef = collection(db, 'meet_doctor')
66:     setEnable(false)
67:     await addDoc(docRef, {
68:         description: description,
69:         doctor_name: doctorInfo.firstname + " " + doctorInfo.lastname,
70:         time: time,
71:         uid_doctor: doctorInfo.uid,
72:         uid_patient: patientInfo.uid,
73:         title: title,
74:         note: ""
75:     })
76:     setDescription('');
77:     setTitle('');
78:     setShow(false);
79:     setModalVisible(false)
80:     setEnable(true)
81: }
82:
83: const update_lv = async () => {
84:     if (description_al_lv === "" || medicine === "") {
85:         Alert.alert("à, ¢à, ºà, \232à, ,à, \202à¹\211à, -à, \204à, $à, ºà, ;à¹, "à, ¢à, ºà,
\232à, ,à, \202à¹\211à, -à, \204à, $à, ºà, ;à¹\203à, \231à, \212à¹\210à, -à, \207à, $à¹\210à, ºà, \207")
86:         return
87:     }
88:     const docRef = doc(db, 'users', patientInfo.uid)
89:     setEnable(false)
90:     await setDoc(docRef, {
91:         alzheimer_lv: description_al_lv,
92:         medicine: medicine
93:     }, {merge: true})
94:     setDescription_al_lv('');
95:     setMedicine('');
96:     setModalAl(false)
97:     setEnable(true)
98: }
99:
100: return (
101:     <View isSafe style={styles.container}>
102:         <View style={styles.viewHeader}>
103:             <Pressable onPress={() => navigation.goBack()}>
104:                 <Icons name={'arrow-back-circle'} size={30} color={Colors.blu
e />
105:             </Pressable>
106:             <View style={{flex: 1, flexDirection: 'row', justifyContent: 'flex-e
nd'}}>
107:                 <Button
108:                     style={{paddingRight: 5}}
109:                     title='à, \234à, ¥à¹\201à, \232à, \232à, \227à, \224à, ºà, -à, \232'
110:                     borderless
111:                     onPress={() => navigation.navigate("Tests", { patientInfo: p
atientInfo })}
112:                     // à, \227à, º screen list all test
113:                     // à, \227à, º test info
114:                 />
115:                 <Button
116:                     style={{paddingHorizontal: 5}}
117:                     title='à, -à, ¢à, \236à¹\200à, \224à, \225à, \202à¹\211à, -à, ;à, ºà,
¥à, \234à, ,à¹\211à, \233à¹\210à, $à, ¢'
118:                     borderless
119:                     onPress={() => setModalAl(true)}
120:                 />
121:                 <Button
122:                     style={{paddingHorizontal: 5}}

```

```
123:         title='à,²à,£à¹\211à,²à,\207à,\231à,±à,\224'
124:         borderless
125:         onPress={() => setModalVisible(true)}
126:     />
127: </View>
128: </View>
129: { /* modal 1 */ }
130: <Modal
131:     animationType="slide"
132:     visible={modalAl}
133: >
134:     <KeyboardAwareScrollView enableOnAndroid={true}>
135:     <View isSafe style={[styles.container, {height: height, width: width
136:     ]}>
137:         <View style={styles.viewHeader}>
138:             <Pressable
139:                 onPress={() => {
140:                     setModalAl(false)
141:                     setDescription_al_lv('');
142:                     setMedicine('');
143:                 }}
144:                 >
145:                     <Ionicons name={'arrow-back-circle'} size={30} color={Co
146:                     lors.blue} />
147:                 </Pressable>
148:                 <Text style={styles.textHeader}>à¹\200à,\233à,¥à,µà¹\210à,çà
149:                 ,\231à¹\201à,\233à,¥à,\207à,-à,²à,\201à,²à,£</Text>
150:             </View>
151:             <Text style={[styles.textStyle, {marginLeft: width*0.05}]}>à,£à,
152:             ²à,çà,¥à,°à¹\200à,-à,µà,çà,\224à,-à,²à,\201à,²à,£</Text>
153:             <TextInput
154:                 placeholder='à,£à,²à,çà,¥à,°à¹\200à,-à,µà,çà,\224'
155:                 multiline={true}
156:                 style={styles.multiInput}
157:                 value={description_al_lv}
158:                 onChangeText={setDescription_al_lv}
159:             />
160:             <Text style={[styles.textStyle, {marginLeft: width*0.05}]}>à,£à,
161:             ²à,çà,¥à,°à¹\200à,-à,µà,çà,\224à,çà,²à,\227à,µà¹\210à,\225à¹\211à,-à,\207à,£à,±à,\232à,\233à
162:             ,£à,°à,\230à,²à,\231</Text>
163:             <TextInput
164:                 placeholder='à,£à,²à,çà,¥à,°à¹\200à,-à,µà,çà,\224'
165:                 multiline={true}
166:                 style={styles.multiInput}
167:                 value={medicine}
168:                 onChangeText={setMedicine}
169:             />
170:             <Pressable
171:                 style={[styles.button, styles.buttonSummit, {marginBottom: 1
172:                 00, opacity: enable ? 1 : 0.5}]}
173:                 onPress={update_lv}
174:                 >
175:                     <Text style={[styles.textStyle, {color: 'white'}]}>à,\225à,
176:                     \201à,¥à,\207</Text>
177:                 </Pressable>
178:             </View>
179:             </KeyboardAwareScrollView>
180:         </Modal>
181:     { /* modal 2 */ }
182:     <Modal
183:         animationType="slide"
184:         visible={modalVisible}
185:     >
186:         <KeyboardAwareScrollView enableOnAndroid={true}>
187:         <View isSafe style={[styles.container, {height: height, width: width
```

```
188:     ]}>
189:         <View style={styles.viewHeader}>
190:             <Pressable
191:                 onPress={() => {
192:                     setModalVisible(false)
193:                     setDescription('')
194:                     setTitle('')
195:                 }}
196:             >
197:                 <Ionicons name={'arrow-back-circle'} size={30} color={Co
198:                 lors.blue} />
199:             </Pressable>
200:             <Text style={styles.textHeader}>à,²à,£à¹\211à,²à,\207à,\231à
201:             ,±à,\224</Text>
202:         </View>
203:         <Text style={styles.textStyle, {marginLeft: width*0.05}}>à,«à,
204:         ±à,§à,\202à¹\211à,-</Text>
205:         <TextInput
206:             placeholder='à,«à,±à,§à,\202à¹\211à,-'
207:             style={styles.titleInput}
208:             value={title}
209:             onChangeText={setTitle}
210:         />
211:         <Text style={styles.textStyle, {marginLeft: width*0.05}}>à,£à,
212:         ²à,çà,¥à,°à¹\200à,-à,µà,çà,\224</Text>
213:         <TextInput
214:             placeholder='à,£à,²à,çà,¥à,°à¹\200à,-à,µà,çà,\224'
215:             multiline={true}
216:             style={styles.multiInput}
217:             value={description}
218:             onChangeText={setDescription}
219:         />
220:         {show &&
221:         <View style={{width: '100%'}}>
222:             <DateTimePicker
223:                 style={styles.picker}
224:                 testID="datePicker"
225:                 value={time}
226:                 mode={'date'}
227:                 is24Hour={true}
228:                 display="default"
229:                 onChange={onChangeDateTime}
230:             />
231:             <DateTimePicker
232:                 style={styles.picker}
233:                 testID="datePicker"
234:                 value={time}
235:                 mode={'time'}
236:                 is24Hour={true}
237:                 display="default"
238:                 onChange={onChangeDateTime}
239:             />
240:         </View>
241:     }
242:     <Pressable
243:         style={[styles.button, styles.buttonOpen]}
244:         onPress={() => setShow(true)}
245:     >
246:         <Text style={[styles.textStyle, {color: 'white'}]}>à¹\200à,¥
247:         à,•à,-à,\201à,§à,±à,\231à,\227à,µà¹\210à¹\201à,¥à,°à¹\200à,§à,¥à,²</Text>
248:     </Pressable>
249:     <Pressable
250:         style={[styles.button, styles.buttonSummit, {marginBottom: 1
251:         00, opacity: enable ? 1 : 0.5}]}
252:         onPress={createMeet}
253:     >
```

```

285:         paddingBottom: 5,
286:     }}
287:     onPress={() => navigation.navigate("MeetInfo", { data: d
ata })})
288:         >
289:             <Text style={styles.itemTitle}>{data.title}</Text>
290:             {/* (new Intl.DateTimeFormat("th-TH",{ dateStyle: 'full'
, timeStyle: 'short' })).format((new Date(data.item.time.seconds * 1000))).toString() */}
291:             <Text style={styles.itemTime}>{(new Intl.DateTimeFormat(
"th-TH",{ dateStyle: 'full', timeStyle: 'short' })).format((new Date(data.time.seconds * 1000
))).toString()}</Text>
292:             </Pressable>
293:         }}}
294:     </ScrollView>
295: </View>
296: )
297: }
298:
299: export default PatientInfo
300:
301: const styles = StyleSheet.create({
302:     container: {
303:         flex: 1,
304:         alignItems: 'center',
305:         backgroundColor: 'white',
306:         height: height
307:         // justifyContent: 'center'
308:     },
309:     content: {
310:         width: width*0.8,
311:         alignItems: 'center'
312:     },
313:     button: {
314:         height: 50,
315:         width: width*0.9,
316:         margin: 12,
317:         padding: 10,
318:         borderRadius: 5,
319:         justifyContent: 'center',
320:         alignItems: 'center',
321:         marginVertical: 10,
322:     },
323:     buttonOpen: {
324:         backgroundColor: Colors.orange,
325:     },
326:     buttonSummit: {
327:         backgroundColor: '#1597e5',
328:     },
329:     textStyle: {
330:         fontWeight: "bold",
331:         fontSize: 20,
332:     },
333:     textHeader: {
334:         fontWeight: 'bold',
335:         fontSize: 30,
336:         marginLeft: 20,
337:         marginBottom: 10
338:     },
339:     viewHeader: {
340:         width: width*0.9,
341:         flexDirection: 'row',
342:         justifyContent: 'flex-start',
343:         alignItems: 'center',
344:         alignSelf: 'center'
345:     },
346:     itemPatient:{

```

```
347:     width: width,
348:   },
349:   item:{
350:     height: 'auto',
351:     paddingHorizontal: width * 0.05,
352:     borderBottomWidth: 4,
353:     borderColor: '#f0f0f0',
354:     paddingBottom: 5,
355:     flexDirection: 'row',
356:   },
357:   imagePatient: {
358:     height: 70,
359:     width: 70,
360:   },
361:   image: {
362:     borderRadius: 5,
363:     marginVertical: 12,
364:     alignSelf: 'flex-start'
365:   },
366:   itemTitle:{
367:     fontSize: 20,
368:     fontWeight: '600',
369:   },
370:   itemTime: {
371:     fontSize: 16,
372:     fontWeight: '400'
373:   },
374:   itemViewText: {
375:     alignSelf: 'center',
376:     paddingLeft: 10
377:   },
378:   picker: {
379:     width: 90,
380:     marginHorizontal: (width - 90)/2,
381:     marginVertical: 5
382:   },
383:   titleInput:{
384:     height: 50,
385:     width: width*0.9,
386:     margin: 12,
387:     padding: 10,
388:     borderRadius: 5,
389:     fontSize: 20,
390:     backgroundColor: "#f7f7f7",
391:   },
392:   multiInput: {
393:     minHeight: 100,
394:     maxHeight: height*0.5,
395:     width: width*0.9,
396:     margin: 12,
397:     padding: 10,
398:     borderRadius: 5,
399:     fontSize: 20,
400:     backgroundColor: "#f7f7f7",
401:   },
402: })
```

```

1: import React, {useEffect} from 'react'
2: import { StyleSheet, Text, Dimensions, ScrollView, Pressable } from 'react-native'
3: import { View } from '../../components'
4: import { Colors, db } from '../../../config';
5: import { collection, onSnapshot } from 'firebase/firestore';
6: import { useRecoilValue, useRecoilState } from 'recoil';
7: import { sortcurrentPatientTests, currentPatientTests } from '../../../store';
8: import { Ionicons } from '@expo/vector-icons';
9: const {width, height} = Dimensions.get('window');
10:
11: const PatientTests = ({navigation, route}) => {
12:   const { patientInfo } = route.params;
13:
14:   const [e,setCurrentPatienTests] = useRecoilState(currentPatientTests);
15:   const sortCurrentPatientTests = useRecoilValue(sortcurrentPatientTests);
16:
17:   useEffect(() => {
18:     // effect
19:     const collectionTests = collection(db, "users", patientInfo.uid, "patient_tests");
20:     const unsubscribe = onSnapshot(collectionTests, (snapshot) => {
21:       setCurrentPatienTests(snapshot.docs.map((doc) => /**/{id: doc.id, ...do
c.data()})/**/));
22:     });
23:     return () => {
24:       // cleanup
25:       unsubscribe()
26:     }
27:   }, [])
28:
29:   return (
30:     <View isSafe style={styles.container}>
31:       <View style={styles.viewHeader}>
32:         <Pressable onPress={() => navigation.goBack()}>
33:           <Ionicons name='arrow-back-circle' size={30} color={Colors.blue} />
34:         </Pressable>
35:         <Text style={styles.textHeader}>&\234à,\201à,\227à,\201à,\232à,\232à,\227à,\224à,*à,-à,\232</Text>
36:       </View>
37:       <ScrollView style={{height: 'auto', width: width}}>
38:         <View style={{height: 4, width: width, backgroundColor: '#f0f0f0'}}>
39:           {sortCurrentPatientTests.map((data) => {
40:             return <Pressable
41:               key={data.id}
42:               style={{
43:                 height: 'auto',
44:                 minHeight: 70,
45:                 width: width,
46:                 paddingHorizontal: width * 0.05,
47:                 borderBottomWidth: 4,
48:                 borderColor: '#f0f0f0',
49:                 paddingBottom: 5,
50:                 justifyContent: 'center'
51:               }}
52:               onPress={() => navigation.navigate("TestInfo", { data: d
ata })}
53:             >
54:               <Text style={styles.itemTitle}>{(new Intl.DateTimeFormat
("th-TH",{ dateStyle: 'full', timeStyle: 'short' }).format((new Date(data.user_time.seconds
* 1000))).toString())</Text>
55:             </Pressable>
56:           }}}
57:       </ScrollView>
58:     </View>

```

```

59:     )
60: }
61:
62: export default PatientTests
63:
64: const styles = StyleSheet.create({
65:   container: {
66:     flex: 1,
67:     alignItems: 'center',
68:     backgroundColor: 'white',
69:     height: height
70:     // justifyContent: 'center'
71:   },
72:   content: {
73:     width: width*0.8,
74:     alignItems: 'center'
75:   },
76:   button: {
77:     height: 50,
78:     width: width*0.9,
79:     margin: 12,
80:     padding: 10,
81:     borderRadius: 5,
82:     justifyContent: 'center',
83:     alignItems: 'center',
84:     marginVertical: 10,
85:   },
86:   buttonOpen: {
87:     backgroundColor: Colors.orange,
88:   },
89:   buttonSummit: {
90:     backgroundColor: '#1597e5',
91:   },
92:   textHeader: {
93:     fontWeight: 'bold',
94:     fontSize: 30,
95:     marginLeft: 20,
96:     marginBottom: 10
97:   },
98:   viewHeader: {
99:     width: width*0.9,
100:     flexDirection: 'row',
101:     justifyContent: 'flex-start',
102:     alignItems: 'center',
103:     alignSelf: 'center'
104:   },
105:   itemTitle:{
106:     fontSize: 20,
107:     fontWeight: '600',
108:   },
109: })

```



```

1: import React, { useState, useEffect } from 'react'
2: import { StyleSheet, Text, Image, Dimensions, ScrollView, TextInput, Pressable, Alert, RefreshControl } from 'react-native'
3: import { View, LoadingIndicator } from '../components'
4: import { Colors, db, auth } from '../config';
5: import { collection, query, startAt, endAt, orderBy, onSnapshot, doc, setDoc, getDoc, getDocs } from "firebase/firestore";
6: import { allPatientsAtom, userInfoAtom } from '../store';
7: import { useRecoilState, useRecoilValue } from 'recoil';
8: const {width, height} = Dimensions.get('window');
9:
10: // add patient and list patient
11: export const DoctorHomeScreen = ({navigation}) => {
12:
13:   // patients of doctor
14:   const [allPatients, setAllPatients] = useRecoilState(allPatientsAtom);
15:   const doctorInfo = useRecoilValue(userInfoAtom);
16:
17:   const [isLoading, setLoading] = useState(false);
18:   const [loadingPatients, setLoadingPatients] = useState(false);
19:
20:   const [searchPatient, setSearchPatient] = useState([]);
21:   const [search, setSearch] = useState('');
22:
23:   const [refreshing, setRefreshing] = React.useState(false);
24:
25:   useEffect(() => {
26:     // effect
27:     loadOptions(search)
28:   }, [search])
29:
30:   useEffect(() => {
31:     // effect
32:     setLoadingPatients(true)
33:     const patientsRef = collection(db, "users", auth.currentUser.uid, "all-patients")
34:
35:     const unsubPatients = onSnapshot(patientsRef, (snapshot) => {
36:       setAllPatients([]);
37:       // console.log(snapshot.docs.map((item) => item.data()));
38:       snapshot.docs.map(async (item) => {
39:         const patient_uid = item.data().uid
40:         const patientRef = doc(db, "users", patient_uid)
41:         const docsnap = await getDoc(patientRef)
42:         console.log('docsnap', docsnap.data());
43:         setAllPatients((prev) => [...prev, {...docsnap.data()}]);
44:       })
45:       // console.log('fetch patients', allPatients);
46:       setLoadingPatients(false)
47:     });
48:
49:     return () => {
50:       // cleanup
51:       unsubPatients();
52:     }, []
53:
54:     const onRefresh = React.useCallback(() => {
55:       setRefreshing(true);
56:       loadOptions(search);
57:       refreshUserData()
58:       setRefreshing(false);
59:     }, []);
60:
61:     const refreshUserData = async () => {
62:       console.log('refresh');
63:       setLoadingPatients(true)

```

```

64: setAllPatients([]);
65: const patientsRef = collection(db, "users", auth.currentUser.uid, "all-patients")

66: const allpatients = await getDocs(patientsRef);
67: allpatients.docs.map(async (item) => {
68:   const patient_uid = item.data().uid
69:   const patientRef = doc(db, "users", patient_uid)
70:   const docsnap = await getDoc(patientRef)
71:   // console.log('docsnap', docsnap.data());
72:   setAllPatients((prev) => [...prev, {...docsnap.data()}]);
73: })
74: // console.log('refetch patients', allPatients);
75: setLoadingPatients(false)
76: }
77:
78: const loadOptions = async (inputValue) => {
79:   setLoading(true);
80:   setSearchPatient([]);
81:   // console.log('click query');
82:   const userRef = collection(db, "users");
83:   const q = query(userRef, orderBy("uid"), startAt(inputValue), endAt(inputValue+"
uf8ff")));
84:   const querySnapshot = await getDocs(q);
85:   const users = querySnapshot.docs.filter(doc => doc.data().person_type == "patien
t");
86:   // console.log('found user', users.map((doc) => doc.data()));
87:   setSearchPatient(users.map((doc) => {
88:     return {
89:       id: doc.id,
90:       ...doc.data()
91:     }
92:   }))
93:   setLoading(false);
94: }
95:
96: const checkSelectPatientInPatients = (uid) => {
97:   if (allPatients.length === 0) {
98:     return true
99:   }
100:
101:   const findPatient = allPatients.filter(e => e.uid === uid);
102:   // console.log("find patient ==>", findPatient);
103:
104:   if (findPatient.length === 0) {
105:     return true
106:   } else {
107:     return false
108:   }
109: }
110:
111: const addPatient = async (patient) => {
112:   // check patient that is not added in lists
113:   console.log('clicked patient ==> ', patient.firstname, '\n      uid ==> ', pat
ient.uid);
114:   const r = checkSelectPatientInPatients(patient.uid)
115:   if (r === true) {
116:     const docRef = doc(db, "users", auth.currentUser.uid, "all-patients", patient.
uid);
117:     await setDoc(docRef, {uid: patient.uid}, {merge: true}).then(() => {
118:       alertMessage("à\200à,"à,\207à,\210à,"à,\211à,\231", "\234à,\211à
,\233à,\210à,\226à,\201à,\200à,\236à,\210à,\210à,\211à,\211à,\211à,\210à,-à,\231à.\231à");
119:     })
120:   } else {
121:     alertMessage("à\204à,\210à,"à,\2à,\2à,\2à,\226à,\200à,\236à,\210à,\210à,\211à,\233à,\210à,\226à,\201à,\200à,\236à,\210à,\210à,\211à,\211à,\210à,-à,\231à.\231à");

```

