EASTERN INTERNATIONAL UNIVERSITY

SCHOOL OF COMPUTING

AND INFORMATION TECHNOLOGY

Lab 4

Practice Assignment – Quarter 2, 2023-2024

Course Name: MOBILE PROGRAMMING

Course Code: CSE 441

Student's Full Name:

Student ID:

REDUX TOOLKIT& ASYNCSTORAGE & REACT NAVIGATION

Create github Lab4 with Lab4 folder on computer. Submit assignments to the moodle system.

Create project Lab4, Develop contact management mobile apps like the interfaces below, Using Redux toolkit and React Navigation

Install libraries:

Redux toolkit: https://redux-toolkit.js.org/usage/usage-guide

AsyncStorage: https://reactnative.dev/docs/asyncstorage

React navigation: https://reactnavigation.org/docs/stack-navigator

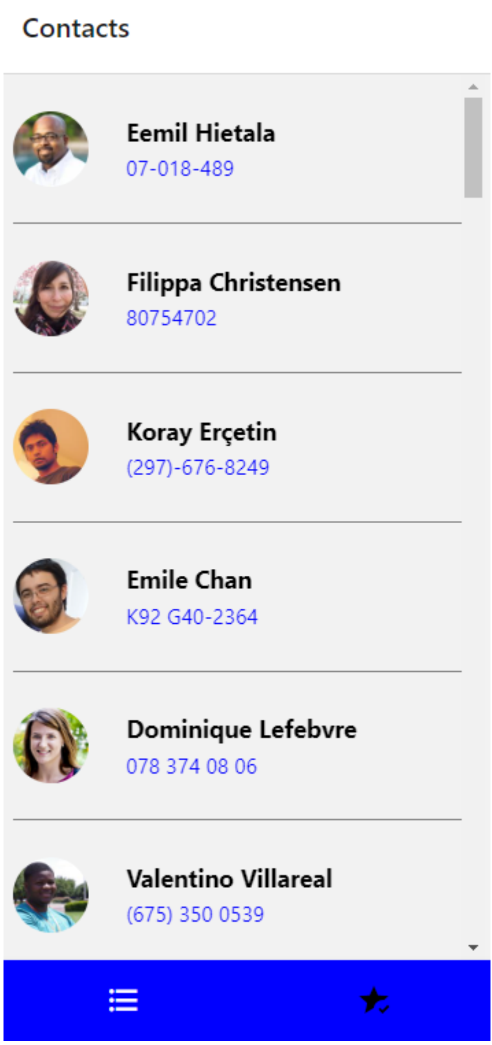
https://reactnavigation.org/docs/drawer-navigator