```

\231à¹\211à,²à,\231à,µà¹\211");
122:   }
123: }
124:
125: const alertMessage = (title, message) => {
126:   Alert.alert(
127:     title,
128:     message,
129:     [
130:       { text: "à,\225à,\201à,¥à,\207", onPress: () => {} }
131:     ]
132:   );
133: }
134:
135:
136: const gotoPatientInfo = (patient) => {
137:   console.log("naviagte to ", patient);
138:   navigation.navigate("Info", {patientInfo: patient, doctorInfo: doctorInfo});
139: }
140:
141: const RenderPatients = () => {
142:   return allPatients.map(patient => {
143:     return <Pressable
144:       key={patient.uid}
145:       style={[styles.item, styles.itemPatient]}
146:       onPress={() => {
147:         // navigate to patientInfo
148:         gotoPatientInfo(patient)
149:       }}
150:     >
151:       {!patient.urlImage ?
152:         <Image
153:           style={[styles.image, styles.imagePatient]}
154:           source={require('.././assets/avatar.webp')}
155:         /> :
156:         <Image
157:           style={[styles.image, styles.imagePatient]}
158:           source={{uri: patient.urlImage}}
159:         />
160:       <View style={styles.itemViewText}>
161:         <Text style={styles.itemTitle}>{patient.firstname} {patient.lastna
me}</Text>
162:         <Text style={styles.itemTitle}>à,-à,²à,çà,,: {patient.age}à,\233à,
µ à¹\200à,\236à,": {patient.sex_type === "female" ? "à,«à,\215à,´à,\207" : "à,\212à,²à,ç"}</
Text>
163:         <Text style={styles.itemTitle, {fontSize: 14, paddingTop: 3}}>ui
d: {patient.uid}</Text>
164:       </View>
165:     </Pressable>
166:   )}
167: }
168:
169: return (
170:   <View isSafe style={styles.container}>
171:     <Text style={styles.textHeader}>à,£à,²à,çà,\212à,•à¹\210à,-à,\204à,\231à¹\204à
,\202à¹\211</Text>
172:     <View style={styles.viewHeader}>
173:       <TextInput
174:         onChangeText={setSearch}
175:         placeholder="à,\236à,´à,jà,\236à¹\214 uid à,\202à,-à,\207à,\204à,\231à¹
\203à,\202à¹\211"
176:         value={search}
177:         style={styles.titleInput}
178:       />
179:       <Pressable
180:         style={[styles.buttonSummit, styles.button]}

```

```

181:     onPress={() => {
182:       loadOptions(search)
183:     }}
184:   >
185:     <Text style={styles.textBtn}>à,\204à¹\211à,\231à,«à,²</Text>
186:   </Pressable>
187: </View>
188: {!isLoading ? <View style={styles.scrollView}>
189:   {searchPatient.length == 0 ? <Text>à¹\204à,jà¹\210à,\236à,\232à,\234à,¹à¹
\211à,\233à¹\210à,§à,çà,\227à,µà¹\210 uid à,\202à,¶à¹\211à,\231à,\225à¹\211à,\231à,\224à¹
\211à,§à,ç {search}</Text>
190:   : (
191:     <ScrollView>
192:       <View style={{height: 4, width: width*0.8, backgroundColor: '#f0f0f0'}}
193:     ></View>
194:     {searchPatient.map(patient => {
195:       return <View
196:         key={patient.uid}
197:         style={[styles.item, styles.itemSearch]}
198:       >
199:         <Image
200:           style={[styles.image, styles.imageSearch]}
201:           source={{uri: patient.urlImage}}
202:         />
203:         <View style={styles.itemViewText}>
204:           <Text style={styles.itemTitle}>{patient.firstname}</Text>
205:           <Text style={styles.itemTitle}>{patient.lastname}</Text>
206:         </View>
207:         {checkSelectPatientInPatients(patient.uid) ?
208:           <Pressable
209:             style={[styles.buttonOpen, styles.button]}
210:             onPress={() => {
211:               addPatient(patient)
212:             }}
213:           >
214:             <Text style={styles.textBtn}>à¹\200à,\236à,´à¹\210à,jà,
\204à,\231à¹\204à,\202à¹\211</Text>
215:           </Pressable>
216:         :
217:         <Pressable
218:           style={[styles.button, {backgroundColor: "#00af00"}]}
219:           onPress={() => {
220:             // navigate to patientInfo
221:             gotoPatientInfo(patient)
222:           }}
223:         >
224:           <Text style={styles.textBtn}>à¹\204à,\233à,\227à,µà¹\210à
,\202à¹\211à,-à,jà,¹à,¥</Text>
225:         </Pressable>
226:       }
227:     )
228:   }
229: </ScrollView>
230: )
231: }
232: </View>: <View style={styles.scrollView}><LoadingIndicator/></View>
233: <View style={{height: 5, width: width, backgroundColor: '#1597e5'}}></View>
234: {!loadingPatients ?
235:   <ScrollView
236:     refreshControl={
237:       <RefreshControl
238:         refreshing={refreshing}
239:         onRefresh={onRefresh}
240:       />
241:     }

```

```
242:         >
243:         <View style={{height: 4, width: width, backgroundColor: '#f0f0f0'}}></View>
>
244:         <RenderPatients/>
245:         </ScrollView>) : <LoadingIndicator/>)
246:     </View>
247:   );
248: };
249:
250: const styles = StyleSheet.create({
251:   container: {
252:     flex: 1,
253:     alignItems: 'center',
254:     backgroundColor: 'white',
255:     height: height
256:     // justifyContent: 'center'
257:   },
258:   scrollView: {
259:     width: width*0.8,
260:     minHeight: 70,
261:     maxHeight: 200,
262:     alignItems: 'center',
263:     justifyContent: "center",
264:     borderRadius: 5,
265:     backgroundColor: "#efefef",
266:     marginVertical: 12,
267:     // shadow
268:     shadowRadius: 2.62,
269:     shadowOpacity: 0.23,
270:     shadowColor: "#000",
271:     shadowOffset: {
272:       width: 0,
273:       height: 2
274:     },
275:     elevation: 4,
276:   },
277:   button: {
278:     justifyContent: 'center',
279:     width: 'auto',
280:     height: 50,
281:     borderRadius: 5,
282:     paddingHorizontal: 10,
283:   },
284:   buttonOpen: {
285:     backgroundColor: Colors.orange,
286:   },
287:   buttonSummit: {
288:     backgroundColor: '#1597e5',
289:   },
290:   textBtn: {
291:     alignSelf: 'center',
292:     fontWeight: 'bold',
293:     color: 'white'
294:   },
295:   textHeader: {
296:     fontWeight: 'bold',
297:     fontSize: 30,
298:     marginLeft: 20,
299:     marginBottom: 10
300:   },
301:   viewHeader: {
302:     width: width*0.9,
303:     flexDirection: 'row',
304:     justifyContent: 'space-evenly',
305:     alignItems: 'center',
306:     alignSelf: 'center'
```

```
307:   },
308:   titleInput: {
309:     height: 50,
310:     width: width*0.7,
311:     padding: 10,
312:     borderRadius: 5,
313:     fontSize: 20,
314:     backgroundColor: "#f7f7f7",
315:   },
316:   itemSearch: {
317:     width: width*0.8,
318:     justifyContent: 'space-evenly',
319:     alignItems: 'center',
320:   },
321:   itemPatient: {
322:     width: width,
323:   },
324:   item: {
325:     height: 'auto',
326:     paddingHorizontal: width * 0.05,
327:     borderBottomWidth: 4,
328:     borderColor: '#f0f0f0',
329:     paddingBottom: 5,
330:     flexDirection: 'row',
331:   },
332:   imageSearch: {
333:     height: 50,
334:     width: 50,
335:   },
336:   imagePatient: {
337:     height: 70,
338:     width: 70,
339:   },
340:   image: {
341:     borderRadius: 5,
342:     marginVertical: 12,
343:     alignSelf: 'flex-start'
344:   },
345:   itemTitle: {
346:     fontSize: 20,
347:     fontWeight: '600',
348:   },
349:   itemViewText: {
350:     alignSelf: 'center',
351:     paddingLeft: 10
352:   },
353: })
354:
```



```
1: import { LoginScreen } from './LoginScreen';
2: import { SignupScreen } from './SignupScreen';
3: import { DoctorHomeScreen } from './doctor/DoctorHomeScreen';
4: import { CaretakerHomeScreen } from './caretaker/CaretakerHomeScreen';
5: import { PatientStatScreen } from './patient/PatientStatScreen';
6: import { PatientSocialScreen } from './patient/PatientSocialScreen';
7: import { ForgotPasswordScreen } from './ForgotPasswordScreen';
8: import { UpdateUser } from './UpdateUser';
9: import { UserScreen } from './UserScreen';
10:
11: export {
12:     LoginScreen,
13:     SignupScreen,
14:     DoctorHomeScreen,
15:     CaretakerHomeScreen,
16:     PatientStatScreen,
17:     PatientSocialScreen,
18:     ForgotPasswordScreen,
19:     UpdateUser,
20:     UserScreen
21: };
```



```

1: import React, { useState } from 'react';
2: import { Text, StyleSheet } from 'react-native';
3: import { Formik } from 'formik';
4: import { KeyboardAwareScrollView } from 'react-native-keyboard-aware-scroll-view';
5: import { View, TextInput, Logo, Button, FormErrorMessage } from '../components';
6: import { DropDownList } from 'react-native-ultimate-modal-picker';
7:
8: import { useTogglePasswordVisibility } from '../hooks';
9: import { signupValidationSchema } from '../utils';
10: import { createUserWithEmailAndPassword } from 'firebase/auth';
11: import { Images, Colors, auth, db } from '../config';
12: import { doc, setDoc } from "firebase/firestore";
13:
14: export const SignupScreen = ({ navigation }) => {
15:   const [errorState, setErrorState] = useState('');
16:
17:   const {
18:     passwordVisibility,
19:     handlePasswordVisibility,
20:     rightIcon,
21:     handleConfirmPasswordVisibility,
22:     confirmPasswordIcon,
23:     confirmPasswordVisibility
24:   } = useTogglePasswordVisibility();
25:
26:   const handleSignup = async values => {
27:     const { email, password, person_type, age, sex_type, name, lastname } = values;
28:     await createUserWithEmailAndPassword(auth, email, password).then(async (userCred
entia) => {
29:       const docRef = doc(db, "users", auth.currentUser.uid)
30:       const payload = {
31:         firstname: name,
32:         lastname: lastname,
33:         age: age,
34:         sex_type: sex_type,
35:         person_type: person_type,
36:         email: auth.currentUser.email,
37:         uid: auth.currentUser.uid,
38:         urlImage: ''
39:       };
40:       console.log(payload);
41:       console.log('click');
42:
43:       //custom
44:       if (person_type == 'patient') {
45:         // add test point
46:         await setDoc(docRef, { isTest: false, allergy: "", address: "", like: "", un
like: "", alzheimer_lv: "", medicine: "" }, { merge: true }).then(() => {
47:           console.log('add patient test point');
48:         }).catch(error => setErrorState(error.message));
49:       }
50:
51:       // bug
52:       // fix https://github.com/firebase/firebase-js-sdk/issues/5667#issuecomment-95
2079600
53:       // fix bug with const db = initializeFirestore(firebaseApp, {useFetchStreams:
false})
54:       await setDoc(docRef, payload, { merge: true }).then(() => {
55:         console.log('create user success');
56:       }).catch(error => setErrorState(error.message));
57:
58:     }).catch(error =>{
59:       console.log('create user error', error.message);
60:       setErrorState(error.message);
61:     });
62:   };

```

```

63:
64:   return (
65:     <View isSafe style={styles.container}>
66:       <KeyboardAwareScrollView enableOnAndroid={true}>
67:         {/** LogoContainer: consits app logo and screen title */}
68:         <View style={styles.logoContainer}>
69:           <Logo uri={Images.logo} />
70:           <Text style={styles.screenTitle}>à,²à,\207à,\232à,±à,\215à,
\212à,²à,\234à,¹à'\211à'\203à,\212à¹\211à¹\203à,«à,ìà'\210</Text>
71:         </View>
72:         {/** Formik Wrapper */}
73:         <Formik
74:           initialValues={{
75:             email: '',
76:             password: '',
77:             confirmPassword: '',
78:             person_type: '',
79:             age: '',
80:             sex_type: '',
81:             name: '',
82:             lastname: ''
83:           }}
84:           validationSchema={signupValidationSchema}
85:           onSubmit={values => handleSignup(values)}
86:         >
87:           {({
88:             values,
89:             touched,
90:             errors,
91:             handleChange,
92:             handleSubmit,
93:             handleBlur
94:           }) => (
95:             <
96:               <TextInput
97:                 name='email'
98:                 leftIconName='email'
99:                 placeholder='Enter email'
100:                 autoCapitalize='none'
101:                 keyboardType='email-address'
102:                 textContentType='emailAddress'
103:                 autoFocus={true}
104:                 value={values.email}
105:                 onChangeText={handleChange('email')}
106:                 onBlur={handleBlur('email')}
107:               />
108:               <FormErrorMessage error={errors.email} visible={touched.email} />
109:               <TextInput
110:                 name='password'
111:                 leftIconName='key-variant'
112:                 placeholder='Enter password'
113:                 autoCapitalize='none'
114:                 autoComplete={false}
115:                 secureTextEntry={passwordVisibility}
116:                 textContentType='newPassword'
117:                 rightIcon={rightIcon}
118:                 handlePasswordVisibility={handlePasswordVisibility}
119:                 value={values.password}
120:                 onChangeText={handleChange('password')}
121:                 onBlur={handleBlur('password')}
122:               />
123:               <FormErrorMessage
124:                 error={errors.password}
125:                 visible={touched.password}
126:               />
127:               <TextInput

```

```
128:         name='confirmPassword'
129:         leftIconName='key-variant'
130:         placeholder='Enter password'
131:         autoCapitalize='none'
132:         autoCorrect={false}
133:         secureTextEntry={confirmPasswordVisibility}
134:         textContentType='password'
135:         rightIcon={confirmPasswordIcon}
136:         handlePasswordVisibility={handleConfirmPasswordVisibility}
137:         value={values.confirmPassword}
138:         onChangeText={handleChange('confirmPassword')}
139:         onBlur={handleBlur('confirmPassword')}
140:     />
141:     <FormErrorMessage
142:       error={errors.confirmPassword}
143:       visible={touched.confirmPassword}
144:     />
145:     <TextInput
146:       name='name_lastname'
147:       placeholder='à, \201à, fà, , à, \223à, ²à, fà, °à, \232à, , à, \212à, *à¹\210à, -'
148:       autoCapitalize='none'
149:       autoFocus={true}
150:       value={values.name}
151:       onChangeText={handleChange('name')}
152:       onBlur={handleBlur('name')}
153:     />
154:     <FormErrorMessage
155:       error={errors.name}
156:       visible={touched.name}
157:     />
158:     <TextInput
159:       name='lastname'
160:       placeholder='à, \201à, fà, , à, \223à, ²à, fà, °à, \232à, , à, \231à, ²à, à, à,
\201à, , à, ¥'
161:       autoCapitalize='none'
162:       autoFocus={true}
163:       value={values.lastname}
164:       onChangeText={handleChange('lastname')}
165:       onBlur={handleBlur('lastname')}
166:     />
167:     <FormErrorMessage
168:       error={errors.lastname}
169:       visible={touched.lastname}
170:     />
171:     <TextInput
172:       name='age'
173:       placeholder='à¹\203à, *à¹\210à, -à, ²à, çà, , à, \202à, -à, \207à, \204à, , à,
\223'
174:       autoCapitalize='none'
175:       autoFocus={true}
176:       value={values.age}
177:       onChangeText={handleChange('age')}
178:       onBlur={handleBlur('age')}
179:     />
180:     <FormErrorMessage
181:       error={errors.age}
182:       visible={touched.age}
183:     />
184:     <DropDownList
185:       items=[
186:         { label: 'à¹\201à, \236à, \227à, çà¹\214', value: 'doctor' },
187:         { label: 'à, \234à, , à¹\211à, \233à¹\210à, $à, ç¹', value: 'patient' },
188:         { label: 'à, \234à, , à¹\211à, \224à, ¹à¹\201à, ¥', value: 'caretaker' }
189:       ]
190:       title="à¹\200à, ¥à, *à, -à, \201à, «à, \231à¹\211à, ²à, \227à, uà¹\210"
```

```
191:         onChange={handleChange('person_type')}
192:       />
193:     <FormErrorMessage
194:       error={errors.person_type}
195:       visible={touched.person_type}
196:     />
197:     <DropDownList
198:       title="à¹\200à, ¥à, *à, -à, \201à¹\200à, \236à, ""
199:       items=[
200:         { label: 'à, «à, \215à, 'à, \207', value: 'female' },
201:         { label: 'à, \212à, ²à, ç¹', value: 'male' },
202:       ]
203:       onChange={handleChange('sex_type')}
204:     />
205:     <FormErrorMessage
206:       error={errors.sex_type}
207:       visible={touched.sex_type}
208:     />
209:     { /* Display Screen Error Mesages */ }
210:     {errorState !== '' ? (
211:       <FormErrorMessage error={errorState} visible={true} />
212:     ) : null}
213:
214:     { /* Signup button */ }
215:     <Button style={styles.button} onPress={handleSubmit}>
216:       <Text style={styles.buttonText}>à, *à, fà¹\211à, ²à, \207à, \232à, à, \215
à, \212à, uà, \234à, ¹à¹\211à¹\203à, \212à¹\211</Text>
217:     </Button>
218:   </>
219:   )}
220: </Formik>
221: { /* Button to navigate to Login screen */ }
222: <Button
223:   style={styles.borderlessButtonContainer}
224:   borderless
225:   title={'à¹\200à, \233à¹\207à, \231à, *à, à, ²à, \212à, 'à, \201à¹\201à, ¥à¹\211à, $
' }
226:   onPress={() => navigation.navigate('Login')}
227: />
228: </KeyboardAwareScrollView>
229: </View>
230: );
231: };
232:
233: const styles = StyleSheet.create({
234:   container: {
235:     flex: 1,
236:     backgroundColor: Colors.white,
237:     paddingHorizontal: 12
238:   },
239:   logoContainer: {
240:     alignItems: 'center'
241:   },
242:   screenTitle: {
243:     fontSize: 32,
244:     fontWeight: '700',
245:     color: Colors.black,
246:     paddingTop: 20
247:   },
248:   button: {
249:     width: '100%',
250:     justifyContent: 'center',
251:     alignItems: 'center',
252:     marginTop: 8,
253:     backgroundColor: Colors.orange,
254:     padding: 10,
```



```
255:     borderRadius: 8
256:   },
257:   buttonText: {
258:     fontSize: 20,
259:     color: Colors.white,
260:     fontWeight: '700'
261:   },
262:   borderlessButtonContainer: {
263:     marginTop: 16,
264:     marginBottom: 32,
265:     alignItems: 'center',
266:     justifyContent: 'center'
267:   }
268: });
269:
```