https://reactnavigation.org/docs/material-bottom-tab-navigator

https://reactnavigation.org/docs/material-top-tab-navigator

React native paper: https://callstack.github.io/react-native-paper/

UUID: https://github.com/uuidjs/uuid



Contacts

Eemil Hietala
07-018-489

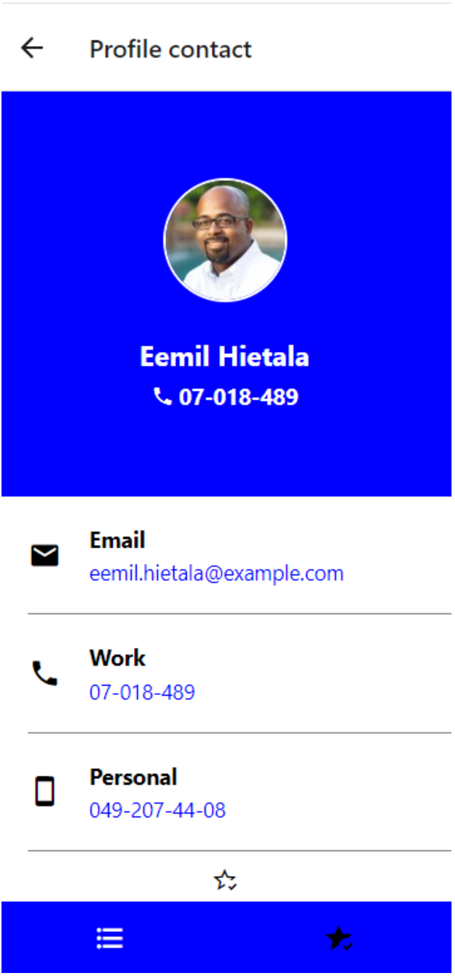
Filippa Christensen
80754702

Koray Erçetin
(297)-676-8249

Emile Chan
K92 G40-2364

Dominique Lefebvre
078 374 08 06

Valentino Villareal
(675) 350 0539



Profile contact

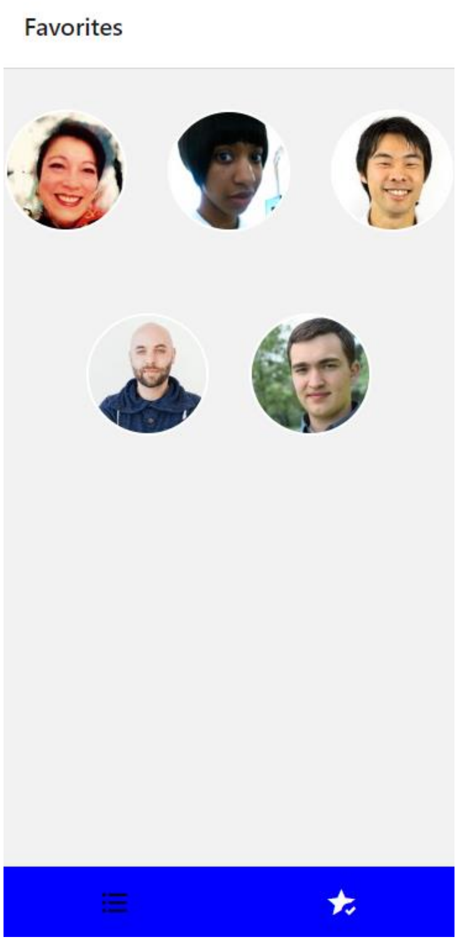
Eemil Hietala