```
1: import React from 'react'
2: import { StyleSheet, Text, Image, ScrollView, Dimensions } from 'react-native'
3: import { Colors } from '../config';
4: import { Button, View, Logo } from '../components'
5:
6: import { useRecoilValue } from 'recoil';
7: import { userInfoAtom } from '../store';
8: import { auth, Images } from '../config';
9: import { signOut } from '@firebase/auth';
10:
11: const {height, width} = Dimensions.get('window');
12:
13: export const UserScreen = ({ navigation }) => {
14:   const userInfo = useRecoilValue(userInfoAtom);
15:   return (
16:     <ScrollView>
17:       <View isSafe style={styles.container}>
18:         <Text style={styles.screenTitle}>à,\202à¹\211à,-à,jà,¹à,¥à,\234à,¹à¹\211
à¹\203à,\212à¹\211</Text>
19:         </* <Text>{JSON.stringify(userInfo)}</Text> */>
20:         {userInfo.urlImage ?
21:           <Image
22:             source={{ uri: userInfo.urlImage }}
23:             style={{
24:               width: 200,
25:               height: 200,
26:               borderRadius: 100,
27:               marginVertical: 10,
28:               alignSelf: 'center'
29:             }}
30:           /> :
31:           <Logo
32:             uri={Images.logo}
33:             style={{alignSelf: 'center'}}
34:           />
35:         }
36:         <Text style={styles.text}>userID: {userInfo.uid}</Text>
37:         <Text style={styles.text}>à,\212à,·à¹\210à,-: {userInfo.firstname} {user
Info.lastname}</Text>
38:         <Text style={styles.text}>email: {userInfo.email}</Text>
39:         <Text style={styles.text}>à,-à,²à,çà,,: {userInfo.age} à,\233à,u</Text>
40:         <Text style={styles.text}>à¹\200à,\236à,": {userInfo.sex_type == 'female
' ? 'à,«à,\215à,¹à,\207' : 'à,\212à,²à,ç'}</Text>
41:
42:         </* patient */>
43:         {userInfo.person_type === 'patient' ?
44:           <>
45:             <Text style={styles.text, {color: (userInfo.address !== "") ? "blac
k" : "red"}}>à,\227à,µà¹\210à,-à,çà,¹à¹\210: {userInfo.address !== "" ? userInfo.address :
"à,\201à,£à,à,\223à,²à,£à,°à,\232à,,"}</Text>
46:             <Text style={styles.text, {color: (userInfo.like !== "") ? "black"
: "red"}}>à,à,¹à¹\210à,\207à,\227à,µà¹\210à,\212à,-à,\232: {userInfo.like !== "" ? userInf
o.like : "à,\201à,£à,à,\223à,²à,£à,°à,\232à,,"}</Text>
47:             <Text style={styles.text, {color: (userInfo.unlike !== "") ? "black
" : "red"}}>à,à,¹à¹\210à,\207à,\227à,µà¹\210à¹\204à,jà¹\210à,\212à,-à,\232: {userInfo.unli
ke !== "" ? userInfo.unlike : "à,\201à,£à,à,\223à,²à,£à,°à,\232à,,"}</Text>
48:             <Text style={styles.text, {color: (userInfo.allergy !== "") ? "blac
k" : "red"}}>à,à,¹à¹\210à,\207à,\227à,µà¹\210à¹\201à,\236à¹\211: {userInfo.allergy !== ""
? userInfo.allergy : "à,\201à,£à,à,\223à,²à,£à,°à,\232à,,"}</Text>
49:             <Text style={styles.text, {color: (userInfo.alzheimer_lv !== "") ?
"black" : "red"}}>à,£à,°à,çà,°à,-à,²à,\201à,²à,£: {userInfo.alzheimer_lv !== "" ? userInfo.
alzheimer_lv : "à¹\203à,«à¹\211à¹\201à,\236à,\227à,çà¹\214à,\233à,£à,°à¹\200à,jà,¹à,\223"}</
Text>
50:             <Text style={styles.text, {color: (userInfo.medicine !== "") ? "bla
ck" : "red"}}>à,çà,²à,\227à,µà¹\210à,\225à¹\211à,-à,\207à,\227à,²à,\231: {userInfo.medicine
!="" ? userInfo.medicine : "à¹\203à,«à¹\211à¹\201à,\236à,\227à,çà¹\214à,\233à,£à,°à¹\200à
```

```
,jà,¹à,\223"}</Text>
51:       </> :
52:       null}
53:
54:       </* patient test */>
55:       {userInfo.person_type === 'patient' ?
56:         <Button
57:           style={styles.borderlessButtonContainer}
58:           borderless
59:           title={'à,\227à,³à¹\201à,\232à,\232à,\227à,\224à,·à,-à,\232 moca'}
60:           onPress={() => navigation.navigate('DoTestMOCA')}
61:         /> :
62:         null}
63:       {userInfo.person_type === 'patient' ?
64:         <Button
65:           style={styles.borderlessButtonContainer}
66:           borderless
67:           title={'à,\227à,³à¹\201à,\232à,\232à,\227à,\224à,·à,-à,\232 à,\204à,
³à,\226à,²à,ì'}
68:           onPress={() => navigation.navigate('DoSimpleTest')}
69:         /> :
70:         null}
71:
72:         <Button
73:           style={styles.borderlessButtonContainer}
74:           borderless
75:           title={'à¹\200à,\233à,¥à,µà¹\210à,çà,\231à¹\201à,\233à,¥à,\207à,\202
à¹\211à,-à,jà,¹à,¥à,\234à,¹à¹\211à¹\203à,\212à¹\211'}
76:           onPress={() => navigation.navigate('UpdateUser')}
77:         />
78:         <Button
79:           style={styles.borderlessButtonContainer}
80:           onPress={() => {
81:             signOut(auth);
82:           }}
83:           borderless
84:           title={'à,-à,-à,\201à,\210à,²à,\201à,£à,°à,\232à,\232"}
85:         />
86:       </View>
87:     </ScrollView>
88:   )
89: }
90:
91: const styles = StyleSheet.create({
92:   container: {
93:     flex: 1,
94:     backgroundColor: Colors.white,
95:     alignItems: 'flex-start',
96:     minHeight: height,
97:     paddingHorizontal: width * 0.1
98:   },
99:   screenTitle: {
100:     fontSize: 32,
101:     fontWeight: '700',
102:     color: Colors.black,
103:     paddingTop: 20,
104:     alignSelf: 'center'
105:   },
106:   borderlessButtonContainer: {
107:     marginTop: 16,
108:     marginBottom: 16,
109:     alignItems: 'center',
110:     justifyContent: 'center',
111:     alignSelf: 'center'
112:   },
113:   text: {
```

```
114:     fontSize: 20,  
115:   }  
116: });
```

```

1: import React, { useState } from 'react';
2: import { Text, StyleSheet } from 'react-native';
3: import { Formik } from 'formik';
4: import { signInWithEmailAndPassword } from 'firebase/auth';
5: import { KeyboardAwareScrollView } from 'react-native-keyboard-aware-scroll-view';
6:
7: import { View, TextInput, Logo, Button, FormErrorMessage } from '../components';
8: import { Images, Colors, auth } from '../config';
9: import { useTogglePasswordVisibility } from '../hooks';
10: import { loginValidationSchema } from '../utils';
11:
12: export const LoginScreen = ({ navigation }) => {
13:   const [errorState, setErrorState] = useState('');
14:   const { passwordVisibility, handlePasswordVisibility, rightIcon } =
15:     useTogglePasswordVisibility();
16:
17:   const handleLogin = values => {
18:     const { email, password } = values;
19:     signInWithEmailAndPassword(auth, email, password).catch(error =>
20:       setErrorState(error.message)
21:     );
22:   };
23:   return (
24:     <>
25:       <View isSafe style={styles.container}>
26:         <KeyboardAwareScrollView enableOnAndroid={true}>
27:           { /* LogoContainer: consists app logo and screen title */ }
28:           <View style={styles.logoContainer}>
29:             <Logo uri={Images.logo} />
30:             <Text style={styles.screenTitle}>à, ÿà, \207à, \212à, •à¹\210à, -à¹\200à, \202
à¹\211à, ºà¹\203à, \212à¹\211à, \207à, ºà, \231 merzheial</Text>
31:           </View>
32:           <Formik
33:             initialValues={{
34:               email: '',
35:               password: ''
36:             }}
37:             validationSchema={loginValidationSchema}
38:             onSubmit={values => handleLogin(values)}
39:           >
40:             {({
41:               values,
42:               touched,
43:               errors,
44:               handleChange,
45:               handleSubmit,
46:               handleBlur
47:             }) => (
48:               <>
49:                 { /* Input fields */ }
50:                 <TextInput
51:                   name='email'
52:                   leftIconName='email'
53:                   placeholder='à, \201à, fà, .à, \223à, ºà¹\203à, ºà¹\210 email'
54:                   autoCapitalize='none'
55:                   keyboardType='email-address'
56:                   textContentType='emailAddress'
57:                   autoFocus={true}
58:                   value={values.email}
59:                   onChangeText={handleChange('email')}
60:                   onBlur={handleBlur('email')}
61:                 />
62:                 <FormErrorMessage
63:                   error={errors.email}
64:                   visible={touched.email}
65:                 />

```

```

66:         <TextInput
67:             name='password'
68:             leftIconName='key-variant'
69:             placeholder='à,\201à,à,\223à,\2à\203à,\2à\210à,à,\2à,\2à,\234à\210à,\2à,\231'
70:             autoCapitalize='none'
71:             autoComplete={false}
72:             secureTextEntry={passwordVisibility}
73:             textContentType='password'
74:             rightIcon={rightIcon}
75:             handlePasswordVisibility={handlePasswordVisibility}
76:             value={values.password}
77:             onChangeText={handleChange('password')}
78:             onBlur={handleBlur('password')}
79:         />
80:         <FormErrorMessage
81:             error={errors.password}
82:             visible={touched.password}
83:         />
84:         { /* Display Screen Error Messages */ }
85:         {errorState !== '' ? (
86:             <FormErrorMessage error={errorState} visible={true} />
87:         ) : null}
88:         { /* Login button */ }
89:         <Button style={styles.button} onPress={handleSubmit}>
90:             <Text style={styles.buttonText}>à,\207à,\212à,\2à\210à,\2à\200à,\202à\211à,\2à\203à,\212à\211à,\207à,\2à,\231</Text>
91:         </Button>
92:     </>
93: )}
94: </Formik>
95: { /* Button to navigate to SignupScreen to create a new account */ }
96: <Button
97:     style={styles.borderlessButtonContainer}
98:     borderless
99:     title={'à,\2à,\207à,\232à,\2à,\215à,\212à,\2à,\234à,\2à\211à\203à,\212à\211à\203à,\2à,\2à\210'}
100:     onPress={() => navigation.navigate('Signup')}
101: />
102: <Button
103:     style={styles.borderlessButtonContainer}
104:     borderless
105:     title={'à,\2à,\2à,\2à,\2à,\2à,\234à\210à,\2à,\231'}
106:     onPress={() => navigation.navigate('ForgotPassword')}
107: />
108: </KeyboardAwareScrollView>
109: </View>
110: </>
111: );
112: };
113:
114: const styles = StyleSheet.create({
115:     container: {
116:         flex: 1,
117:         backgroundColor: Colors.white,
118:         paddingHorizontal: 12
119:     },
120:     logoContainer: {
121:         alignItems: 'center'
122:     },
123:     screenTitle: {
124:         fontSize: 32,
125:         fontWeight: '700',
126:         color: Colors.black,
127:         paddingTop: 20
128:     },

```

```
129:   button: {
130:     width: '100%',
131:     justifyContent: 'center',
132:     alignItems: 'center',
133:     marginTop: 8,
134:     backgroundColor: Colors.orange,
135:     padding: 10,
136:     borderRadius: 8
137:   },
138:   buttonText: {
139:     fontSize: 20,
140:     color: Colors.white,
141:     fontWeight: '700'
142:   },
143:   borderlessButtonContainer: {
144:     marginTop: 16,
145:     alignItems: 'center',
146:     justifyContent: 'center'
147:   }
148: });
149:
```



```
121:         fontWeight: '700'
122:     },
123:     stat_text: {
124:         fontWeight: "500",
125:         fontSize: 20
126:     },
127:     item:{
128:         width: width,
129:         height: 100,
130:         paddingHorizontal: width * 0.05,
131:         borderBottomWidth: 4,
132:         borderColor: '#f0f0f0',
133:         justifyContent: 'center'
134:     },
135:     itemTitle:{
136:         fontSize: 20,
137:         fontWeight: '600'
138:     },
139:     itemTime: {
140:         fontSize: 16,
141:         fontWeight: '400'
142:     }
143: })
```



```
1: import React from 'react'
2: import { StyleSheet, Text, Dimensions, ScrollView, Pressable } from 'react-native'
3: import { View } from '../components'
4: import { Colors } from '../config';
5: import { Ionicons } from '@expo/vector-icons';
6: const {width, height} = Dimensions.get('window');
7:
8: const PatientMeetItem = ({navigation, route}) => {
9:
10:   const { data } = route.params;
11:
12:   return (
13:     <ScrollView>
14:       <View isSafe style={styles.container}>
15:         <View style={styles.viewHeader}>
16:           <Pressable onPress={() => navigation.goBack()}>
17:             <Ionicons name='arrow-back-circle' size={30} color={Colors
.blue} />
18:           </Pressable>
19:           <Text style={styles.textHeader}>{data.title}</Text>
20:         </View>
21:         <View style={styles.content}>
22:           <Text style={[styles.textStyle, {fontWeight: '300'}}>{(new Intl
.DateTimeFormat("th-TH",{ dateStyle: 'full', timeStyle: 'short' }).format((new Date(data.tim
e.seconds * 1000))).toString())}</Text>
23:           <Text style={styles.textStyle}>à¹\201à,\236à,\227à,çà¹\214: {dat
a.doctor_name}</Text>
24:           <Text style={styles.textStyle}>à,£à,²à,çà,¥à,°à¹\200à,-à,µà,çà,
\224: {data.description}</Text>
25:           <Text style={[styles.textStyle, {color : (data.note.length > 0)
? "black" : "red"}]}>note à,\202à,-à,\207à¹\201à,\236à,\227à,çà¹\214: {(data.note.length > 0)
? data.note : "à,£à,-à¹\201à,\236à,\227à,çà¹\214à,\233à,£à,°à¹\200à,¡à,´à,\223à,«à,¥à,±à,
\207à,\231à,±à,\224à,\236à,\232"}</Text>
26:         </View>
27:       </View>
28:     </ScrollView>
29:   )
30: }
31:
32: export default PatientMeetItem
33:
34: const styles = StyleSheet.create({
35:   container: {
36:     flex: 1,
37:     alignItems: 'center',
38:     backgroundColor: 'white',
39:     minHeight: height
40:     // justifyContent: 'center'
41:   },
42:   content: {
43:     width: width*0.8,
44:     alignItems: 'center'
45:   },
46:   button: {
47:     justifyContent: 'center',
48:     width: width*0.9,
49:     height: 50,
50:     borderRadius: 5,
51:     marginVertical: 10
52:   },
53:   buttonOpen: {
54:     backgroundColor: Colors.orange,
55:   },
56:   buttonSummit: {
57:     backgroundColor: '#1597e5',
58:   },
```

```
59:   textStyle: {
60:     fontWeight: "bold",
61:     fontSize: 20,
62:     textAlign: "left",
63:     alignSelf: 'flex-start'
64:   },
65:   textHeader: {
66:     fontWeight: 'bold',
67:     fontSize: 30,
68:     marginLeft: 20,
69:     marginBottom: 10
70:   },
71:   viewHeader: {
72:     width: width*0.9,
73:     flexDirection: 'row',
74:     justifyContent: 'flex-start',
75:     alignItems: 'center',
76:     alignSelf: 'center'
77:   },
78: })
79:
80:
```



```

1: import React, { useState, useEffect } from 'react'
2: import { StyleSheet, Text, Dimensions, Pressable, Platform, TextInput, Image, Modal
} from 'react-native'
3: import { View } from '../../components'
4: import { KeyboardAwareScrollView } from 'react-native-keyboard-aware-scroll-view';
5: import DateTimePicker from '@react-native-community/datetimepicker'
6: import * as ImagePicker from 'expo-image-picker';
7: import uuid from 'react-native-uuid';
8: import { Ionicons } from '@expo/vector-icons';
9: import { doc, setDoc } from "firebase/firestore";
10: import { ref, uploadBytesResumable, getDownloadURL } from "firebase/storage";
11: import { Colors, db, storage, auth } from '../../config';
12: const {width, height} = Dimensions.get('window');
13:
14: export default function CreateList({navigation}) {
15:   const [title, setTitle] = useState('')
16:   const [description, setDescription] = useState('')
17:   const [imageUrl, setImage] = useState('')
18:   const [date, setDate] = useState(new Date())
19:   const [show, setShow] = useState(false)
20:   const [enable, setEnable] = useState(true)
21:   const [percentUp, setPercent] = useState(0)
22:
23:   useEffect(() => {
24:     (async () => {
25:       if (Platform.OS !== 'web') {
26:         const { status } = await ImagePicker.requestMediaLibraryPermissionsAsync
27:         if (status !== 'granted') {
28:           Alert.alert(
29:             "à,\201à,\2à,\200à,\202à,\211à,\2à,\226à,\207à,\2à,\233à,\236",
30:             "Sorry, we need camera roll permissions to make pick the photo",
31:             [
32:               { text: "OK", onPress: () => console.log("OK Pressed") }
33:             ]
34:           )
35:         }
36:       })();
37:     }, []);
38:
39:     const pickImage = async () => {
40:       let result = await ImagePicker.launchImageLibraryAsync({
41:         mediaTypes: ImagePicker.MediaTypeOptions.Images,
42:         quality: 1,
43:       });
44:       //console.log(result);
45:       if (!result.cancelled) {
46:         setImage(result.uri);
47:       }
48:     };
49:
50:     const onChangeDateTime = (event, selectedDate) => {
51:       const currentDate = selectedDate || date;
52:       console.log(currentDate);
53:       setShow(Platform.OS === 'ios')
54:       setDate(currentDate);
55:     };
56:
57:     const uploadImageAsync = async (uri, path, callBack) => {
58:       if (uri) {
59:         console.log('image ok', uri);
60:         const imageRef = ref(storage, path);
61:         await fetch(uri)
62:           .then(response => response.blob())
63:           .then(blob => {

```

```

64:         const uploadTask = uploadBytesResumable(imageRef, blob);
65:         uploadTask.on('state_changed', (snapshot) => {
66:           const progress = (snapshot.bytesTransferred / snapshot.totalByte
s) * 100;
67:           console.log('Upload is ' + progress + '% done');
68:           setPercent(progress.toFixed(1))
69:         },
70:         (error) => {
71:         },
72:         () => {
73:           getDownloadURL(uploadTask.snapshot.ref).then((downloadURL) => {
74:             //console.log('File available at', downloadURL);
75:             callBack(downloadURL);
76:           });
77:         })
78:       });
79:     } else {
80:       callBack('');
81:     }
82:   }
83:
84:   const uploadItemList = () => {
85:     setEnable(false)
86:     if (!title || !description) {
87:       alert('à,\201à,\2à,\223à,\2à,\232à,\2à,\2à,\2à,\200à,\224')
88:       setEnable(true)
89:       return
90:     }
91:     const id = uuid.v4()
92:     const listRef = doc(db, "users", auth.currentUser.uid, "List", id)
93:     const payload = {
94:       day: date,
95:       day_string: (new Intl.DateTimeFormat("th-TH", { dateStyle: 'full', timeSt
yle: 'short' }).format(date)).toString(),
96:       title: title,
97:       description: description,
98:       id: id,
99:       imageUrl: '',
100:       isDone: false
101:     }
102:     setDoc(listRef, payload, {merge: true})
103:     uploadImageAsync(imageUrl, 'list-image/' + id, (result) => {
104:       if (result) {
105:         setDoc(listRef, {imageUrl: result}, {merge: true})
106:       }
107:       navigation.goBack()
108:     })
109:   }
110:
111:   return (
112:     <View isSafe style={{backgroundColor: 'white', height: height}}>
113:       <KeyboardAwareScrollView enableOnAndroid={true}>
114:         <View style={styles.container}>
115:           <Modal
116:             transparent={true}
117:             visible={percentUp > 0 && percentUp < 100 ? true : false}
118:           >
119:             <View style={{flex: 1, justifyContent: 'center', alignItems:
'center'}}>
120:               <View style={styles.modalView}>
121:                 <Text style={[styles.textStyle, {color: 'black'}]}>à
,\236à,\202à,\2à,\224à,\201à,\2à,\211à,{percentUp}%</Text>
122:               </View>
123:             </View>
124:           </Modal>

```