☎ 07-018-489

Email
eemil.hietala@example.com

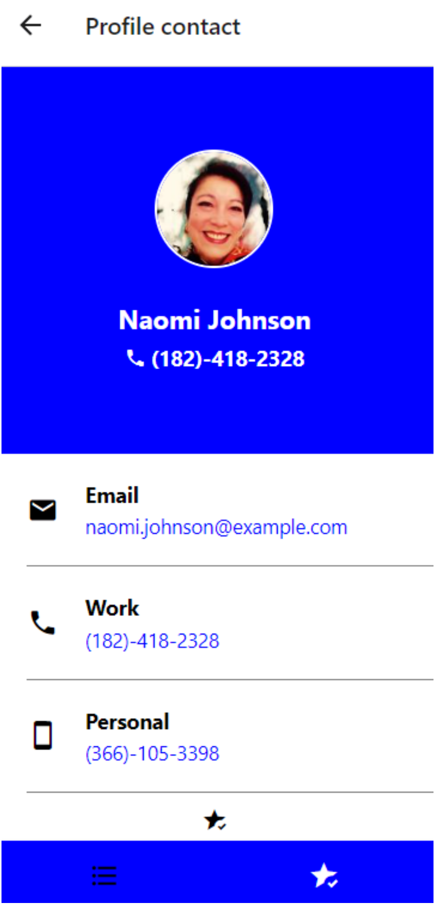
Work
07-018-489

Personal
049-207-44-08

Screen 1 Screen 2



Screen 3



Naomi Johnson

(182)-418-2328

Email
naomi.johnson@example.com

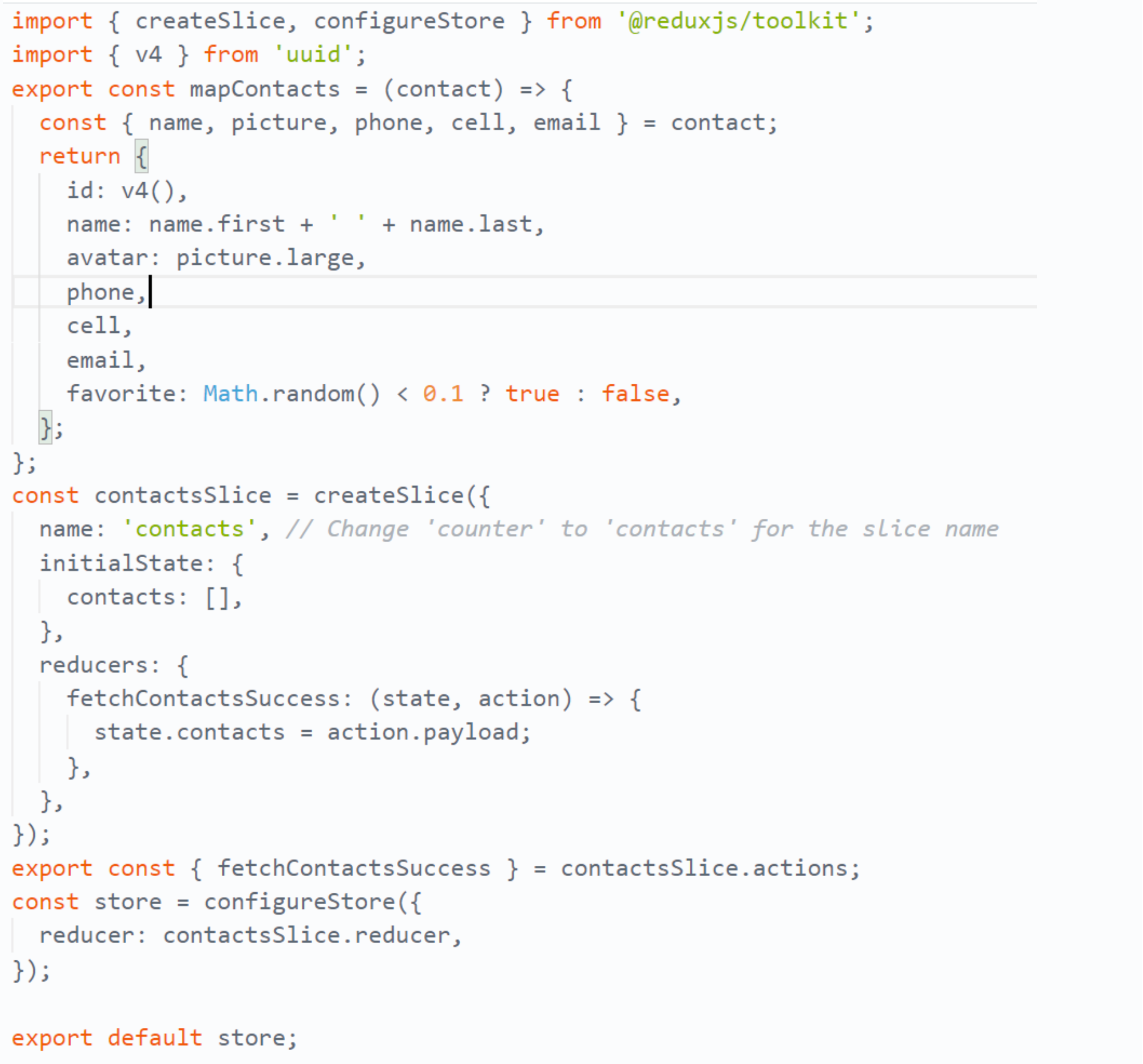
Work
(182)-418-2328

Personal
(366)-105-3398

Screen 4

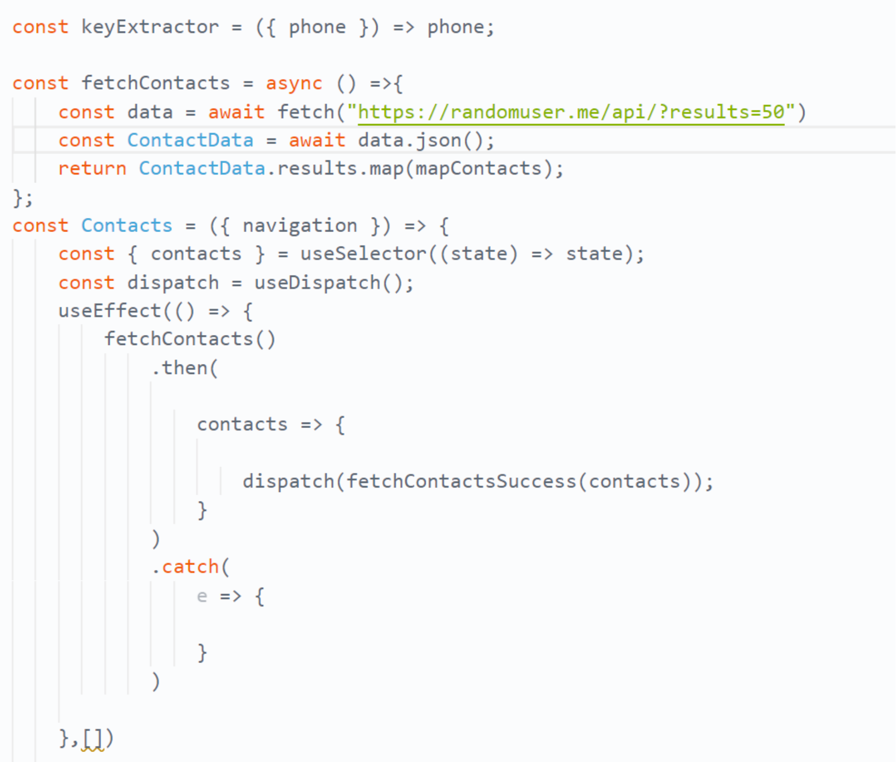
API: https://randomuser.me/api/?results=50

1. Create a "src" folder to contain the source code.
2. Create files Store.js with content.
3. Install the Redux toolkit library

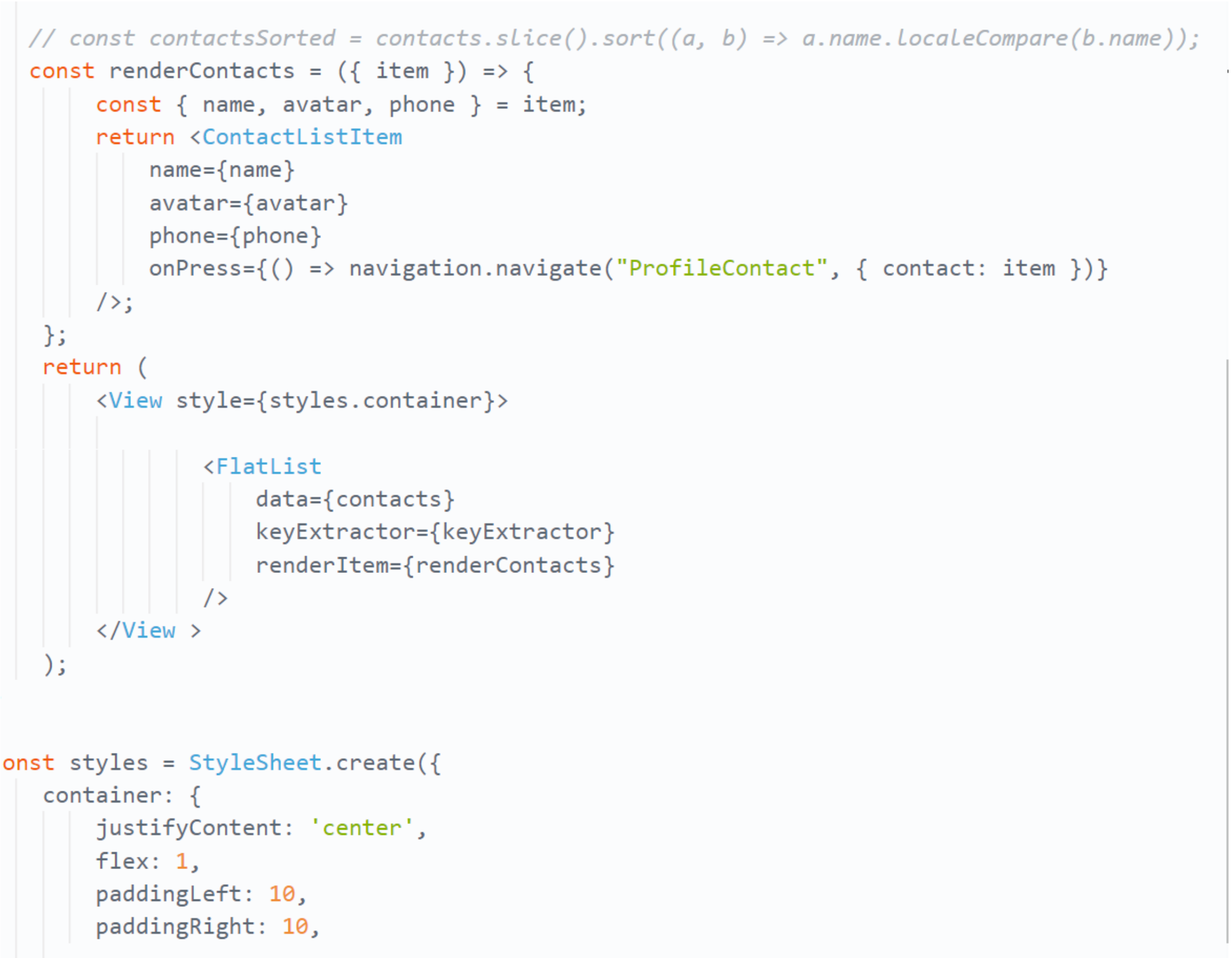


```
import { createSlice, configureStore } from '@reduxjs/toolkit';
import { v4 } from 'uuid';
export const mapContacts = (contact) => {
  const { name, picture, phone, cell, email } = contact;
  return {
    id: v4(),
    name: name.first + ' ' + name.last,
    avatar: picture.large,
    phone,
    cell,
    email,
    favorite: Math.random() < 0.1 ? true : false,
  };
};
const contactsSlice = createSlice({
  name: 'contacts', // Change 'counter' to 'contacts' for the slice name
  initialState: {
    contacts: [],
  },
  reducers: {
    fetchContactsSuccess: (state, action) => {
      state.contacts = action.payload;
    },
  },
});
export const { fetchContactsSuccess } = contactsSlice.actions;
const store = configureStore({
  reducer: contactsSlice.reducer,
});

export default store;
```

4. Design "screen 1" with the source code below: File Contact.js

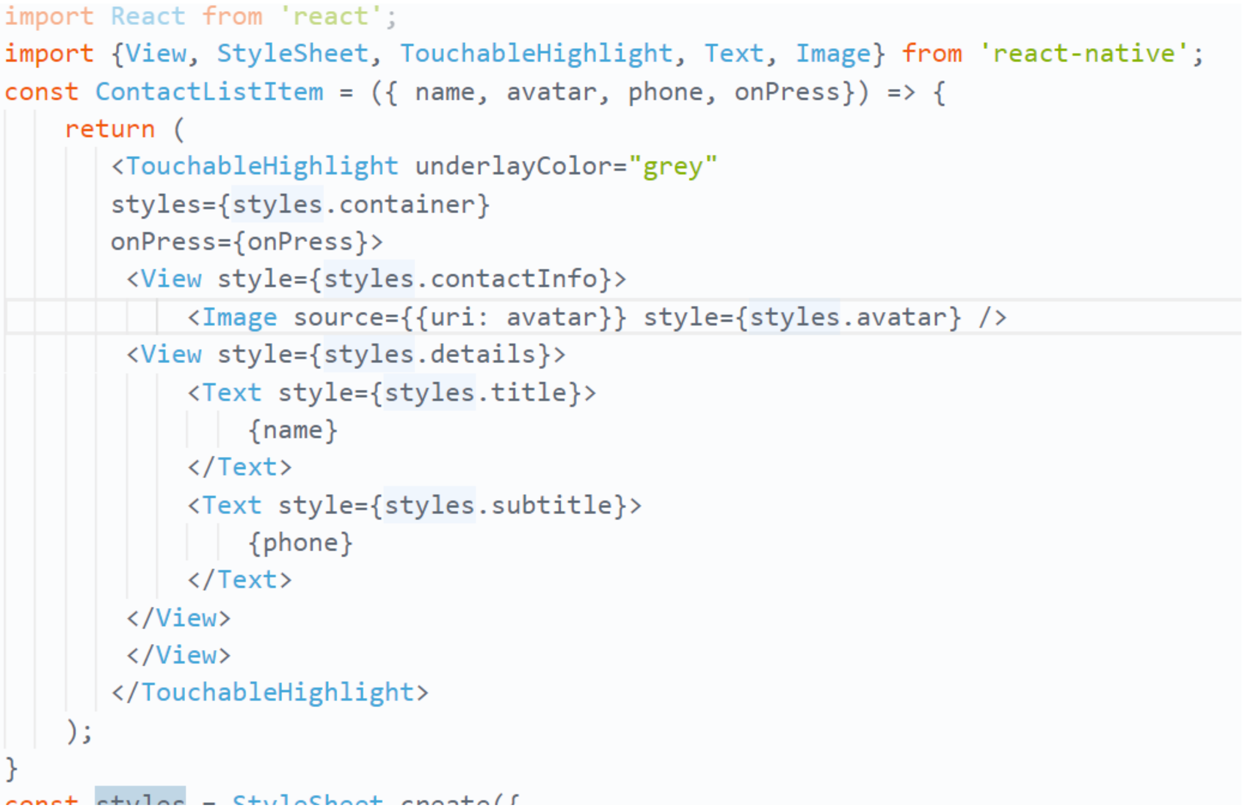


```
const keyExtractor = ({ phone }) => phone;

const fetchContacts = async () =>{
    const data = await fetch("https://randomuser.me/api/?results=50")
    const ContactData = await data.json();
    return ContactData.results.map(mapContacts);
};
const Contacts = ({ navigation }) => {
    const { contacts } = useSelector((state) => state);
    const dispatch = useDispatch();
    useEffect(() => {
        fetchContacts()
            .then(

                contacts => {

                    dispatch(fetchContactsSuccess(contacts));
                }
            )
            .catch(
                e => {

                }
            )

    },[])
```