```
125:         <View style={styles.viewHeader}>
126:           <Pressable onPress={() => navigation.goBack()}>
127:             <Ionicons name='arrow-back-circle' size={30} color={Co
lors.blue} />
128:           </Pressable>
129:           <Text style={styles.textHeader}>à, ªà, ¢à¹\211à, ºà, \207à, ¢à, ºà
, ¢à, \201à, ºà, ¢à¹\203à, ¢à, ¢à¹\210</Text>
130:         </View>
131:         <TextInput
132:           placeholder='à, ¢à, ¢à, ¢à, \202à¹\211à, -'
133:           style={styles.titleInput}
134:           value={title}
135:           onChangeText={setTitle}
136:         />
137:         <TextInput
138:           placeholder='à, ¢à, ºà, ¢à, ¢à, ¢à, ¢à¹\200à, -à, ¢à, ¢à, \224'
139:           multiline={true}
140:           style={styles.multiInput}
141:           value={description}
142:           onChangeText={setDescription}
143:         />
144:         <Text style={styles.textStyle}>à¹\200à, \236à, ¢à¹\210à, ¢à, ¢à, ºà,
\236</Text>
145:         {imageUrl ?
146:           <Image
147:             style={{
148:               height: width*0.7,
149:               width: width*0.7,
150:               borderRadius: 5,
151:               margin: 15
152:             }}
153:             source={{uri: imageUrl}}
154:           /> :
155:           null
156:         }
157:         <Pressable
158:           style={[styles.button, styles.buttonOpen]}
159:           onPress={pickImage}
160:         >
161:           <Text style={styles.textStyle}>à¹\200à, ¢à, ¢à, -à, \201à, ¢à, ºà,
\236</Text>
162:         </Pressable>
163:         {show &&
164:           <View style={{width: '100%'}}>
165:             <DateTimePicker
166:               style={styles.picker}
167:               testID="datePicker"
168:               value={date}
169:               mode='date'
170:               is24Hour={true}
171:               display="default"
172:               onChange={onChangeDateTime}
173:             />
174:             <DateTimePicker
175:               style={styles.picker}
176:               testID="datePicker"
177:               value={date}
178:               mode='time'
179:               is24Hour={true}
180:               display="default"
181:               onChange={onChangeDateTime}
182:             />
183:           </View>
184:         }
185:         <Pressable
186:           disabled={!enable}
```

```
187:           style={[styles.button, styles.buttonOpen]}
188:           onPress={() => setShow(true)}
189:         >
190:           <Text style={styles.textStyle}>à¹\200à, ¢à, ¢à, -à, \201à, ¢à, ºà,
\231à, \227à, ¢à¹\210à¹\201à, ¢à, ¢à¹\200à, ¢à, ¢à, ºà, ºà</Text>
191:         </Pressable>
192:         <Pressable
193:           disabled={!enable}
194:           style={[styles.button, styles.buttonSummit, {marginBottom: 1
00, opacity: enable ? 1 : 0.5}]}
195:           onPress={uploadItemList}
196:         >
197:           <Text style={styles.textStyle}>à, \225à, \201à, ¢à, \207</Text>
198:         </Pressable>
199:       </View>
200:     </KeyboardAwareScrollView>
201:   </View>
202: )
203: }
204:
205: const styles = StyleSheet.create({
206:   container: {
207:     flex: 1,
208:     alignItems: 'center',
209:     backgroundColor: 'white',
210:     // justifyContent: 'center'
211:   },
212:   picker: {
213:     width: 90,
214:     marginHorizontal: (width - 90)/2,
215:     marginVertical: 5
216:   },
217:   button: {
218:     justifyContent: 'center',
219:     width: width*0.9,
220:     height: 50,
221:     borderRadius: 5,
222:     marginVertical: 10
223:   },
224:   buttonOpen: {
225:     backgroundColor: Colors.orange,
226:   },
227:   buttonSummit: {
228:     backgroundColor: '#1597e5',
229:   },
230:   textStyle: {
231:     color: "white",
232:     fontWeight: "bold",
233:     textAlign: "center"
234:   },
235:   textHeader: {
236:     fontWeight: 'bold',
237:     fontSize: 30,
238:     marginLeft: 20,
239:     marginBottom: 10
240:   },
241:   viewHeader: {
242:     width: width*0.9,
243:     flexDirection: 'row',
244:     justifyContent: 'flex-start',
245:     alignItems: 'center',
246:     alignSelf: 'center'
247:   },
248:   titleInput: {
249:     height: 50,
250:     width: width*0.9,
```

```
251:         margin: 12,
252:         padding: 10,
253:         borderRadius: 5,
254:         fontSize: 20,
255:         backgroundColor: "#f7f7f7",
256:     },
257:     multiInput: {
258:         minHeight: 100,
259:         width: width*0.9,
260:         margin: 12,
261:         padding: 10,
262:         borderRadius: 5,
263:         fontSize: 20,
264:         backgroundColor: "#f7f7f7",
265:     },
266:     modalView: {
267:         margin: 20,
268:         backgroundColor: "white",
269:         borderRadius: 5,
270:         padding: 50,
271:         alignItems: "center",
272:         shadowColor: "#000",
273:         shadowOffset: {
274:             width: 0,
275:             height: 2
276:         },
277:         shadowOpacity: 0.25,
278:         shadowRadius: 4,
279:         elevation: 5
280:     },
281: })
```



```

1: import React from 'react'
2: import { StyleSheet, Text, Image, Dimensions, ScrollView, Pressable } from 'react-native'
3: import { View, Button } from '../../components'
4: import { Colors, db, auth } from '../../config';
5: import { setDoc, doc } from '@firebase/firestore';
6: import { Ionicons } from '@expo/vector-icons';
7: const {width, height} = Dimensions.get('window');
8:
9: const ItemList = ({navigation, route}) => {
10:
11:   const onDone = () => {
12:     const { id } = route.params.data
13:     const listRef = doc(db, "users", auth.currentUser.uid , "List", id)
14:     setDoc(listRef, {isDone: true}, {merge: true})
15:     navigation.goBack()
16:   }
17:
18:   return (
19:     <ScrollView>
20:       <View isSafe style={styles.container}>
21:         <View style={styles.viewHeader}>
22:           <Pressable onPress={() => navigation.goBack()}>
23:             <Ionicons name='arrow-back-circle' size={30} color={Colors
.blue} />
24:           </Pressable>
25:           <Text style={styles.textHeader}>{route.params.data.title}</Text>
26:         </View>
27:         <View style={styles.content}>
28:           <Text style={[styles.textStyle, {fontWeight: '300'}]}>{route.par
ams.data.day_string}</Text>
29:           <Text style={styles.textStyle}>à,è,²à,çà,à,°à¹\200à,-à,ùà,çà,
\224: {route.params.data.description}</Text>
30:           {route.params.data.imageUrl != "" ?
31:             <Image
32:               style={{
33:                 height: width*0.7,
34:                 width: width*0.7,
35:                 borderRadius: 5,
36:                 marginVertical: 12,
37:                 alignSelf: 'flex-start'
38:               }}
39:               source={{uri: route.params.data.imageUrl}}
40:             /> :
41:             <View
42:               style={{
43:                 height: width*0.7,
44:                 width: width*0.7,
45:                 borderRadius: 5,
46:                 marginVertical: 12,
47:                 alignSelf: 'flex-start',
48:                 backgroundColor: '#a0a0a0',
49:                 alignItems: 'center',
50:                 justifyContent: 'center'
51:               }}
52:             >
53:             <Text style={[styles.textStyle, {alignSelf: 'center', co
lor: 'white'}}>à¹\204à,à¹\210à,à,ùà, à,²à,\236</Text>
54:           </View>
55:         </View>
56:         {route.params.data.isDone === false ?
57:           <Button
58:             style={styles.button, styles.buttonOpen}
59:             onPress={onDone}
60:           >
61:           <Text style={[styles.textStyle, {alignSelf: 'center', co

```

```

lor: 'white'}}>à,\227à,³à¹\201à,à¹\211à,</Text>
62:           </Button> :
63:           <Text style={styles.textHeader}>à,\227à,³à¹\200à,*à,èà¹\207à
,\210à¹\201à,à¹\211à,</Text>
64:         }
65:       </View>
66:     </View>
67:   </ScrollView>
68: )
69: }
70:
71: export default ItemList
72:
73: const styles = StyleSheet.create({
74:   container: {
75:     flex: 1,
76:     alignItems: 'center',
77:     backgroundColor: 'white',
78:     height: height
79:     // justifyContent: 'center'
80:   },
81:   content: {
82:     width: width*0.8,
83:     alignItems: 'center'
84:   },
85:   button: {
86:     justifyContent: 'center',
87:     width: width*0.9,
88:     height: 50,
89:     borderRadius: 5,
90:     marginVertical: 10
91:   },
92:   buttonOpen: {
93:     backgroundColor: Colors.orange,
94:   },
95:   buttonSummit: {
96:     backgroundColor: '#1597e5',
97:   },
98:   textStyle: {
99:     fontWeight: "bold",
100:    fontSize: 20,
101:    textAlign: "left",
102:    alignSelf: 'flex-start'
103:  },
104:  textHeader: {
105:    fontWeight: 'bold',
106:    fontSize: 30,
107:    marginLeft: 20,
108:    marginBottom: 10
109:  },
110:  viewHeader: {
111:    width: width*0.9,
112:    flexDirection: 'row',
113:    justifyContent: 'flex-start',
114:    alignItems: 'center',
115:    alignSelf: 'center'
116:  },
117: })
118:

```



```

61:         return notDoneLists.map(item => {
62:             return <TouchableOpacity>
63:                 key={item.id}
64:                 style={styles.item}
65:                 onPress={() => navigation.navigate("Item", { data: item })}
66:             >
67:                 <Text style={styles.itemTitle}>{item.title}</Text>
68:                 <Text style={styles.itemTime}>{(new Date(item.day.seconds
s * 1000)).toString()}</Text> */)
69:                 <Text style={styles.itemTime}>{item.day_string}</Text>
70:             </TouchableOpacity>
71:         })
72:     }
73:
74:     const renderItemAll = () => {
75:         return sortLists.map(item => {
76:             return <TouchableOpacity>
77:                 key={item.id}
78:                 style={styles.item}
79:                 onPress={() => navigation.navigate("Item", { data: item })}
80:             >
81:                 <Text style={styles.itemTitle}>{item.title}</Text>
82:                 <Text style={styles.itemTime}>{item.day_string}</Text>
83:                 <Text style={{[styles.itemTitle, {color: (item.isDone) ? "#00
bf0b" : ((curDay - (item.day.seconds * 1000) > 0) ? "red" : "#fbbf00")}]}}>{(item.isDone) ? "
à,\227à,\3à\201à,\8à\211à,\$": ((curDay - (item.day.seconds * 1000)) >= 0) ? "à,\8à,\3à,\227
à,\3": "à,\8à,\2à,\207à\204à,\jà\210à\204à,\224à\211à,\227à,\3"}</Text>
84:                 <Text style={styles.itemTime}>{item.day_string}</Text>
85:             </TouchableOpacity>
86:         })
87:     }
88:
89:     const SwitchRender = () => {
90:         switch (buttonsSelect[indexSelect]) {
91:             case 'à,\227à,\3à\200à,\8à,\8à\207à,\210à\201à,\8à\211à,\$:':
92:                 return <RenderItemDone/>
93:             case 'à,\8à,\2à,\207à\204à,\jà\210à\204à,\224à\211à,\227à,\3':
94:                 return <RenderItemNotDone/>
95:             case 'à,\8à,\8à,\jà,\227à,\2à\211à,\207à,\8à,\jà,\224à,\225à,\2à\211à,\207à\
201à,\225à\210à,\225à\211à,\231à,\210à,\231à,\226à,\8à,\207à,\8à,\8à,\2à,\207 \8à,\2à,\231à
\200à,\233à\207à,\231à,\225à\211à,\231à\204à,\233':
96:                 return <RenderItemAll/>
97:             default:
98:                 break;
99:         }
100:     }
101:
102:     return (
103:         <View isSafe style={styles.container}>
104:             <Text style={styles.header}>à,\8à,\8à\210à,\207à,\227à,\8à\210à,\225à\
211à,\8à,\207à,\227à,\3</Text>
105:             <TouchableOpacity>
106:                 style={styles.btn}
107:                 onPress={() => navigation.navigate("Create_new")}
108:             >
109:                 <Text style={{color: 'white'}}>à,\8à,\8à\211à,\2à,\207à,\8à,\2à,\8à,\201à
,\2à,\8à\203à,\8à,\jà\210</Text>
110:             </TouchableOpacity>
111:             <ButtonGroup>
112:                 onPress={onSelectIndex}
113:                 selectedIndex={indexSelect}
114:                 buttons={buttonsSelect}
115:                 containerStyle={{height: 70, marginBottom: 12, padding: 5}}
116:             />
117:             <!!loading ? (<ScrollView>
118:                 <View style={{height: 4, width: width, backgroundColor: '#f0f0f0'}}>

```

```
</View>
119:         <SwichRender/>
120:     </ScrollView>) : <LoadingIndicator/>}
121: </View>
122: )
123: }
124:
125: const styles = StyleSheet.create({
126:   container: {
127:     flex: 1,
128:     alignItems: 'center',
129:     backgroundColor: 'white',
130:     // justifyContent: 'center'
131:   },
132:   header: {
133:     fontWeight: 'bold',
134:     fontSize: 30,
135:     alignSelf: 'flex-start',
136:     marginLeft: 20,
137:     marginBottom: 10,
138:   },
139:   btn: {
140:     width: width*0.9,
141:     height: 50,
142:     borderRadius: 5,
143:     backgroundColor: Colors.orange,
144:     alignItems: 'center',
145:     justifyContent: 'center',
146:     marginVertical: 12
147:   },
148:   item:{
149:     width: width,
150:     height: 70,
151:     paddingHorizontal: width * 0.05,
152:     borderBottomWidth: 4,
153:     borderColor: '#f0f0f0',
154:     justifyContent: 'center'
155:   },
156:   itemTitle:{
157:     fontSize: 20,
158:     fontWeight: '600'
159:   },
160:   itemTime: {
161:     fontSize: 16,
162:     fontWeight: '400'
163:   }
164: })
```

```
1: import { StyleSheet, Text, View, Dimensions, TextInput } from 'react-native'
2: import React, { useState } from 'react'
3:
4: const {width, height} = Dimensions.get('window');
5: const COLORS = {primary: '#282534', white: '#fff'};
6:
7: export const AnsC1 = ({item, setChoice, choice}) => {
8:   const [text1, setText1] = useState('')
9:   const [text2, setText2] = useState('')
10:  const [text3, setText3] = useState('')
11:
12:  const changeText1 = (e) => {
13:    setText1(e);
14:    var newChoice = choice;
15:    newChoice[item.id].anstext1 = e;
16:    setChoice(newChoice)
17:    // console.log("#####");
18:    // console.log(e);
19:    // console.log(text1) missing last 1 text
20:    // console.log(choice[item.id].anstext1);
21:  }
22:
23:  const changeText2 = (e) => {
24:    setText2(e);
25:    var newChoice = choice;
26:    newChoice[item.id].anstext2 = e;
27:    setChoice(newChoice)
28:  }
29:
30:  const changeText3 = (e) => {
31:    setText3(e);
32:    var newChoice = choice;
33:    newChoice[item.id].anstext3 = e;
34:    setChoice(newChoice)
35:  }
36:
37:  return (
38:    <View style={{flex: 1, width: width, alignItems: 'center', justifyContent: 'center'}}>
39:      <View style={{alignItems: 'center', width: width * 0.9}}>
40:        <Text style={styles.title}>{item.title}</Text>
41:        <Text style={styles.title}>{item.subtitle}</Text>
42:        <TextInput style={styles.input} value={text1} onChangeText={changeText1}></TextInput>
43:        <TextInput style={styles.input} value={text2} onChangeText={changeText2}></TextInput>
44:        <TextInput style={styles.input} value={text3} onChangeText={changeText3}></TextInput>
45:      </View>
46:    </View>
47:  )
48: }
49:
50: const styles = StyleSheet.create({
51:   title: {
52:     color: COLORS.white,
53:     fontSize: 22,
54:     fontWeight: 'bold',
55:     marginTop: 20,
56:     textAlign: 'center',
57:   },
58:   input: {
59:     height: 50,
60:     width: width * 0.9,
61:     borderRadius: 5,
62:     backgroundColor: '#fff',
63:     padding: 10,
64:     margin: 10,
65:   },
66: });
```



```

1: import React, { useState, useEffect } from 'react'
2: import { StyleSheet, Text, View, Image, Dimensions, TouchableOpacity } from 'react-native'
3: import * as ImagePicker from 'expo-image-picker';
4:
5: const {width, height} = Dimensions.get('window');
6: const COLORS = {primary: '#282534', white: '#fff'};
7:
8: export function DrawBlock({item, setChoice, choice}) {
9:
10:   const [image, setImage] = useState();
11:
12:   useEffect(() => {
13:     (async () => {
14:       if (Platform.OS !== 'web') {
15:         const { status } = await ImagePicker.requestCameraPermissionsAsync()
16:
17:         if (status !== 'granted') {
18:           alert('Sorry, we need camera roll permissions to make this work!');
19:         }
20:       })();
21:     }, [])
22:
23:     const pickImage = async () => {
24:       let result = await ImagePicker.launchCameraAsync({
25:         mediaTypes: ImagePicker.MediaTypeOptions.Images,
26:         quality: 1,
27:       });
28:
29:       if (!result.cancelled) {
30:         setImage(result.uri);
31:         var newChoice = choice
32:         newChoice[item.id].url = result.uri
33:         setChoice(newChoice);
34:       }
35:     };
36:
37:
38:     return (
39:       <View style={{flex: 1, width: width, alignItems: 'center', justifyContent: 'center'}}>
40:         <View style={{alignItems: 'center', width: width * 0.9}}>
41:           <Text style={styles.title}>{item.title}</Text>
42:           <Image
43:             style={{
44:               height: width*0.45,
45:               width: width*0.45,
46:               borderRadius: 5,
47:               margin: 15
48:             }}
49:             source={item.boxurl}
50:           />
51:           {image ?
52:             <Image
53:               style={{
54:                 height: width*0.7,
55:                 width: width*0.7,
56:                 borderRadius: 5,
57:                 margin: 15
58:               }}
59:               source={{uri: image}}
60:             /> :
61:             null
62:           }
63:         <TouchableOpacity

```

```

64:         style={{
65:           height: 60,
66:           width: width * 0.9,
67:           backgroundColor: "#fff",
68:           borderRadius: 5,
69:           alignItems: 'center',
70:           justifyContent: 'center',
71:           marginVertical: 5
72:         }}
73:         onPress={pickImage}
74:       >
75:       <Text>à¹\200à,¥à,•à,-à,\201à, à,²à,\236à,$à,²à,\224</Text>
76:     </TouchableOpacity>
77:   </View>
78: </View>
79: )
80: }
81:
82: const styles = StyleSheet.create({
83:   title: {
84:     color: COLORS.white,
85:     fontSize: 22,
86:     fontWeight: 'bold',
87:     marginTop: 20,
88:     textAlign: 'center',
89:   },
90:   input: {
91:     height: 50,
92:     width: width * 0.9,
93:     borderRadius: 5,
94:     backgroundColor: '#fff',
95:     padding: 10,
96:     margin: 10,
97:   },
98: });

```