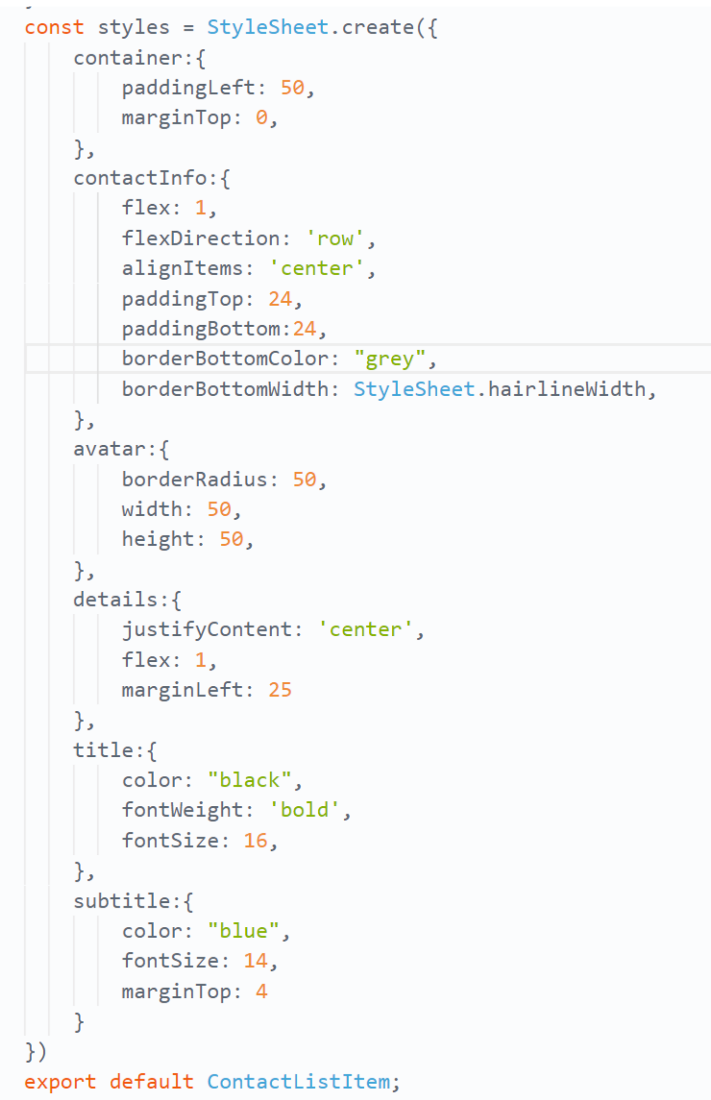


```
 // const contactsSorted = contacts.slice().sort((a, b) => a.name.localeCompare(b.name));
 const renderContacts = ({ item }) => {
     const { name, avatar, phone } = item;
     return <ContactListItem
         name={name}
         avatar={avatar}
         phone={phone}
         onPress={() => navigation.navigate("ProfileContact", { contact: item })}
     />;
 };
 return (
     <View style={styles.container}>

             <FlatList
                 data={contacts}
                 keyExtractor={keyExtractor}
                 renderItem={renderContacts}
             />
     </View >
 );


onst styles = StyleSheet.create({
  container: {
    justifyContent: 'center',
    flex: 1,
    paddingLeft: 10,
    paddingRight: 10,
```

File ContactListITem.js

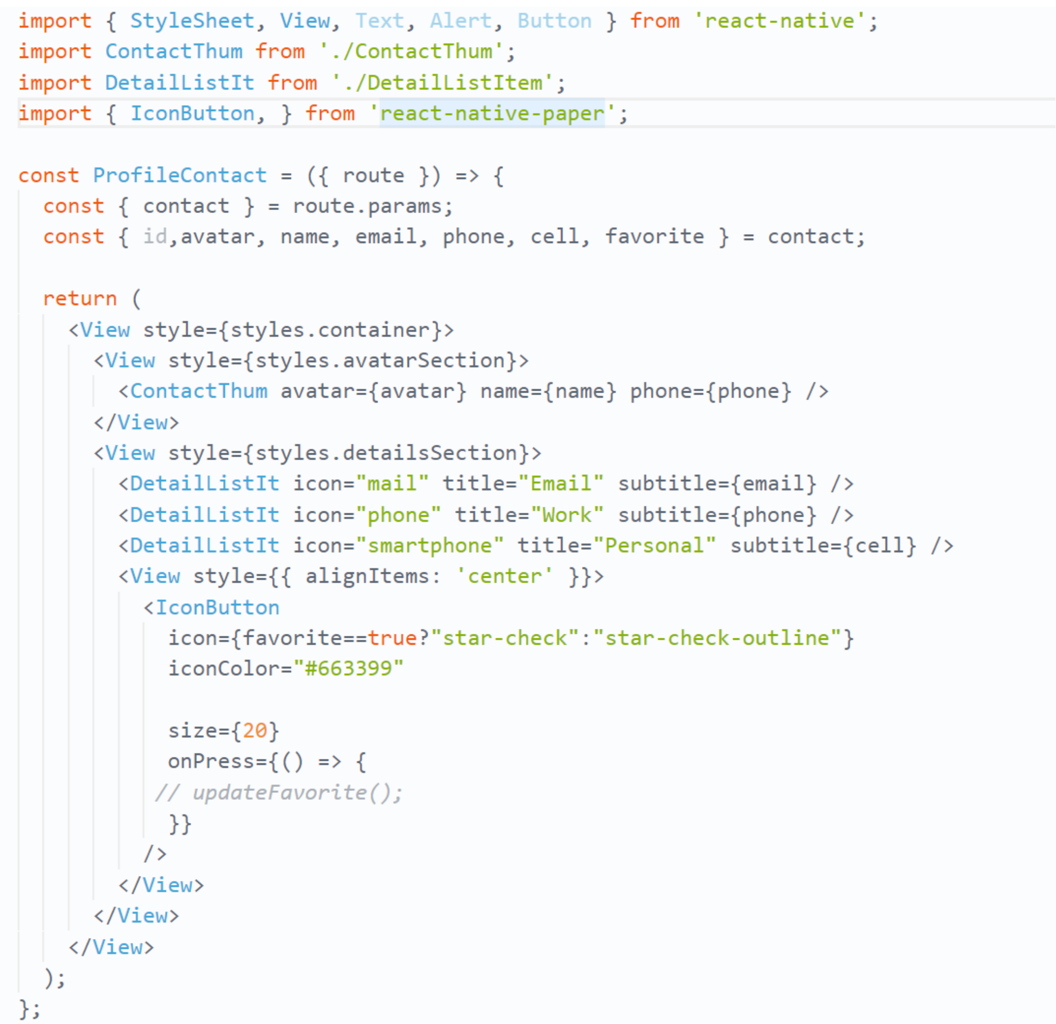


```
import React from 'react';
import {View, StyleSheet, TouchableHighlight, Text, Image} from 'react-native';
const ContactListItem = ({ name, avatar, phone, onPress}) => {
    return (
       <TouchableHighlight underlayColor="grey"
       styles={styles.container}
       onPress={onPress}>
        <View style={styles.contactInfo}>
            <Image source={{uri: avatar}} style={styles.avatar} />
        <View style={styles.details}>
            <Text style={styles.title}>
                {name}
            </Text>
            <Text style={styles.subtitle}>
                {phone}
            </Text>
        </View>
        </View>
       </TouchableHighlight>
    );
}
const styles = StyleSheet.create({
```

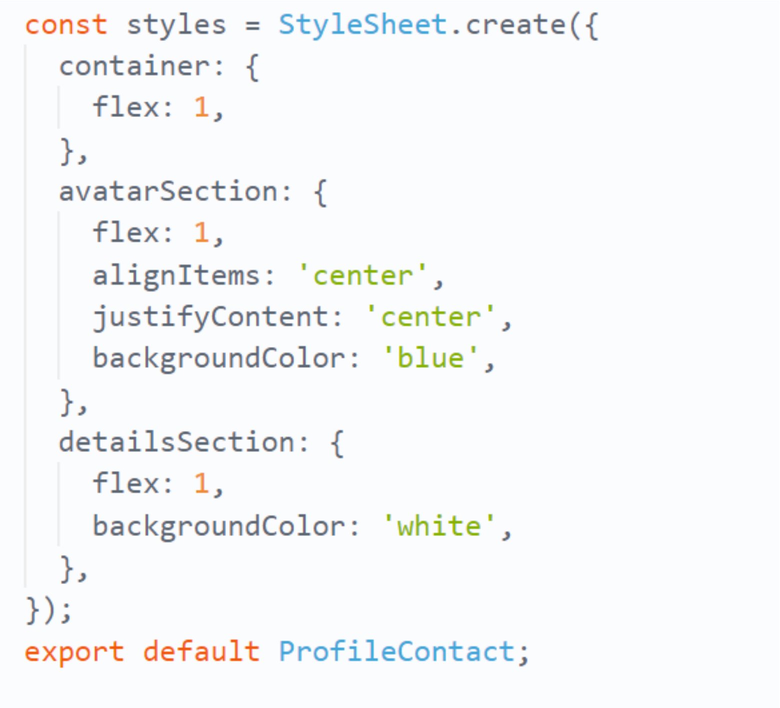


```
const styles = StyleSheet.create({
    container:{
        paddingLeft: 50,
        marginTop: 0,
    },
    contactInfo:{
        flex: 1,
        flexDirection: 'row',
        alignItems: 'center',
        paddingTop: 24,
        paddingBottom:24,
        borderBottomColor: "grey",
        borderBottomWidth: StyleSheet.hairlineWidth,
    },
    avatar:{
        borderRadius: 50,
        width: 50,
        height: 50,
    },
    details:{
        justifyContent: 'center',
        flex: 1,
        marginLeft: 25
    },
    title:{
        color: "black",
        fontWeight: 'bold',
        fontSize: 16,
    },
    subtitle:{
        color: "blue",
        fontSize: 14,
        marginTop: 4
    }
})
export default ContactListItem;
```