```
1: import { StyleSheet, Text, View, Dimensions, TextInput, Image } from 'react-native'
2: import React, { useState } from 'react'
3:
4: const {width, height} = Dimensions.get('window');
5: const COLORS = {primary: '#282534', white: '#fff'};
6:
7: export const Cimage = ({item, setChoice, choice}) => {
8:   const [text, setText] = useState('')
9:
10:   const changeText = (e) => {
11:     setText(e);
12:     // console.log(e);
13:     var newChoice = choice;
14:     newChoice[item.id].user_answer = e;
15:     setChoice(newChoice)
16:     // console.log('choice a', choice[item.id]);
17:   }
18:
19:   return (
20:     <View style={{flex: 1, width: width, alignItems: 'center', justifyContent: 'center'}}>
21:       <View style={{alignItems: 'center', width: width * 0.9}}>
22:         <Text style={styles.title}>{item.title}</Text>
23:         <Image style={{width: 200, height: 200}} source={item.url}/>
24:         <TextInput style={styles.input} value={text} onChangeText={changeText}/>
25:       </View>
26:     </View>
27:   )
28: }
29:
30: const styles = StyleSheet.create({
31:   title: {
32:     color: COLORS.white,
33:     fontSize: 22,
34:     fontWeight: 'bold',
35:     marginTop: 20,
36:     textAlign: 'center',
37:   },
38:   input: {
39:     height: 50,
40:     width: width * 0.9,
41:     borderRadius: 5,
42:     backgroundColor: '#fff',
43:     padding: 10,
44:     margin: 10,
45:   },
46: });
```



```

1: import { StyleSheet, Text, View, Dimensions } from 'react-native'
2: import React, { useEffect, useState } from 'react'
3:
4: const {width, height} = Dimensions.get('window');
5: const COLORS = {primary: '#282534', white: '#fff'};
6:
7: function found(arrayX, x) {
8:   return (arrayX.find(e => e === x) !== undefined) ? true : false
9: }
10:
11: function random(alltext){
12:   var n = [];
13:   var pushed_index = [];
14:   for (let i = 0; i < 3; i++) {
15:     var index = Math.floor(Math.random() * alltext.length)
16:     if (!found(pushed_index, index)) {
17:       n.push(alltext[index])
18:       pushed_index.push(index)
19:     } else {
20:       i = i - 1;
21:     }
22:   }
23:   return n
24: }
25:
26: export const C1 = ({item, setChoice, choice}) => {
27:
28:   const [test_text, setText] = useState([])
29:
30:   useEffect(() => {
31:     let newChoice = choice
32:     var testtext = random(newChoice[item.id].alltext)
33:     newChoice[item.id].text1 = testtext[0]
34:     newChoice[item.id].text2 = testtext[1]
35:     newChoice[item.id].text3 = testtext[2]
36:     setChoice(newChoice)
37:     setText(testtext)
38:   }, [])
39:
40:   return (
41:     <View style={{flex: 1, width: width, alignItems: 'center', justifyContent: '
center'}}>
42:       <View style={{alignItems: 'center', width: width * 0.9}}>
43:         <Text style={styles.title}>{item.title}</Text>
44:         <Text style={styles.title}>{item.subtitle}</Text>
45:         <Text style={styles.title}>{test_text[0]} {test_text[1]} {test_text[
2]}</Text>
46:       </View>
47:     </View>
48:   )
49: }
50:
51: const styles = StyleSheet.create({
52:   subtitle: {
53:     color: COLORS.white,
54:     fontSize: 18,
55:     marginTop: 10,
56:     maxWidth: '70%',
57:     textAlign: 'center',
58:     lineHeight: 23,
59:   },
60:   minisubtitle: {
61:     color: COLORS.white,
62:     fontSize: 13,
63:     marginTop: 10,
64:     maxWidth: '70%',

```

```

65:     textAlign: 'center',
66:     lineHeight: 23,
67:   },
68:   title: {
69:     color: COLORS.white,
70:     fontSize: 22,
71:     fontWeight: 'bold',
72:     marginTop: 20,
73:     textAlign: 'center',
74:   },
75:   indicator: {
76:     height: 2.5,
77:     width: 10,
78:     backgroundColor: '#C36A2D',
79:     marginHorizontal: 3,
80:     borderRadius: 2,
81:   },
82:   btn: {
83:     flex: 1,
84:     height: 50,
85:     borderRadius: 5,
86:     backgroundColor: '#fff',
87:     justifyContent: 'center',
88:     alignItems: 'center',
89:   },
90: });

```



```
1: import { StyleSheet, Text, TouchableOpacity, View, Dimensions } from 'react-native'
2: import React from 'react'
3: const {width, height} = Dimensions.get('window');
4: const COLORS = {primary: '#282534', white: '#fff'};
5:
6: export const Intro = ({item, goNext}) => {
7:   return (
8:     <View style={{flex: 1, width: width, alignItems: 'center', justifyContent: 'center'}}>
9:       <View style={{alignItems: 'center', width: width * 0.9}}>
10:        <Text style={styles.title}>{item.title}</Text>
11:        <Text style={styles.subtitle}>{item.subtitle}</Text>
12:        <Text style={styles.minisubtitle}>{item.description}</Text>
13:        <TouchableOpacity
14:          style={{
15:            height: 60,
16:            width: width * 0.9,
17:            backgroundColor: "#fff",
18:            borderRadius: 5,
19:            alignItems: 'center',
20:            justifyContent: 'center',
21:            marginVertical: 5
22:          }}
23:          onPress={goNext}
24:        >
25:          <Text>à¹\200à,àà,à¹\210à,à,à\201à,àà,\231à¹\200à,àà,à</Text>
26:        </TouchableOpacity>
27:      </View>
28:    </View>
29:  )
30: }
31:
32: const styles = StyleSheet.create({
33:   subtitle: {
34:     color: COLORS.white,
35:     fontSize: 18,
36:     marginTop: 10,
37:     maxWidth: '70%',
38:     textAlign: 'center',
39:     lineHeight: 23,
40:   },
41:   minisubtitle: {
42:     color: COLORS.white,
43:     fontSize: 13,
44:     marginTop: 10,
45:     maxWidth: '70%',
46:     textAlign: 'center',
47:     lineHeight: 23,
48:   },
49:   title: {
50:     color: COLORS.white,
51:     fontSize: 22,
52:     fontWeight: 'bold',
53:     marginTop: 20,
54:     textAlign: 'center',
55:   },
56:   indicator: {
57:     height: 2.5,
58:     width: 10,
59:     backgroundColor: '#C36A2D',
60:     marginHorizontal: 3,
61:     borderRadius: 2,
62:   },
63:   btn: {
64:     flex: 1,
65:     height: 50,
66:     borderRadius: 5,
67:     backgroundColor: '#fff',
68:     justifyContent: 'center',
69:     alignItems: 'center',
70:   },
71: });
```



```

1: import React, { useState, useEffect } from 'react'
2: import { StyleSheet, Text, View, Image, Dimensions, TouchableOpacity } from 'react-native'
3: import * as ImagePicker from 'expo-image-picker';
4:
5: const {width, height} = Dimensions.get('window');
6: const COLORS = {primary: '#282534', white: '#fff'};
7:
8: export function DrawClock({item, setChoice, choice}) {
9:
10:   const [image, setImage] = useState();
11:
12:   useEffect(() => {
13:     (async () => {
14:       if (Platform.OS !== 'web') {
15:         const { status } = await ImagePicker.requestCameraPermissionsAsync()
16:
17:         if (status !== 'granted') {
18:           alert('Sorry, we need camera roll permissions to make this work!');
19:         }
20:       })();
21:     }, [])
22:
23:     const pickImage = async () => {
24:       let result = await ImagePicker.launchCameraAsync({
25:         mediaTypes: ImagePicker.MediaTypeOptions.Images,
26:         quality: 1,
27:       });
28:
29:       if (!result.cancelled) {
30:         setImage(result.uri);
31:         var newChoice = choice
32:         newChoice[item.id].url = result.uri
33:         setChoice(newChoice);
34:       }
35:     };
36:
37:
38:     return (
39:       <View style={{flex: 1, width: width, alignItems: 'center', justifyContent: 'center'}}>
40:         <View style={{alignItems: 'center', width: width * 0.9}}>
41:           <Text style={styles.title}>{item.title}</Text>
42:           {image ?
43:             <Image
44:               style={{
45:                 height: width*0.7,
46:                 width: width*0.7,
47:                 borderRadius: 5,
48:                 margin: 15
49:               }}
50:               source={{uri: image}}
51:             /> :
52:             null
53:           }
54:           <TouchableOpacity
55:             style={{
56:               height: 60,
57:               width: width * 0.9,
58:               backgroundColor: "black",
59:               borderRadius: 5,
60:               alignItems: 'center',
61:               justifyContent: 'center',
62:               marginVertical: 5
63:             }}

```

```

64:               onPress={pickImage}
65:             >
66:               <Text>à'\200à,à,•à,-à,\201à, à,²à,\236à,$à,²à,\224</Text>
67:             </TouchableOpacity>
68:           </View>
69:         </View>
70:       )
71:     }
72:
73:     const styles = StyleSheet.create({
74:       title: {
75:         color: COLORS.white,
76:         fontSize: 22,
77:         fontWeight: 'bold',
78:         marginTop: 20,
79:         textAlign: 'center',
80:       },
81:       input: {
82:         height: 50,
83:         width: width * 0.9,
84:         borderRadius: 5,
85:         backgroundColor: '#fff',
86:         padding: 10,
87:         margin: 10,
88:       },
89:     });

```



```

232a1\211a,2a,\207',
61: subtitle: 'à,à,à,\207à,\210à,3a1\204à,\224a1\211a1\204à,«à, à,\210à,3a1
\204à,\224a1\211a,«à,à,à,-à1\200à,\233à,à1\210à,2?',
62: anstext1: '',
63: anstext2: '',
64: anstext3: '',
65: },
66: {
67: id: 6,
68: title: 'à,à,à,\224à,\231à,2à,-à,à,\201à,2',
69: url: '',
70: },
71: {
72: id: 7,
73: title: 'à,à,à,\224à,à,1à,\233à,\201à,à1\210à,-à,\207à,\225à,2à,à,à,à,2à,
\236',
74: boxurl: require('../../assets/Block.png'),
75: url: '',
76: },
77: {
78: id: 8,
79: title: 'à1\200à,à,à,à1\207à,\210à,à,à,à1\211à,\231'
80: }
81: ]
82:
83: const Slide = ({item, goNext, choice, setChoice}) => {
84:   if (item.id === 0) {
85:     return <Intro item={item} goNext={goNext}/>
86:   } else if (item.id === 1) {
87:     return <C1 item={item} choice={choice} setChoice={setChoice}/>
88:   } else if (item.id === 2) {
89:     return <Cimage item={item} choice={choice} setChoice={setChoice}/>
90:   } else if (item.id === 3){
91:     return <Cimage item={item} choice={choice} setChoice={setChoice}/>
92:   } else if (item.id === 4){
93:     return <Cimage item={item} choice={choice} setChoice={setChoice}/>
94:   } else if (item.id === 5) {
95:     return <AnsC1 item={item} choice={choice} setChoice={setChoice}/>
96:   } else if (item.id === 6) {
97:     return <DrawClock item={item} choice={choice} setChoice={setChoice}/>
98:   } else if (item.id === 7) {
99:     return <DrawBlock item={item} choice={choice} setChoice={setChoice}/>
100:   } else if (item.id === 8){
101:     return <Outtro item={item}/>
102:   } else {
103:     return <View><Text>error item {item.id}</Text></View>
104:   }
105: }
106:
107: export const PatientTest2 = ({ navigation }) => {
108:   const [currentSlideIndex, setCurrentSlideIndex] = useState(0)
109:   const [choice, setChoice] = useState(slides);
110:   const refSlide = useRef();
111:
112:   const updateCurrentSlideIndex = e => {
113:     const contentOffsetX = e.nativeEvent.contentOffset.x;
114:     const currentIndex = Math.round(contentOffsetX / width);
115:     setCurrentSlideIndex(currentIndex);
116:   };
117:
118:   const goToNextSlide = () => {
119:     const nextSlideIndex = currentSlideIndex + 1;
120:     if (nextSlideIndex !== slides.length) {
121:       const offset = nextSlideIndex * width;
122:       refSlide?.current.scrollToOffset({offset});
123:       setCurrentSlideIndex(currentSlideIndex + 1);

```

```

124:     }
125:     // console.log(choice); //test
126: };
127:
128: const uploadImageAsync = async (uri, callBack) => {
129:   if (uri) {
130:     console.log('image ok', uri);
131:     const imageRef = ref(storage, 'test-image/' + uuid.v4());
132:     await fetch(uri)
133:       .then(response => response.blob())
134:       .then(blob => {
135:         const uploadTask = uploadBytesResumable(imageRef, blob);
136:         uploadTask.on('state_changed', (snapshot) => {
137:           const progress = (snapshot.bytesTransferred / snapshot.totalByte
s) * 100;
138:           console.log('Upload is ' + progress + '% done');
139:           setUpLoadProgress(progress.toFixed(2));
140:         },
141:         (error) => {
142:           setErrorState(error.message);
143:         },
144:         () => {
145:           getDownloadURL(uploadTask.snapshot.ref).then((downloadURL) => {
146:             console.log('File available at', downloadURL);
147:             callBack(downloadURL);
148:           });
149:         })
150:       });
151:   } else {
152:     callBack('');
153:   }
154: }
155:
156: const putPointToFirebase = async () => {
157:   console.log(choice);
158:   const uuid_v4 = uuid.v4(); // => ex.'11edc52b-2918-4d71-9058-f7285e29d894'
159:   const docRef = doc(db, "users", auth.currentUser.uid, "patient_test", uuid_v4)
160:
161:   const userRef = doc(db, "users", auth.currentUser.uid)
162:   let payload = {
163:     type: 'test-v2',
164:     server_time: Timestamp.now(),
165:     user_time: new Date(),
166:     choice1: {
167:       title: choice[1].title,
168:       subtitle: choice[1].subtitle,
169:       text1: choice[1].text1,
170:       text2: choice[1].text2,
171:       text3: choice[1].text3,
172:       anstext1: choice[5].anstext1,
173:       anstext2: choice[5].anstext2,
174:       anstext3: choice[5].anstext3
175:     },
176:     choice2: choice[2],
177:     choice3: choice[3],
178:     choice4: choice[4],
179:   }
180:   await setDoc(docRef, payload).then(() => {
181:     console.log('upload choice 1-4');
182:   })
183:
184:   // upload choice 5 (choice[6])
185:   await uploadImageAsync(choice[6].url, async (url) => {
186:     await setDoc(docRef, {
187:       choice5: {

```

```

188:         url: url,
189:         title: choice[6].title
190:       }
191:     }, {merge: true}).then(() => {
192:       console.log('upload choice 5');
193:     })
194:   });
195:
196:   // upload choice 6 (choice[7])
197:   await uploadImageAsync(choice[7].url, async (url) => {
198:     await setDoc(docRef, {
199:       choice6: {
200:         url: url,
201:         title: choice[7].title
202:       }
203:     }, {merge: true}).then(() => {
204:       console.log('upload choice 6');
205:
206:       if (navigation !== null) {
207:         navigation.goBack();
208:       }
209:     })
210:   });
211:
212:   await setDoc(userRef, {isTest: true}, {merge: true}).then(() => {
213:     console.log('add patient moca test');
214:   })
215: }
216:
217: const Footer = () => {
218:   return (
219:     <View
220:       style={{
221:         height: height * 0.15,
222:         justifyContent: 'space-between',
223:         paddingHorizontal: 10,
224:       }}>
225:       {/* Indicator container */}
226:       <View
227:         style={{
228:           flexDirection: 'row',
229:           justifyContent: 'center',
230:           marginTop: 20,
231:         }}>
232:       {/* Render indicator */}
233:       {slides.map((_, index) => (
234:         <View
235:           key={index}
236:           style={{
237:             styles.indicator,
238:             currentSlideIndex == index && {
239:               backgroundColor: COLORS.white,
240:               width: 25,
241:             },
242:           }}
243:         />
244:       ))}
245:     </View>
246:
247:     {/* Render buttons */}
248:     <View style={{marginBottom: 20}}>
249:       {currentSlideIndex == slides.length - 1 ? (
250:         <View style={{height: 50}}>
251:           <TouchableOpacity
252:             style={styles.btn}
253:             onPress={putPointToFirebase}>

```



```
254:         <Text style={{fontWeight: 'bold', fontSize: 15}}>
255:             à,\225à,\201à,\202à,\207
256:         </Text>
257:     </TouchableOpacity>
258: </View>
259: ) : (
260:     (currentSlideIndex >= 1) ?
261:     <View style={{flexDirection: 'row'}}>
262:         <TouchableOpacity
263:             disabled={currentSlideIndex === 0 ? true : false}
264:             activeOpacity={0.8}
265:             onPress={goToNextSlide}
266:             style={styles.btn}>
267:             <Text style={{fontWeight: 'bold', fontSize: 15}}>
268:                 à,\225à¹\210à,-à¹\204à,\233
269:             </Text>
270:         </TouchableOpacity>
271:     </View> :
272:     <View style={{flexDirection: 'row'}}>
273:         <TouchableOpacity
274:             disabled={currentSlideIndex !== 0 ? true : false}
275:             activeOpacity={0.8}
276:             onPress={() => {
277:                 navigation.goBack()
278:             }}
279:             style={styles.btn}>
280:             <Text style={{fontWeight: 'bold', fontSize: 15, color: "
red"}}>
281:                 à,\201à¹\200à,\202à,\207
282:             </Text>
283:         </TouchableOpacity>
284:     </View>
285: )}
286: </View>
287: </View>
288: );
289: };
290:
291: return (
292:     <SafeAreaView style={{flex: 1, backgroundColor: Colors.orange}}>
293:         <StatusBar backgroundColor={Colors.orange} />
294:         <FlatList
295:             ref={refSlide}
296:             onMomentumScrollEnd={updateCurrentSlideIndex}
297:             contentContainerStyle={{height: height * 0.75}}
298:             showsHorizontalScrollIndicator={false}
299:             horizontal
300:             data={slides}
301:             pagingEnabled
302:             renderItem={({item}) =>
303:                 <Slide
304:                     item={item}
305:                     goNext={goToNextSlide}
306:                     choice={choice}
307:                     setChoice={setChoice}
308:                 />
309:             }
310:         />
311:         <Footer />
312:     </SafeAreaView>
313: )
314: }
315:
316: const styles = StyleSheet.create({
317:     subtitle: {
318:         color: COLORS.white,
```

```
319:         fontSize: 18,
320:         marginTop: 10,
321:         maxWidth: '70%',
322:         textAlign: 'center',
323:         lineHeight: 23,
324:     },
325:     minisubtitle: {
326:         color: COLORS.white,
327:         fontSize: 13,
328:         marginTop: 10,
329:         maxWidth: '70%',
330:         textAlign: 'center',
331:         lineHeight: 23,
332:     },
333:     title: {
334:         color: COLORS.white,
335:         fontSize: 22,
336:         fontWeight: 'bold',
337:         marginTop: 20,
338:         textAlign: 'center',
339:     },
340:     indicator: {
341:         height: 2.5,
342:         width: 10,
343:         backgroundColor: '#C36A2D',
344:         marginHorizontal: 3,
345:         borderRadius: 2,
346:     },
347:     btn: {
348:         flex: 1,
349:         height: 50,
350:         borderRadius: 5,
351:         backgroundColor: '#fff',
352:         justifyContent: 'center',
353:         alignItems: 'center',
354:     },
355: });
```



```
1: import { StyleSheet, Text, View, Dimensions} from 'react-native'
2: import React from 'react'
3: const {width, height} = Dimensions.get('window');
4: const COLORS = {primary: '#282534', white: '#fff'};
5:
6: export const Outtro = ({item}) => {
7:   return (
8:     <View style={{flex: 1, width: width, alignItems: 'center', justifyContent: '
center'}}>
9:       <View style={{alignItems: 'center', width: width * 0.9}}>
10:        <Text style={styles.title}>{item.title}</Text>
11:      </View>
12:    </View>
13:  )
14: }
15:
16: const styles = StyleSheet.create({
17:   title: {
18:     color: COLORS.white,
19:     fontSize: 22,
20:     fontWeight: 'bold',
21:     marginTop: 20,
22:     textAlign: 'center',
23:   },
24: });
```