5. Design "Screen 2, 4" view details: File ProfileContact.js

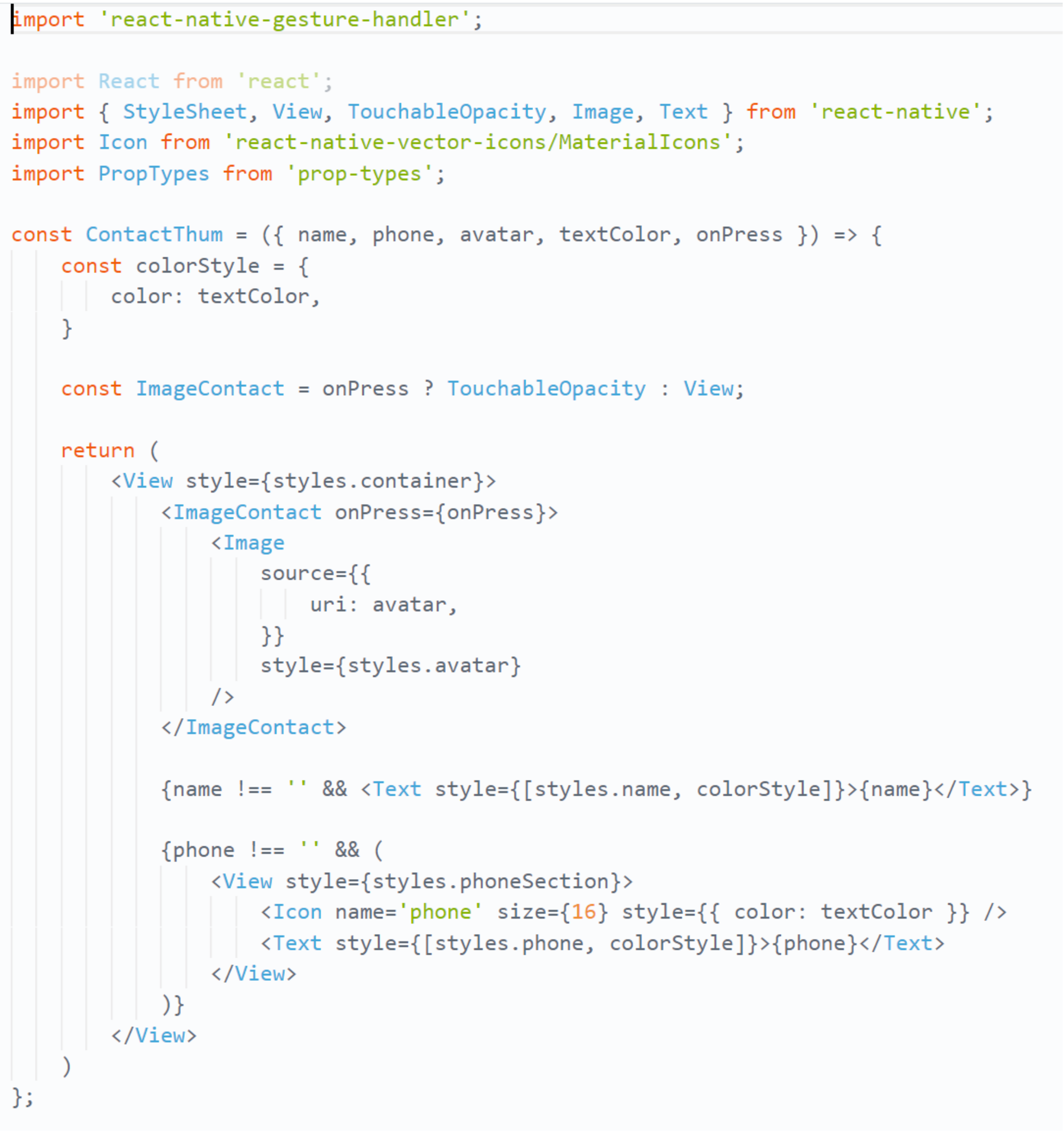


```
import { StyleSheet, View, Text, Alert, Button } from 'react-native';
import ContactThum from './ContactThum';
import DetailListIt from './DetailListItem';
import { IconButton, } from 'react-native-paper';

const ProfileContact = ({ route }) => {
  const { contact } = route.params;
  const { id,avatar, name, email, phone, cell, favorite } = contact;

  return (
    <View style={styles.container}>
      <View style={styles.avatarSection}>
        <ContactThum avatar={avatar} name={name} phone={phone} />
      </View>
      <View style={styles.detailsSection}>
        <DetailListIt icon="mail" title="Email" subtitle={email} />
        <DetailListIt icon="phone" title="Work" subtitle={phone} />
        <DetailListIt icon="smartphone" title="Personal" subtitle={cell} />
        <View style={{ alignItems: 'center' }}>
          <IconButton
            icon={favorite==true?"star-check":"star-check-outline"}
            iconColor="#663399"

            size={20}
            onPress={() => {
           // updateFavorite();
            }}
          />
        </View>
      </View>
    </View>
  );
};
```