```

1: import React, { useState, useEffect } from 'react'
2: import { StyleSheet, Text, ScrollView, Dimensions, TouchableOpacity, Image } from 'react-native'
3: import { View, LoadingIndicator } from '../components'
4:
5: import { onSnapshot, collection } from "firebase/firestore";
6: import { Colors, db, auth } from '../config';
7: const {width} = Dimensions.get('window');
8:
9: import { familyLists } from '../store';
10: import { useRecoilState } from 'recoil';
11:
12: export default function PatientFamily({ navigation }) {
13:
14:   const [familyLists, setLists] = useRecoilState(familyLists)
15:
16:   const [loading, setLoading] = useState(true)
17:
18:   useEffect(() => {
19:     // effect
20:     setLoading(true)
21:
22:     const collRef = collection(db, "users", auth.currentUser.uid, "RelationshipLists")
23:     // query not working in sub collection
24:     // const q = query(collRef, where("day", ">=", new Date()), orderBy("day"))
25:     const unsubscribe = onSnapshot(collRef, (snapshot) => {
26:       setLists(snapshot.docs.map((doc) => doc.data()))
27:       setLoading(false)
28:     });
29:
30:     return () => {
31:       // cleanup
32:       unsubscribe()
33:     }
34:   }, [])
35:
36:   const renderItemAll = () => {
37:     return familyLists.map(item => {
38:       return <View
39:         key={item.id}
40:         style={styles.item}
41:       >
42:         {!item.imageUrl ?
43:           <Image
44:             style={styles.image}
45:             source={require('../assets/avatar.webp')}
46:           /> :
47:           <Image
48:             style={styles.image}
49:             source={{uri: item.imageUrl}}
50:           />
51:         <View style={styles.itemViewText}>
52:           <Text style={styles.itemTitle}>{item.name} {item.lastName}
53:         <Text style={styles.itemTitle}>{item.relations}</Text>
54:       </View>
55:     </View>
56:   })
57: }
58:
59: return (
60:   <View isSafe style={styles.container}>
61:     <Text style={styles.header}>à, ìà, ºà, çà, \212à, •à¹\210à, -à, \204à, \231à, ìà,
62:     <TouchableOpacity

```

```

63:     style={styles.btn}
64:     onPress={() => navigation.navigate("New-person")}
65:   >
66:     <Text style={{color: 'white'}}>à¹\200à, \236à, ´à¹\210à, ìà, ìà, ºà, çà,
67:     </TouchableOpacity>
68:     <Loading ? <ScrollView>
69:       <View style={{height: 4, width: width, backgroundColor: '#f0f0f0'}}>
70:     </View>
71:     <RenderItemAll/>
72:   </ScrollView> : <LoadingIndicator/>
73: </View>
74: }
75:
76: const styles = StyleSheet.create({
77:   container: {
78:     flex: 1,
79:     alignItems: 'center',
80:     backgroundColor: 'white',
81:     // justifyContent: 'center'
82:   },
83:   header: {
84:     fontWeight: 'bold',
85:     fontSize: 30,
86:     alignSelf: 'flex-start',
87:     marginLeft: 20,
88:     marginBottom: 10,
89:   },
90:   btn: {
91:     width: width*0.9,
92:     height: 50,
93:     borderRadius: 5,
94:     backgroundColor: Colors.orange,
95:     alignItems: 'center',
96:     justifyContent: 'center',
97:     marginVertical: 12
98:   },
99:   item: {
100:     width: width,
101:     height: 'auto',
102:     paddingHorizontal: width * 0.05,
103:     borderBottomWidth: 4,
104:     borderColor: '#f0f0f0',
105:     paddingBottom: 5,
106:     flexDirection: 'row'
107:   },
108:   itemTitle: {
109:     fontSize: 20,
110:     fontWeight: '600',
111:   },
112:   itemViewText: {
113:     alignSelf: 'center',
114:     paddingLeft: 10
115:   },
116:   image: {
117:     height: 70,
118:     width: 70,
119:     borderRadius: 5,
120:     marginVertical: 12,
121:     alignSelf: 'flex-start'
122:   }
123: })
124:

```



```
1: import React from 'react';
2: import { TouchableOpacity, Text, ActivityIndicator, StyleSheet, Dimensions, Image }
from 'react-native';
3: import { PicturePuzzle } from 'react-native-picture-puzzle';
4: import { Colors } from '../.../config';
5: import { View } from '../.../components';
6: const {width, height} = Dimensions.get('window');
7:
8: function shuffle(array) {
9:   for (var i = array.length - 1; i > 0; i--) {
10:     var j = Math.floor(Math.random() * (i + 1));
11:     var temp = array[i];
12:     array[i] = array[j];
13:     array[j] = temp;
14:   }
15: }
16:
17: const ImgList = [
18:   'https://cdn3.vectorstock.com/i/thumb-large/39/62/head-of-cartoon-cow-vector-23863
962.jpg',
19:   'https://cdn.icon-icons.com/icons2/2643/PNG/512/male_boy_person_people_avatar_icon
_159358.png',
20:   'https://icons.iconarchive.com/icons/google/noto-emoji-animals-nature/256/22215-do
g-icon.png'
21: ]
22:
23: export default function Game() {
24:
25:   const [index, setIndex] = React.useState(0);
26:
27:   const [source, setSource] = React.useState({
28:     uri: 'https://icons.iconarchive.com/icons/google/noto-emoji-animals-nature/256/2
2215-dog-icon.png',
29:   });
30:
31:   const originalPieces = React.useMemo(() => (
32:     [...Array(9)].map( (_, i) => i)
33:   ), []);
34:
35:   const shuffledPieces = React.useMemo(() => {
36:     const p = [...originalPieces];
37:     shuffle(p);
38:     return p;
39:   }, [originalPieces]);
40:
41:   const [hidden, setHidden] = React.useState(0);
42:   const [pieces, setPieces] = React.useState(shuffledPieces);
43:
44:   const renderLoading = React.useCallback(() => (
45:     <View style={styles.center}>
46:       <ActivityIndicator />
47:     </View>
48:   ), []);
49:
50:   const onChange = React.useCallback((nextPieces, nextHidden) => {
51:     setPieces([...nextPieces]);
52:     setHidden(nextHidden);
53:   }, [setPieces, setHidden]);
54:
55:   const solve = React.useCallback(() => {
56:     setPieces(originalPieces);
57:     setHidden(null);
58:   }, [setPieces, originalPieces]);
59:
60:   const retry = React.useCallback(() => {
61:     setPieces(shuffledPieces);
```

```
62:     setHidden(0);
63:   }, [setPieces, shuffledPieces]);
64:
65:   const RandomImg = () => {
66:     setIndex(index => index >= 2 ? 0 : index+=1)
67:     // console.log(ImgList[index]);
68:     switch (index) {
69:       case 0:
70:         setSource({ uri: ImgList[0]})
71:         break;
72:
73:       case 1:
74:         setSource({ uri: ImgList[1]})
75:         break;
76:
77:       default:
78:         setSource({ uri: ImgList[2]})
79:         break;
80:     }
81:   }
82:
83:   return (
84:     <View isSafe style={styles.container}>
85:       <Text style={styles.header}>à¹\200à,\201à,;à¹\200à,à,à,à,\207à,à,¹à,\233à,
à,²à,\236</Text>
86:       <Text style={{fontSize: 20, fontWeight: 'bold'}}>à, à,²à,\236à,\227à,à¹\210 {
index + 1}</Text>
87:       <Image source={source} style={{width: 50, height: 50}}/>
88:       <View style={{justifyContent: "center", height: height, width: width}}>
89:         <PicturePuzzle
90:           style={styles.shadow}
91:           renderLoading={renderLoading}
92:           pieces={pieces}
93:           hidden={hidden}
94:           onChange={onChange}
95:           size={290}
96:           source={source}
97:         />
98:         <View style={[[styles.row]]>
99:           <TouchableOpacity onPress={retry} style={styles.btn}>
100:             <Text style={{color: 'white', fontSize: 20}}>
101:               à¹\200à,à¹\210à,\231à,-à,à,\201à,à,-à,\232
102:             </Text>
103:           </TouchableOpacity>
104:           <TouchableOpacity onPress={solve} style={styles.btn}>
105:             <Text style={{color: 'white', fontSize: 20}}>
106:               à¹\200à,\211à,à,à
107:             </Text>
108:           </TouchableOpacity>
109:         </View>
110:         <TouchableOpacity onPress={RandomImg} style={[[styles.btn, {marginBottom: hei
ght * 0.3, alignSelf: 'center'}]]>
111:           <Text style={{color: 'white', fontSize: 20}}>
112:             à,²à,à¹\210à,à, à,²à,\236à¹\203à,«à,à¹\210
113:           </Text>
114:         </TouchableOpacity>
115:       </View>
116:     </View>
117:   );
118: }
119:
120: const styles = StyleSheet.create({
121:   container: {
122:     flex: 1,
123:     alignItems: 'center',
124:     backgroundColor: 'white',
```

```
125:     backgroundColor: 'white',
126:     width: width
127:   },
128:   header: {
129:     fontWeight: 'bold',
130:     fontSize: 30,
131:     alignSelf: 'flex-start',
132:     marginLeft: 20,
133:     marginBottom: 10,
134:   },
135:   btn: {
136:     width: width*0.3,
137:     height: 50,
138:     borderRadius: 5,
139:     backgroundColor: Colors.orange,
140:     alignItems: 'center',
141:     justifyContent: 'center',
142:     marginVertical: 12,
143:   },
144:   center: {
145:     alignItems: 'center',
146:     justifyContent: 'center',
147:   },
148:   row: { flexDirection: 'row', justifyContent: "space-evenly", width: width},
149:   shadow: {
150:     alignSelf: 'center',
151:     shadowColor: "#555",
152:     shadowOffset: {
153:       width: 1,
154:       height: 1,
155:     },
156:     shadowOpacity: 0.6,
157:     shadowRadius: 10.0,
158:   },
159: })
```



```

1: import React, { useState, useRef } from 'react'
2: import { StyleSheet, Text, FlatList, TouchableOpacity, Dimensions, StatusBar, SafeAreaView, View } from 'react-native'
3: import { Colors, db, auth } from '.././config';
4: import { doc, setDoc, Timestamp } from "firebase/firestore";
5: import uuid from 'react-native-uuid';
6:
7: const {width, height} = Dimensions.get('window');
8: const COLORS = {primary: '#282534', white: '#fff'};
9:
10: const slides = [
11:   {
12:     id: 0,
13:     title: 'à¹\201à¹\232à¹\232à¹\204à¹,±à¹\224à¹.\201à¹,èà¹,-à¹\207à¹,à¹,jà¹,èà¹,èà¹,\226à¹,à¹,²à¹\236à¹,\204à¹,$à¹\210à¹,³ 14 à¹\204à¹,³à¹\226à¹,²à¹,í¹',
14:     subtitle: 'à¹\204à¹,³ à¹\212à¹,µà¹\211à¹\201à¹,\210à¹\207¹',
15:     des1: 'à¹\204à¹,à¹¹\201à¹,\231à¹\231 1 à¹\200à¹,jà¹,à¹¹\210à¹,-à¹¹\200à¹,«à¹\225à¹,,à¹\201à¹,²à¹,èà¹\223à¹¹\214à¹,\231à¹,±à¹¹\211à¹,\231à¹¹\204à¹,jà¹¹\210à¹¹\200à¹,\204à¹,çà¹¹\200à¹,\201à¹,´à¹,\224à¹\202à¹,¶à¹¹\211à¹,\231à¹¹\200à¹,¥à¹,çà¹,«à¹,èà¹,à¹,-à¹,-à¹,²à¹,\210à¹,\210à¹,à¹¹\200à¹,\201à¹,´à¹,\224à¹,\202à¹,¶à¹¹\211à¹,\231à¹,\231à¹,²à¹,\231à¹¹\206 à¹\204à¹,èà¹,±à¹¹\211à¹,\207 à¹¹\203à¹,\231à¹,èà¹,à¹,çà¹,à¹¹\200à¹,$à¹,¥à¹,² 1 à¹\233à¹,µ¹',
16:     des2: 'à¹\204à¹,à¹¹\201à¹,\231à¹\231 2 à¹¹\200à¹,jà¹,à¹¹\210à¹,-à¹¹\200à¹,«à¹\225à¹,,à¹\201à¹,²à¹,èà¹\223à¹¹\214à¹,\231à¹,±à¹¹\211à¹,\231à¹¹\200à¹,\201à¹,´à¹,\224à¹,\202à¹,¶à¹¹\211à¹,\231à¹¹\204à¹,jà¹¹\210à¹,\232à¹¹\210à¹,-à¹,çà¹,\231à¹,±à¹,\201à¹,«à¹,èà¹,à¹,-à¹,-à¹,²à¹,\210à¹,\210à¹,à¹¹\200à¹,\201à¹,´à¹,\224à¹\202à¹,¶à¹¹\211à¹,\231 1 à¹,«à¹,èà¹,à¹,- 2 à¹\204à¹,èà¹,±à¹¹\211à¹,\207 à¹¹\203à¹,\231à¹,èà¹,à¹,çà¹,à¹¹\200à¹,$à¹,¥à¹,² 1 à¹¹\200à¹,\224à¹,à¹,-à¹,\231¹',
17:     des3: 'à¹\204à¹,à¹¹\201à¹,\231à¹,\231 3 à¹¹\200à¹,jà¹,à¹¹\210à¹,-à¹¹\200à¹,«à¹\225à¹,,à¹\201à¹,²à¹,èà¹\223à¹¹\214à¹,\231à¹,±à¹¹\211à¹,\231à¹¹\200à¹,\201à¹,´à¹,\224à¹,\202à¹,¶à¹¹\211à¹,\231à¹,\204à¹¹\210à¹,-à¹,\231à¹,\202à¹¹\211à¹,²à¹,\207à¹,\232à¹¹\210à¹,-à¹,çà¹,«à¹,èà¹,à¹,-à¹,-à¹,²à¹,\210à¹,\210à¹,à¹¹\200à¹,\201à¹,´à¹,\224à¹,\202à¹,¶à¹¹\211à¹,\231à¹¹\200à¹,\201à¹,à¹,-à¹,\232à¹,\227à¹,,à¹\201à¹,à¹,±à¹,\233à¹,\224à¹,²à¹,«à¹¹\214¹',
18:     des4: 'à¹\204à¹,à¹¹\201à¹,\231à¹,\231 4 à¹¹\200à¹,jà¹,à¹¹\210à¹,-à¹¹\200à¹,«à¹\225à¹,,à¹\201à¹,²à¹,èà¹\223à¹¹\214à¹,\231à¹,±à¹¹\211à¹,\231à¹¹\200à¹,\201à¹,´à¹,\224à¹,\202à¹,¶à¹¹\211à¹,\231à¹¹\200à¹,\201à¹,à¹,-à¹,\232à¹,\227à¹,à¹,\201à¹,$à¹,±à¹,\231¹',
19:     description: 'à¹\227à¹,µà¹¹\210à¹,jà¹,² : à¹¹\201à¹,\233à¹,¥à¹,\210à¹,²à¹,\201à¹,\232à¹,\227à¹,\204à¹,$à¹,²à¹,jà¹¹\203à¹,\231 British Medical Association à¹¹\202à¹,\224à¹,ç à¹\212à¹,²à¹,\215à¹,\201à¹,±à¹,\215à¹,\215à¹,² à¹\225à¹,±à¹,\231à¹,\225à¹,´à¹,¥à¹,µà¹,\233à¹,´à¹,\201à¹,£ (à¹,$à¹,\227.à¹,í.à¹,\210à¹,´à¹,\225à¹,$à¹,´à¹,\227à¹,çà¹,²à¹,\204à¹,¥à¹,´à¹,\231à¹,´à¹,\201) à¹,à¹,\226à¹,²à¹,\232à¹,±à¹,\231à¹,$à¹,´à¹,\210à¹,±à¹,çà¹¹\201à¹,¥à¹,à¹,\236à¹,±à¹,\222à¹,\231à¹,²à¹,$à¹,´à¹,\227à¹,çà¹,²à¹,"à¹,²à¹,à¹,\225à¹,èà¹¹\214à¹¹\201à¹,¥à¹,à¹¹\200à¹,\227à¹,\204à¹¹\202à¹,\231à¹¹\202à¹,¥à¹,çà¹,µà¹,jà¹,«à¹,²à¹,$à¹,´à¹,\227à¹,çà¹,²à¹,¥à¹,±à¹,çà¹,jà¹,«à¹,´à¹,\224à¹,¥ à¹¹\201à¹,¥à¹,à¹,\236à¹,\215.à¹¹\202à¹,à¹,à¹,² à¹¹\200à¹,\201à¹,èà¹,´à¹,\201à¹¹\204à¹,\201à¹,èà¹,\201à¹,,à¹,¥ à¹,«à¹,\231à¹¹\210à¹,$à¹,çà¹,\233à¹,èà¹,à¹,à¹,²à¹,\227à¹,$à¹,´à¹,\227à¹,çà¹,² à¹,$à¹,´à¹,\227à¹,çà¹,²à¹,¥à¹,±à¹,çà¹¹\201à¹,\236à¹,\227à¹,çà¹,"à¹,²à¹,à¹,\225à¹,èà¹¹\214 à¹\201à¹,èà¹,,à¹,\207à¹¹\200à¹,\227à¹,\236à¹,-à¹¹\201à¹,¥à¹,à¹,$à¹,\212à¹,´à¹,èà¹,\236à¹,çà¹,²à¹,\232à¹,²à¹,¥¹',
20:     isTest: false,
21:   },
22:   {
23:     id: 1,
24:     title: 'à¹,«à¹,²à¹\202à¹,-à¹,\207à¹¹\203à¹,\212à¹¹\211à¹¹\203à¹,\231à¹,\232à¹¹\211à¹,²à¹,\231à¹¹\204à¹,jà¹¹\210à¹,\236à¹,\232¹',
25:     choose_num: null,
26:     isTest: true,
27:   },
28:   {
29:     id: 2,
30:     title: 'à¹,\210à¹,³à¹,à¹,\226à¹,²à¹,\231à¹,\227à¹,µà¹¹\210à¹,\227à¹,µà¹¹\210à¹¹\200à¹,\204à¹,çà¹¹\204à¹,\233à¹,\232à¹¹\210à¹,-à¹,çà¹¹\206 à¹¹\204à¹,jà¹¹\210à¹¹\204à¹,\224à¹¹\211¹',
31:     choose_num: null,
32:     isTest: true,
33:   },
34:   {
35:     id: 3,
36:     title: 'à¹,\225à¹¹\211à¹,-à¹,\207à¹,\201à¹,¥à¹,±à¹,\232à¹¹\204à¹,\233à¹,\227à¹,\232à¹,\227à¹,$à¹,\231à¹,\207à¹,²à¹,\231à¹,\227à¹,µà¹¹\210à¹¹\201à¹,jà¹¹\211à¹,\210à¹,à¹,\225à¹,±à¹¹\211à¹,\207à¹¹\203à¹,\210à¹,\227à¹,³ à¹,\213à¹¹\215à¹¹\211à¹,²à¹,\226à¹,¶à¹,\207 2 à¹,\204à¹,èà¹,±à¹¹\211à¹,\207¹',
37:     choose_num: null,

```

```

38:     isTest: true,
39:   },
40:   {
41:     id: 4,
42:     title: 'à¹,¥à¹,à¹,à¹,à¹,à¹,´à¹¹\210à¹,\207à¹,\202à¹,-à¹,\207à¹,\227à¹,µà¹¹\210à¹,\225à¹,±à¹¹\211à¹,\207à¹¹\203à¹,\210à¹,$à¹¹\210à¹,²à¹,\210à¹,à¹,\231à¹,³à¹¹\200à¹,-à¹,²à¹,-à¹,-à¹,\201à¹¹\204à¹,\233à¹,\231à¹,-à¹,\201à¹,\232à¹¹\211à¹,²à¹,\231à¹,\224à¹¹\211à¹,$à¹,ç¹',
43:     choose_num: null,
44:     isTest: true,
45:   },
46:   {
47:     id: 5,
48:     title: 'à¹¹\204à¹,à¹¹\210à¹,à¹,²à¹,à¹,²à¹,èà¹,\226à¹¹\200à¹,\202à¹¹\211à¹,²à¹¹\203à¹,\210à¹¹\200à¹,\231à¹,à¹¹¹\211à¹,-à¹¹\200à¹,èà¹,à¹¹\210à¹,-à¹,\207à¹¹\203à¹,\231à¹,«à¹,\231à¹,±à¹,\207à¹,à¹,à¹,à¹,-à¹,\236à¹,´à¹,à¹,\236à¹¹\214 à¹,«à¹,èà¹,à¹,-à¹,$à¹,²à¹,èà¹,à¹,²à¹,èà¹,\227à¹,µà¹¹\210à¹,-à¹¹\210à¹,²à¹,\231¹',
49:     choose_num: null,
50:     isTest: true,
51:   },
52:   {
53:     id: 6,
54:     title: 'à¹,¥à¹,à¹,à¹,à¹¹\200à¹,èà¹,à¹¹\210à¹,-à¹,\207à¹,\227à¹,µà¹¹\210à¹¹\204à¹,\224à¹¹\211à¹,èà¹,±à¹,\232à¹,\237à¹,±à¹,\207à¹,à¹,²à¹¹\200à¹,à¹,à¹¹\210à¹,-à¹,$à¹,²à¹,\231à¹,\231à¹,µà¹¹\211 à¹,«à¹,èà¹,à¹,- à¹¹\200à¹,à¹,à¹¹\210à¹,- 2-3 à¹,$à¹,±à¹,\231à¹,\201à¹¹\210à¹,-à¹,\231¹',
55:     choose_num: null,
56:     isTest: true,
57:   },
58:   {
59:     id: 7,
60:     title: 'à¹,¥à¹,à¹,à¹,\232à¹,-à¹,\201à¹,\202à¹¹\211à¹,-à¹,\204à¹,$à¹,²à¹,à¹,\227à¹,µà¹¹\210à¹,\204à¹,\231à¹,-à¹,à¹¹\210à¹,\231à¹,$à¹,²à¹,\231à¹¹\203à¹,«à¹¹\211\201à¹,à¹,²à¹,\232à¹,-à¹,\201à¹,-à¹,µà¹,\201à¹,\204à¹,\231à¹,«à¹,\231à¹,¶à¹¹\210à¹,\207¹',
61:     choose_num: null,
62:     isTest: true,
63:   },
64:   {
65:     id: 8,
66:     title: 'à¹,¥à¹,à¹,à¹,\202à¹¹\211à¹,-à¹,à¹,à¹,¥à¹,à¹¹\210à¹,$à¹,\231à¹,\225à¹,±à¹,$à¹,\202à¹,-à¹,\207à¹,\225à¹,\231à¹¹\200à¹,-à¹,\207 à¹¹\200à¹,\212à¹¹\210à¹,\231 à¹,$à¹,±à¹,\231à¹¹\200à¹,\201à¹,´à¹,\224 à¹,\227à¹,µà¹¹\210à¹,-à¹,çà¹,¹à¹¹\210¹',
67:     choose_num: null,
68:     isTest: true,
69:   },
70:   {
71:     id: 9,
72:     title: 'à¹,à¹,±à¹,\232à¹,à¹,\231à¹¹\203à¹,\231à¹,èà¹,²à¹,çà¹,¥à¹,à¹¹\200à¹,-à¹,µà¹,çà¹,\224à¹,\202à¹,-à¹,\207à¹¹\200à¹,èà¹,à¹¹\210à¹,-à¹,\207à¹,\227à¹,µà¹¹\210à¹¹\204à¹,\224à¹¹\211à¹,èà¹,±à¹,\232à¹,\237à¹,±à¹,\207à¹,à¹,²¹',
73:     choose_num: null,
74:     isTest: true,
75:   },
76:   {
77:     id: 10,
78:     title: 'à¹,¥à¹,à¹,à¹,\227à¹,µà¹¹\210à¹,\227à¹,µà¹¹\210à¹¹\200à¹,\204à¹,çà¹,$à¹,²à¹,\207à¹,à¹,´à¹¹\210à¹,\207à¹,\202à¹,-à¹,\207à¹,\231à¹,±à¹¹\211à¹,\231à¹¹\200à¹,\233à¹¹\207à¹,\231à¹,\233à¹,èà¹,à¹,\210à¹,³ à¹,«à¹,èà¹,à¹,- à¹,à¹,-à¹,\207à¹,«à¹,²à¹,à¹,´à¹¹\210à¹,\207à¹,\202à¹,-à¹,\207à¹,\231à¹,±à¹¹\211à¹,\231à¹¹\203à¹,\231à¹,\227à¹,µà¹¹\210à¹,\227à¹,µà¹¹\210à¹¹\204à¹,à¹¹\210à¹,\231à¹¹\210à¹,²à¹,\210à¹,à¹,$à¹,²à¹,\207à¹¹\204à¹,$à¹¹\211¹',
79:     choose_num: null,
80:     isTest: true,
81:   },
82:   {
83:     id: 11,
84:     title: 'à¹,\202à¹,\223à¹,à¹¹\200à¹,\224à¹,´à¹,\231à¹,\227à¹,²à¹,\207 à¹,«à¹,èà¹,à¹,-à¹¹\200à¹,\224à¹,´à¹,\231à¹¹\200à¹,¥à¹¹\210à¹,\231 à¹,«à¹,èà¹,à¹,-à¹,-à¹,çà¹,¹à¹¹\210à¹¹\203à¹,\231à¹,-à¹,²à¹,\204à¹,²à¹,èà¹,\227à¹,µà¹¹\210à¹¹\200à¹,\204à¹,çà¹¹\204à¹,\233à¹,\232à¹¹\210à¹,-à¹,ç à¹¹\206 à¹,à¹,±à¹,\201à¹¹\200à¹,\201à¹,´à¹,\224à¹,-à¹,²à¹,\201à¹,²à¹,èà¹,«à¹,¥à¹,\207à¹,\227à¹,´à¹,"à¹,«à¹,èà¹,à¹,-à¹,«à¹,¥à¹,\207à¹,\227à¹,²à¹,

```

```
\207',
  85:      choose_num: null,
  86:      isTest: true,
  87:    },
  88:    {
  89:      id: 12,
  90:      title: 'à,\225à¹\211à,-à,\207à,\227à,³à,\201à,´à,\210à,$à,±à,\225à,£à,\233à,
£à,°à,\210à,³à,$à,±à,\231à,\232à,²à,\207à,-à,çà¹\210à,²à,\207à,\213à¹\215à¹\211à,²à,\226à,¶à
,\207 2 à,\204à,£à,±à¹\211à,\207 à¹\200à,\236à,£à,²à,°à,jà,µà,\204à,$à,²à,jà,\234à,´à,\224à,
\236à,¥à,²à,\224à¹\200à,\201à,´à,\224à,\202à,¶à¹\211à,\231 à¹\200à,\212à¹\210à,\231 à¹\203à,
ªà¹\210à,\231à¹\215à¹\211à,²à,\225à,²à,¥à,jà,²à,\201à¹\200à,\201à,´à,\231à¹\204à,\233à¹\203à
,\231à¹\200à,$à,¥à,²à,\233à,£à,,à,\207à,-à,²à,«à,²à,£ à,«à,£à,°à,-à¹\200à,\224à,´à,\231à¹
\204à,\233à,«à,$à,µà,\234à,jà,\213à¹\215à¹\211à,²à,-à,µà,\201à,\204à,£à,±à¹\211à,\207 à,\213
à,¶à¹\210à,\207à¹\200à,jà,ªà¹\210à,-à,\213à,±à,\201à,\204à,£à,¹à¹\210à¹\200à,\236à,´à¹\210à,
\207à¹\204à,\224à¹\211à,«à,$à,µà,\234à,jà¹\200à,ªà,£à¹\207à,\210à¹\204à,\233´,
  91:      choose_num: null,
  92:      isTest: true,
  93:    },
  94:    {
  95:      id: 13,
  96:      title: 'à¹\200à,¥à¹\210à,²à¹\200à,£à,ªà¹\210à,-à,\207à¹\200à,\224à,´à,jà,
\213à¹\215à¹\211à,²à,-à,µà,\201à,\204à,£à,±à¹\211à,\207à,\213à,¶à¹\210à,\207à¹\200à,jà,ªà¹
\210à,-à,ªà,±à,\201à,\204à,£à,¹à¹\210à¹\200à,\236à,´à¹\210à,\207à¹\204à,\224à¹\211à¹\200à,¥à
¹\210à,²à¹\200à,ªà,£à¹\207à,\210´,
  97:      choose_num: null,
  98:      isTest: true,
  99:    },
  100:    {
  101:      id: 14,
  102:      title: 'à,¥à,ªà,jà¹\200à,\236à,ªà¹\210à,-à,\231à,ªà,\231à,´à,\227 à,«à,£à,ªà
,- à,\215à,²à,\225à,´à,ªà,\231à,´à,\227 à,«à,£à,ªà,- à,\232à,,à,\204à,\204à,¥à,\227à,µà¹\210
à,\204à,\232à,«à,²à,\201à,±à,\231à,\232à¹\210à,-à,çà¹\206´,
  103:      choose_num: null,
  104:      isTest: true,
  105:    },
  106:    {
  107:      id: 15,
  108:      title: 'à¹\200à,ªà,£à¹\207à,\210à,ªà,´à¹\211à,\231´,
  109:      subtitle: 'à,\204à,³à¹\201à,\231à,°à,\231à,³´,
  110:      isTest: false,
  111:    },
  112: ]
  113:
  114: const answer_type = [
  115:   "à¹\200à,«à,\225à,,à,\201à,²à,£à,\223à¹\214à¹\204à,jà¹\210à¹\200à,\204à,çà¹\200à
,\201à,´à,\224à,\202à,¶à¹\211à,\231à¹\200à,¥à,çà,«à,£à,ªà,-à¹\200à,\201à,´à,\224à,\202à,¶à¹
\211à,\231à,\231à,²à,\231à¹\206 à,\204à,£à,±à¹\211à,\207 à¹\203à,\231 1 à,\233à,µ",
  116:   "à¹\200à,«à,\225à,,à,\201à,²à,£à,\223à¹\214à¹\200à,\201à,´à,\224à,\202à,¶à¹\211à
,\231 1 à,«à,£à,ªà,- 2 à,\204à,£à,±à¹\211à,\207 à¹\203à,\231à,£à,°à,çà,°à¹\200à,$à,¥à,² 1 à¹
\200à,\224à,ªà,-à,\231",
  117:   "à¹\200à,«à,\225à,,à,\201à,²à,£à,\223à¹\214à¹\200à,\201à,´à,\224à,\202à,¶à¹\211à
,\231à¹\200à,\201à,ªà,-à,\232à,\227à,,à,\201à,ªà,±à,\233à,\224à,²à,«à¹\214",
  118:   "à¹\200à,«à,\225à,,à,\201à,²à,£à,\223à¹\214à¹\200à,\201à,´à,\224à,\202à,¶à¹\211à
,\231à¹\200à,\201à,ªà,-à,\232à,\227à,,à,\201à,$à,±à,\231"
  119: ]
  120:
  121: const result_type = {
  122:   a: 'à,ªà,jà,£à,£à,\226à, à,²à,\236à,\227à,²à,\207à,ªà,jà,-à,\207à,-à,çà,¹à¹\210à
¹\203à,\231à¹\200à,\201à,\223à,\221à¹\214à,\224à,µà,jà,²à,\201´,
  123:   b: 'à,ªà,jà,£à,£à,\226à, à,²à,\236à,\227à,²à,\207à,ªà,jà,-à,\207à,-à,çà,¹à¹\210à
¹\203à,\231à,£à,°à,\224à,±à,\232à,\233à,²à,\231à,\201à,¥à,²à,\207à,\204à,$à,£à¹\200à,\202à¹
\211à,²à,£à,±à,\232à,\204à,³à¹\201à,\231à,°à,\231à,³à¹\200à,\236à,ªà¹\210à,-à¹\200à,\236à,´à
¹\210à,jà,ªà,jà,£à,£à,\226à, à,²à,\236à,\227à,²à,\207à,ªà,jà,-à,\207´,
  124:   c: 'à,ªà,jà,£à,£à,\226à, à,²à,\236à,\227à,²à,\207à,ªà,jà,-à,\207à,-à,çà,¹à¹\210à
¹\203à,\231à,£à,°à,\224à,±à,\232à,\204à¹\210à,-à,\231à,\202à¹\211à,²à,\207à,\225à¹\215à¹\210
à,²à,\204à,$à,£à¹\200à,\202à¹\211à,²à,£à,±à,\232à,\204à,³à¹\201à,\231à,°à,\231à,³à¹\200à,
```

```
\236à,ªà¹\210à,-à¹\200à,\236à,´à¹\210à,jà,ªà,jà,£à,£à,\226à, à,²à,\236à,\227à,²à,\207à,ªà,jà
,-à,\207´,
  125:   d: 'à,ªà,jà,£à,£à,\226à, à,²à,\236à,\227à,²à,\207à,ªà,jà,-à,\207à,\225à¹\215à¹
\210à,² à,\204à,$à,£à,\236à,\232à¹\201à,\236à,\227à,çà¹\214´
  126: }
  127:
  128: const TestItem = ({item, clickChioce}) => {
  129:   return (
  130:     <View style={{flex: 1, width: width, alignItems: 'center'}} key={item.id}>
  131:       <View style={{alignItems: 'center', width: width * 0.9}}>
  132:         <Text style={styles.title}>à,\204à,³à,\226à,²à,jà,\202à¹\211à,-à,
\227à,µà¹\210 {item.id} à,\210à,²à,\201à,\227à,±à¹\211à,\207à,«à,jà,\224 14 à,\202à¹\211à,-<
/Text>
  133:         <Text style={styles.title}>{item.title}</Text>
  134:       </View>
  135:       {answer_type.map((value, index) =>
  136:         //bug
  137:         <TouchableOpacity
  138:           key={index}
  139:           style={{
  140:             height: 100,
  141:             flexDirection: 'row',
  142:             borderWidth: (item.choose_num == (index+1)) ? 4 : 0,
  143:             borderColor: "#fff",
  144:             borderRadius: 5,
  145:             width: width * 0.9,
  146:             backgroundColor: (item.choose_num == (index+1)) ? '#FF4848'
: '#fff', //FF4848 red 9DDAC6 green
  147:             alignItems: 'center',
  148:             marginVertical: 5
  149:           }}
  150:           onPress={() => clickChioce(item.id, index+1)}
  151:         >
  152:           <Text
  153:             style={{
  154:               marginHorizontal: 20,
  155:               fontWeight: 'bold',
  156:               fontSize: 20,
  157:               color: (item.choose_num == (index+1)) ? '#fff' : Colors.
black
  158:             }}
  159:           >
  160:             {value}
  161:           </Text>
  162:         </TouchableOpacity>
  163:       )}
  164:     </View>
  165:   )
  166: }
  167:
  168: const Intro = ({item, goNext}) => {
  169:   return (
  170:     <View style={{flex: 1, width: width, alignItems: 'center', justifyContent: '
center'}}>>
  171:     <View style={{alignItems: 'center', width: width * 0.9}}>
  172:       <Text style={styles.title}>{item.title}</Text>
  173:       <Text style={styles.subtitle}>{item.des1}</Text>
  174:       <Text style={styles.subtitle}>{item.des2}</Text>
  175:       <Text style={styles.subtitle}>{item.des3}</Text>
  176:       <Text style={styles.subtitle}>{item.des4}</Text>
  177:       <Text style={styles.minisubtitle}>{item.description}</Text>
  178:     <TouchableOpacity
  179:       style={{
  180:         height: 60,
  181:         width: width * 0.9,
  182:         backgroundColor: "#fff",
```

```

183:         borderRadius: 5,
184:         alignItems: 'center',
185:         justifyContent: 'center',
186:         marginVertical: 5
187:     }}
188:     onPress={goNext}
189:   >
190:     <Text>à¹\200à,à, 'à¹\210à,;à,\201à,±à,\231à¹\200à,à,¢</Text>
191:   </TouchableOpacity>
192: </View>
193: </View>
194: )
195: }
196:
197: const Outtro = ({item, point}) => {
198:
199:   const Comment = () => {
200:     if (point >= 14 && point <= 19) {
201:       return <Text style={styles.title}>{result_type.a}</Text>
202:     } else if (point >= 20 && point <= 29) {
203:       return <Text style={styles.title}>{result_type.b}</Text>
204:     } else if (point >= 30 && point <= 39) {
205:       return <Text style={styles.title}>{result_type.c}</Text>
206:     } else if (point >= 40 && point <= 56) {
207:       return <Text style={styles.title}>{result_type.d}</Text>
208:     } else {
209:       return <Text style={styles.title}></Text>
210:     }
211:   }
212:   return (
213:     <View style={{flex: 1, width: width, alignItems: 'center', justifyContent: '
center'}}>
214:       <View style={{alignItems: 'center', width: width * 0.9}}>
215:         <Text style={styles.title}>{item.title}</Text>
216:         <Text style={styles.title}>{item.subtitle}</Text>
217:         <Comment />
218:       </View>
219:     </View>
220:   )
221: }
222:
223: const Slide = ({item, clickChioce, goNext, point}) => {
224:   if (item.isTest) {
225:     return <TestItem item={item} clickChioce={clickChioce}/>
226:   } else if (!item.isTest && item.id == 0) {
227:     return <Intro item={item} goNext={goNext} />
228:   } else if (!item.isTest && item.id == 15) {
229:     return <Outtro item={item} point={point}/>
230:   }
231: }
232:
233: export const PatientSimpleTest = ({navigation}) => {
234:   const [currentSlideIndex, setCurrentSlideIndex] = useState(0)
235:   const [choice, setChoice] = useState(slides);
236:   const [point, setPoint] = useState(0);
237:   const ref = useRef();
238:
239:   const updateCurrentSlideIndex = e => {
240:     const contentOffsetX = e.nativeEvent.contentOffset.x;
241:     const currentIndex = Math.round(contentOffsetX / width);
242:     setCurrentSlideIndex(currentIndex);
243:   };
244:
245:   const goToNextSlide = () => {
246:     const nextSlideIndex = currentSlideIndex + 1;
247:     if (nextSlideIndex != slides.length) {

```

```

248:     const offset = nextSlideIndex * width;
249:     ref?.current.scrollToOffset({offset});
250:     setCurrentSlideIndex(currentSlideIndex + 1);
251:   }
252: };
253:
254: const goToPrevSlide = () => {
255:   const prevSlideIndex = currentSlideIndex - 1;
256:   if (prevSlideIndex > 0) {
257:     const offset = prevSlideIndex * width;
258:     ref?.current.scrollToOffset({offset});
259:     setCurrentSlideIndex(currentSlideIndex - 1);
260:   }
261: };
262:
263: const click_choice = (id, answer) => {
264:   console.log(id, answer);
265:   var new_choice = choice;
266:   new_choice[id].choose_num = answer;
267:   setChoice(new_choice);
268:   // console.log(new_choice);
269:   // console.log(choice[id]);
270:   if (id === 14) { //last choice
271:     calculatePoint()
272:   }
273:   goToNextSlide();
274: }
275:
276: const calculatePoint = () => {
277:   console.log('calculate point');
278:   choice.map(item => {
279:     if (item.choose_num) {
280:       setPoint(prev => prev + item.choose_num);
281:     }
282:   })
283: }
284:
285: const putPointToFirebase = async () => {
286:   console.log(point);
287:   const uuid_v4 = uuid.v4(); // => ex.'11edc52b-2918-4d71-9058-f7285e29d894'
288:   const docRef = doc(db, "users", auth.currentUser.uid, "patient_test", uuid_v4)
289:
290:   const userRef = doc(db, "users", auth.currentUser.uid)
291:   let payload = {
292:     server_time: Timestamp.now(),
293:     user_time: new Date(),
294:     score: point,
295:     description: '',
296:     uuid: uuid_v4,
297:     choice: choice,
298:     type: 'simple'
299:   }
300:   if (point >= 14 && point <= 19) {
301:     payload.description = result_type.a;
302:   } else if (point >= 20 && point <= 29) {
303:     payload.description = result_type.b;
304:   } else if (point >= 30 && point <= 39) {
305:     payload.description = result_type.c;
306:   } else if (point >= 40 && point <= 56) {
307:     payload.description = result_type.d;
308:   }
309:   await setDoc(docRef, payload).then(() => {
310:     console.log('add patient_test-point');
311:     if (navigation != null) {
312:       navigation.goBack();

```

```

313:     })
314:     await setDoc(userRef, {isTest: true}, {merge: true}).then(() => {
315:       console.log('add patient test');
316:     })
317:   }
318:
319:   const Footer = () => {
320:     return (
321:       <View
322:         style={{
323:           height: height * 0.15,
324:           justifyContent: 'space-between',
325:           paddingHorizontal: 10,
326:         }}>
327:         { /* Indicator container */ }
328:         <View
329:           style={{
330:             flexDirection: 'row',
331:             justifyContent: 'center',
332:             marginTop: 20,
333:           }}>
334:           { /* Render indicator */ }
335:           {slides.map((_, index) => (
336:             <View
337:               key={index}
338:               style={[
339:                 styles.indicator,
340:                 currentSlideIndex == index && {
341:                   backgroundColor: COLORS.white,
342:                   width: 25,
343:                 },
344:               ]}
345:             />
346:           ))}
347:         </View>
348:
349:         { /* Render buttons */ }
350:         <View style={{marginBottom: 20}}>
351:           {currentSlideIndex == slides.length - 1 ? (
352:             <View style={{height: 50}}>
353:               <TouchableOpacity
354:                 style={styles.btn}
355:                 onPress={putPointToFirebase}>
356:                 <Text style={{fontWeight: 'bold', fontSize: 15}}>
357:                   à,\225à,\201à,\207
358:                 </Text>
359:               </TouchableOpacity>
360:             </View>
361:           ) : (
362:             {currentSlideIndex >= 2} ?
363:             <View style={{flexDirection: 'row'}}>
364:               <TouchableOpacity
365:                 disabled={currentSlideIndex === 0 ? true : false}
366:                 activeOpacity={0.8}
367:                 onPress={goToPrevSlide}
368:                 style={styles.btn}>
369:                 <Text style={{fontWeight: 'bold', fontSize: 15}}>
370:                   à,\202à¹\211à,-à,\201à¹\210à,-à,\231à,«à,\231à¹\211à
371:                 </Text>
372:               </TouchableOpacity>
373:             </View> : (
374:               {currentSlideIndex === 0} ?
375:               <View style={{flexDirection: 'row'}}>
376:                 <TouchableOpacity
377:                   disabled={currentSlideIndex !== 0 ? true : false}
378:                   activeOpacity={0.8}
379:                   onPress={() => {
380:                     navigation.goBack()
381:                   }}
382:                   style={styles.btn}>
383:                   <Text style={{fontWeight: 'bold', fontSize: 15,
384:                     à,çà,\201à¹\200à,\207
385:                   </Text>
386:                   </TouchableOpacity>
387:                 </View> : null
388:               )
389:             </View>
390:           )
391:         </View>
392:       );
393:   };
394:
395:   return (
396:     <SafeAreaView style={{flex: 1, backgroundColor: Colors.orange}}>
397:       <StatusBar backgroundColor={Colors.orange} />
398:       <FlatList
399:         ref={ref}
400:         onMomentumScrollEnd={updateCurrentSlideIndex}
401:         contentContainerStyle={{height: height * 0.75}}
402:         showsHorizontalScrollIndicator={false}
403:         horizontal
404:         data={slides}
405:         pagingEnabled
406:         renderItem={({item}) =>
407:           <Slide
408:             item={item}
409:             clickChioce={click_choice}
410:             goNext={goToNextSlide}
411:             point={point}
412:           />
413:         )
414:       </FlatList>
415:       <Footer />
416:     </SafeAreaView>
417:   )
418: }
419:
420: const styles = StyleSheet.create({
421:   subtitle: {
422:     color: COLORS.white,
423:     fontSize: 18,
424:     marginTop: 10,
425:     maxWidth: '70%',
426:     textAlign: 'center',
427:     lineHeight: 23,
428:   },
429:   minisubtitle: {
430:     color: COLORS.white,
431:     fontSize: 10,
432:     marginTop: 10,
433:     maxWidth: '70%',
434:     textAlign: 'center',
435:     lineHeight: 23,
436:   },
437:   title: {
438:     color: COLORS.white,
439:     fontSize: 22,
440:     fontWeight: 'bold',
441:     marginTop: 20,

```

```
442:     textAlign: 'center',
443:   },
444:   indicator: {
445:     height: 2.5,
446:     width: 10,
447:     backgroundColor: '#C36A2D',
448:     marginHorizontal: 3,
449:     borderRadius: 2,
450:   },
451:   btn: {
452:     flex: 1,
453:     height: 50,
454:     borderRadius: 5,
455:     backgroundColor: '#fff',
456:     justifyContent: 'center',
457:     alignItems: 'center',
458:   },
459: });
```



```
66:     link_item_text: {
67:         fontSize: 20,
68:         fontWeight: "bold",
69:         color: 'white'
70:     }
71: })
72:
```



```

123:         value={name}
124:         onChangeText={setName}
125:     />
126:     <TextInput
127:       placeholder='à,\231à,\2à,jà,à,\201à,,à,¥'
128:       multiline={true}
129:       style={styles.titleInput}
130:       value={lastname}
131:       onChangeText={setLastname}
132:     />
133:     <TextInput
134:       placeholder='à,\204à,\2à,jà,à,\236à,\231à,\230à\
\214'
135:       multiline={true}
136:       style={styles.titleInput}
137:       value={relations}
138:       onChangeText={setRelations}
139:     />
140:     <Text style={styles.textStyle}>à\200à,\236à,\210à,jà,à,\2à,
\236</Text>
141:     {imageUrl ?
142:       <Image
143:         style={{
144:           height: width*0.7,
145:           width: width*0.7,
146:           borderRadius: 5,
147:           margin: 15
148:         }}
149:         source={{uri: imageUrl}}
150:       /> :
151:       null
152:     }
153:     <Pressable
154:       style={[styles.button, styles.buttonOpen]}
155:       onPress={pickImage}
156:     >
157:       <Text style={styles.textStyle}>à\200à,\2à,à,\201à,à,\2à,
\236</Text>
158:     </Pressable>
159:     <Pressable
160:       disabled={!enable}
161:       style={[styles.button, styles.buttonSummit, {marginBottom: 1
00, opacity: enable ? 1 : 0.5}]}
162:       onPress={uploadItemList}
163:     >
164:       <Text style={styles.textStyle}>à,\225à,\201à,\2à,\207</Text>
165:     </Pressable>
166:   </View>
167: </KeyboardAwareScrollView>
168: </View>
169: )
170: }
171:
172: const styles = StyleSheet.create({
173:   container: {
174:     flex: 1,
175:     alignItems: 'center',
176:     backgroundColor: 'white',
177:     // justifyContent: 'center'
178:   },
179:   button: {
180:     justifyContent: 'center',
181:     width: width*0.9,
182:     height: 50,
183:     borderRadius: 5,
184:     marginVertical: 10

```

```

185:   },
186:   buttonOpen: {
187:     backgroundColor: Colors.orange,
188:   },
189:   buttonSummit: {
190:     backgroundColor: '#1597e5',
191:   },
192:   textStyle: {
193:     color: "white",
194:     fontWeight: "bold",
195:     textAlign: "center"
196:   },
197:   textHeader: {
198:     fontWeight: 'bold',
199:     fontSize: 30,
200:     marginLeft: 20,
201:     marginBottom: 10
202:   },
203:   viewHeader: {
204:     width: width*0.9,
205:     flexDirection: 'row',
206:     justifyContent: 'flex-start',
207:     alignItems: 'center',
208:     alignSelf: 'center'
209:   },
210:   titleInput:{
211:     height: 50,
212:     width: width*0.9,
213:     margin: 12,
214:     padding: 10,
215:     borderRadius: 5,
216:     fontSize: 20,
217:     backgroundColor: "#f7f7f7",
218:   },
219:   modalView: {
220:     margin: 20,
221:     backgroundColor: "white",
222:     borderRadius: 5,
223:     padding: 50,
224:     alignItems: "center",
225:     shadowColor: "#000",
226:     shadowOffset: {
227:       width: 0,
228:       height: 2
229:     },
230:     shadowOpacity: 0.25,
231:     shadowRadius: 4,
232:     elevation: 5
233:   },
234: })

```

```
1: import React from 'react';
2: import { TextInput as RNTextInput } from 'react-native';
3:
4: import { View } from './View';
5: import { Icon } from './Icon';
6: import { Button } from './Button';
7: import { Colors } from '../config';
8:
9: export const TextInput = ({
10:   width = '100%',
11:   leftIconName,
12:   rightIcon,
13:   handlePasswordVisibility,
14:   ...otherProps
15: }) => {
16:   return (
17:     <View
18:       style={{
19:         backgroundColor: Colors.white,
20:         borderRadius: 8,
21:         flexDirection: 'row',
22:         padding: 12,
23:         marginVertical: 12,
24:         width,
25:         borderWidth: 1,
26:         borderColor: Colors.mediumGray
27:       }}
28:     >
29:     {leftIconName ? (
30:       <Icon
31:         name={leftIconName}
32:         size={22}
33:         color={Colors.mediumGray}
34:         style={{ marginRight: 10 }}
35:       />
36:     ) : null}
37:     <RNTextInput
38:       style={{
39:         flex: 1,
40:         width: '100%',
41:         fontSize: 18,
42:         color: Colors.black
43:       }}
44:       placeholderTextColor={Colors.mediumGray}
45:       {...otherProps}
46:     />
47:     {rightIcon ? (
48:       <Button onPress={handlePasswordVisibility}>
49:         <Icon
50:           name={rightIcon}
51:           size={22}
52:           color={Colors.mediumGray}
53:           style={{ marginRight: 10 }}
54:         />
55:       </Button>
56:     ) : null}
57:   </View>
58: );
59: };
60:
```



```
1: import React from 'react';
2: import { ActivityIndicator, StyleSheet } from 'react-native';
3:
4: import { Colors } from '../config';
5: import { View } from './View';
6:
7: export const LoadingIndicator = () => {
8:   return (
9:     <View style={styles.container}>
10:      <ActivityIndicator size='large' color={Colors.orange} />
11:    </View>
12:  );
13: };
14:
15: const styles = StyleSheet.create({
16:   container: {
17:     flex: 1,
18:     justifyContent: 'center',
19:     alignItems: 'center'
20:   }
21: });
```



```
1: import React from 'react';
2: import { MaterialCommunityIcons } from '@expo/vector-icons';
3:
4: export const Icon = ({ name, size, color, style }) => {
5:   return (
6:     <MaterialCommunityIcons
7:       name={name}
8:       size={size}
9:       color={color}
10:      style={style}
11:    />
12:   );
13: };
14:
```



```
1: import { TextInput } from './TextInput';
2: import { View } from './View';
3: import { Logo } from './Logo';
4: import { Icon } from './Icon';
5: import { Button } from './Button';
6: import { FormErrorMessage } from './FormErrorMessage';
7: import { LoadingIndicator } from './LoadingIndicator';
8:
9: export {
10:   View,
11:   TextInput,
12:   Logo,
13:   Icon,
14:   Button,
15:   FormErrorMessage,
16:   LoadingIndicator
17: };
```



```
1: import React, { useCallback } from 'react';
2: import { Pressable, Text, StyleSheet } from 'react-native';
3:
4: import { Colors } from '../config';
5:
6: export const Button = ({
7:   children,
8:   onPress,
9:   activeOpacity = 0.3,
10:  borderless = false,
11:  title,
12:  style,
13:  disabled
14: }) => {
15:   const _style = useCallback(({ pressed }) => [
16:     style,
17:     { opacity: pressed ? activeOpacity : 1 },
18:     { opacity: disabled ? 0.5 : 1}
19:   ]);
20:
21:   if (borderless) {
22:     return (
23:       <Pressable onPress={onPress} style={_style} disabled={disabled ? disabled : false}
24:       >
25:         <Text style={styles.borderlessButtonText}>{title}</Text>
26:       </Pressable>
27:     );
28:   }
29:   return (
30:     <Pressable onPress={onPress} style={_style} disabled={disabled ? disabled : false}
31:     >
32:       {children}
33:     </Pressable>
34:   );
35: };
36: const styles = StyleSheet.create({
37:   borderlessButtonText: {
38:     fontSize: 16,
39:     color: Colors.blue
40:   }
41: });
42:
```



```
1: import React from 'react';
2: import { View as RNView, StyleSheet } from 'react-native';
3: import { useSafeAreaInsets } from 'react-native-safe-area-context';
4:
5: export const View = ({ isSafe, style, children }) => {
6:   const insets = useSafeAreaInsets();
7:
8:   if (isSafe) {
9:     return (
10:      <RNView style={{ paddingTop: insets.top, ...style }}>{children}</RNView>
11:    );
12:   }
13:
14:   return <RNView style={StyleSheet.flatten(style)}>{children}</RNView>;
15: };
```



```
1: import React from 'react';
2: import { Image, StyleSheet } from 'react-native';
3:
4: import { Images } from '../config';
5:
6: export const Logo = ({ uri }) => {
7:   return <Image source={uri} style={styles.image} />;
8: };
9:
10: const styles = StyleSheet.create({
11:   image: {
12:     width: 200,
13:     height: 200
14:   }
15: });
16:
```



```
1: import React from 'react';
2: import { StyleSheet, Text } from 'react-native';
3:
4: import { Colors } from '../config';
5:
6: export const FormErrorMessage = ({ error, visible }) => {
7:   if (!error || !visible) {
8:     return null;
9:   }
10:
11:   return <Text style={styles.errorText}>{error}</Text>;
12: };
13:
14: const styles = StyleSheet.create({
15:   errorText: {
16:     marginLeft: 15,
17:     color: Colors.red,
18:     fontSize: 16,
19:     marginVertical: 8,
20:     fontWeight: '600'
21:   }
22: });
23:
```



```
1: module.exports = function(api) {  
2:   api.cache(true);  
3:   return {  
4:     presets: ['babel-preset-expo'],  
5:   };  
6: };
```



```
1: import { useState } from 'react';
2:
3: export const useTogglePasswordVisibility = () => {
4:   // password will not be initially visible
5:   const [passwordVisibility, setPasswordVisibility] = useState(true);
6:   const [rightIcon, setRightIcon] = useState('eye');
7:   const [confirmPasswordIcon, setConfirmPasswordIcon] = useState('eye');
8:   const [confirmPasswordVisibility, setConfirmPasswordVisibility] =
9:     useState(true);
10:
11:   // function that toggles password visibility on a TextInput component on a password field
12:   const handlePasswordVisibility = () => {
13:     if (rightIcon === 'eye') {
14:       setRightIcon('eye-off');
15:       setPasswordVisibility(!passwordVisibility);
16:     } else if (rightIcon === 'eye-off') {
17:       setRightIcon('eye');
18:       setPasswordVisibility(!passwordVisibility);
19:     }
20:   };
21:
22:   // function that toggles password visibility on a TextInput component on a confirm password field
23:   const handleConfirmPasswordVisibility = () => {
24:     if (confirmPasswordIcon === 'eye') {
25:       setConfirmPasswordIcon('eye-off');
26:       setConfirmPasswordVisibility(!confirmPasswordVisibility);
27:     } else if (confirmPasswordIcon === 'eye-off') {
28:       setConfirmPasswordIcon('eye');
29:       setConfirmPasswordVisibility(!confirmPasswordVisibility);
30:     }
31:   };
32:
33:   return {
34:     passwordVisibility,
35:     handlePasswordVisibility,
36:     rightIcon,
37:     confirmPasswordVisibility,
38:     handleConfirmPasswordVisibility,
39:     confirmPasswordIcon
40:   };
41: };
```



```
1: import { useTogglePasswordVisibility } from './useTogglePasswordVisibility';  
2:  
3: export { useTogglePasswordVisibility };
```



```
1: import React from 'react';
2: import { SafeAreaProvider } from 'react-native-safe-area-context';
3: import { RecoilRoot } from 'recoil';
4:
5: import { RootNavigator } from './navigation/RootNavigator';
6: import { AuthenticatedUserProvider } from './providers';
7:
8: const App = () => {
9:   return (
10:     <AuthenticatedUserProvider>
11:       <RecoilRoot>
12:         <SafeAreaProvider>
13:           <RootNavigator />
14:         </SafeAreaProvider>
15:       </RecoilRoot>
16:     </AuthenticatedUserProvider>
17:   );
18: };
19:
20: export default App;
```



```

1: import { atom, selector } from "recoil";
2:
3: // current day in microseconds with set hours, minutes, and seconds to 0
4: export const dayAtom = atom({
5:   key: 'day-atom',
6:   default: new Date().getSeconds
7: })
8:
9:
10: //====patient =====//
11: export const userInfoAtom = atom({
12:   key: 'user-infomation',
13:   default: {}
14: })
15:
16: export const userListsAtom = atom({
17:   key: 'user-lists',
18:   default: []
19: })
20:
21: export const sortListsSelector = selector({
22:   key: 'sort-lists',
23:   get: ({get}) => {
24:     const temp = get(userListsAtom)
25:     const lists = temp.filter(e => e) // create new array for fix attempted read on
ly property
26:     return lists.sort((a,b) => {
27:       if (a && b) {
28:         return a.day.seconds - b.day.seconds
29:       }
30:       return 1
31:     })
32:   }
33: })
34:
35: export const listLastSevenDays = selector({
36:   key: 'lists-last-7-days',
37:   get: ({get}) => (get(sortListsSelector).filter(e => get(dayAtom) - (e.day.seconds
* 1000) >= (60 * 60 * 24 * 7)))
38: });
39:
40: export const listAfterCurrent = selector({
41:   key: 'lists-after-current-day',
42:   get: ({get}) => (get(sortListsSelector).filter(e => (e.day.seconds * 1000) - get(d
ayAtom) >= 0))
43: });
44:
45: export const userListsDone = selector({
46:   key: 'lists-done',
47:   get: ({get}) => (get(listAfterCurrent).filter(e => e.isDone == true))
48: });
49:
50: export const userListsNotDone = selector({
51:   key: 'lists-not-done',
52:   get: ({get}) => (get(listAfterCurrent).filter(e => e.isDone == false))
53: });
54:
55: export const userMeetDocs = atom({
56:   key: 'users-meet-doctor',
57:   default: []
58: })
59:
60: export const userSortMeetDocs = selector({
61:   key: 'sort-users-meet-doctor',
62:   get: ({get}) => {
63:     const temp = get(userMeetDocs)

```

```

64:     const lists = temp.filter(e => e) // create new array for fix attempted read on
ly property
65:     return lists.sort((a,b) => {
66:       if (a && b) {
67:         return a.time.seconds - b.time.seconds
68:       }
69:       return 1
70:     })
71:   }
72: })
73:
74: export const familyLists = atom({
75:   key: 'family-lists',
76:   default: []
77: })
78:
79: //====doctor and caretaker=====//
80:
81: export const allPatientsAtom = atom({
82:   key: 'all-patients',
83:   default: []
84: })
85:
86: export const currentPatientMeetDocs = atom({
87:   key: 'current-patient-meet-doctor',
88:   default: []
89: })
90:
91: export const sortcurrentPatientMeetDocs = selector({
92:   key: 'sort-current-patient-meet-doctor',
93:   get: ({get}) => {
94:     const temp = get(currentPatientMeetDocs)
95:     const lists = temp.filter(e => e) // create new array for fix attempted read on
ly property
96:     return lists.sort((a,b) => {
97:       if (a && b) {
98:         return b.time.seconds - a.time.seconds
99:       }
100:       return 1
101:     })
102:   }
103: })
104:
105: export const currentPatientTests = atom({
106:   key: 'current-patient-tests',
107:   default: []
108: })
109:
110: export const sortcurrentPatientTests = selector({
111:   key: 'sort-current-patient-tests',
112:   get: ({get}) => {
113:     const temp = get(currentPatientTests)
114:     const lists = temp.filter(e => e) // create new array for fix attempted read on
ly property
115:     return lists.sort((a,b) => {
116:       if (a && b) {
117:         return b.user_time.seconds - a.user_time.seconds
118:       }
119:       return 1
120:     })
121:   }
122: })

```