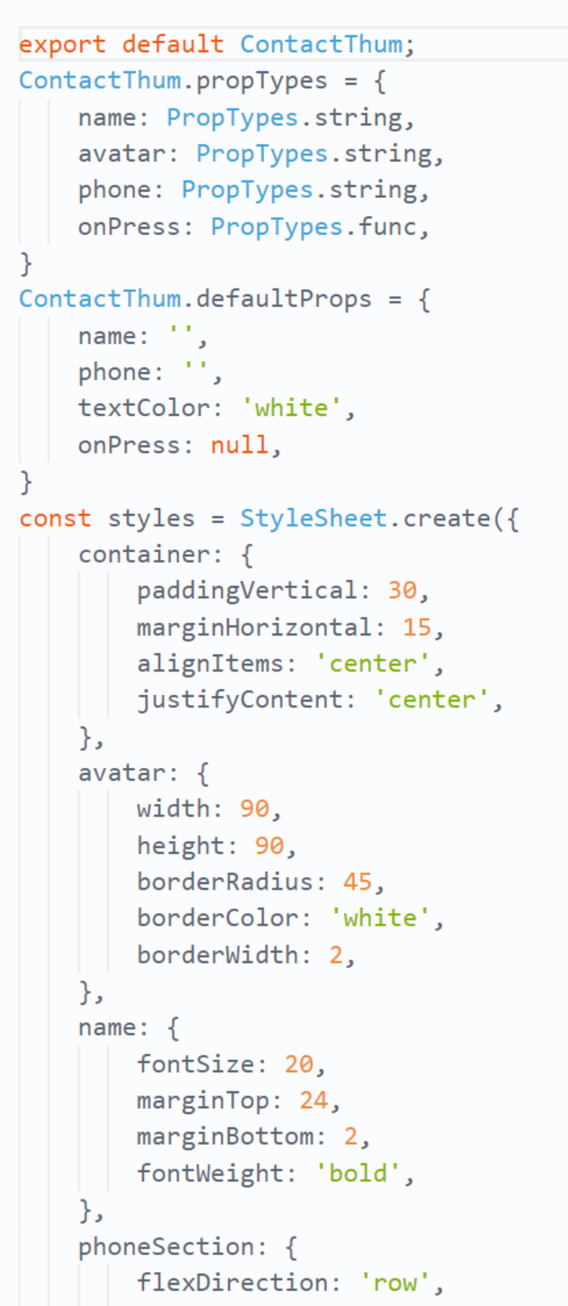


```
const styles = StyleSheet.create({
  container: {
    flex: 1,
  },
  avatarSection: {
    flex: 1,
    alignItems: 'center',
    justifyContent: 'center',
    backgroundColor: 'blue',
  },
  detailsSection: {
    flex: 1,
    backgroundColor: 'white',
  },
});
export default ProfileContact;
```

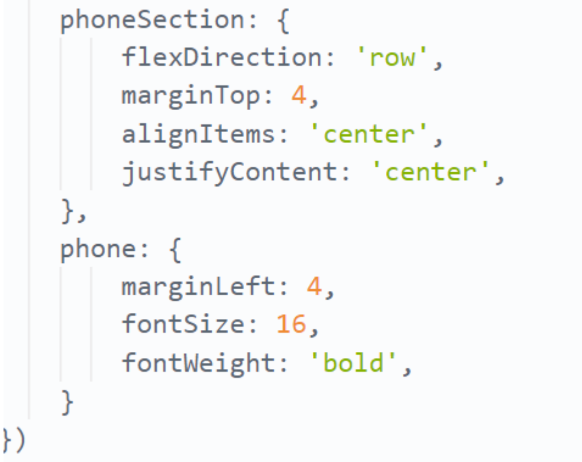
File ContactThum.js



```
import 'react-native-gesture-handler';

import React from 'react';
import { StyleSheet, View, TouchableOpacity, Image, Text } from 'react-native';
import Icon from 'react-native-vector-icons/MaterialIcons';
import PropTypes from 'prop-types';

const ContactThum = ({ name, phone, avatar, textColor, onPress }) => {
    const colorStyle = {
        color: textColor,
    }

    const ImageContact = onPress ? TouchableOpacity : View;

    return (
        <View style={styles.container}>
            <ImageContact onPress={onPress}>
                <Image
                    source={{
                        uri: avatar,
                    }}
                    style={styles.avatar}
                />
            </ImageContact>

            {name !== '' && <Text style={[styles.name, colorStyle]}>{name}</Text>}

            {phone !== '' && (
                <View style={styles.phoneSection}>
                    <Icon name='phone' size={16} style={{ color: textColor }} />
                    <Text style={[styles.phone, colorStyle]}>{phone}</Text>
                </View>
            )}
        </View>
    )
};
```

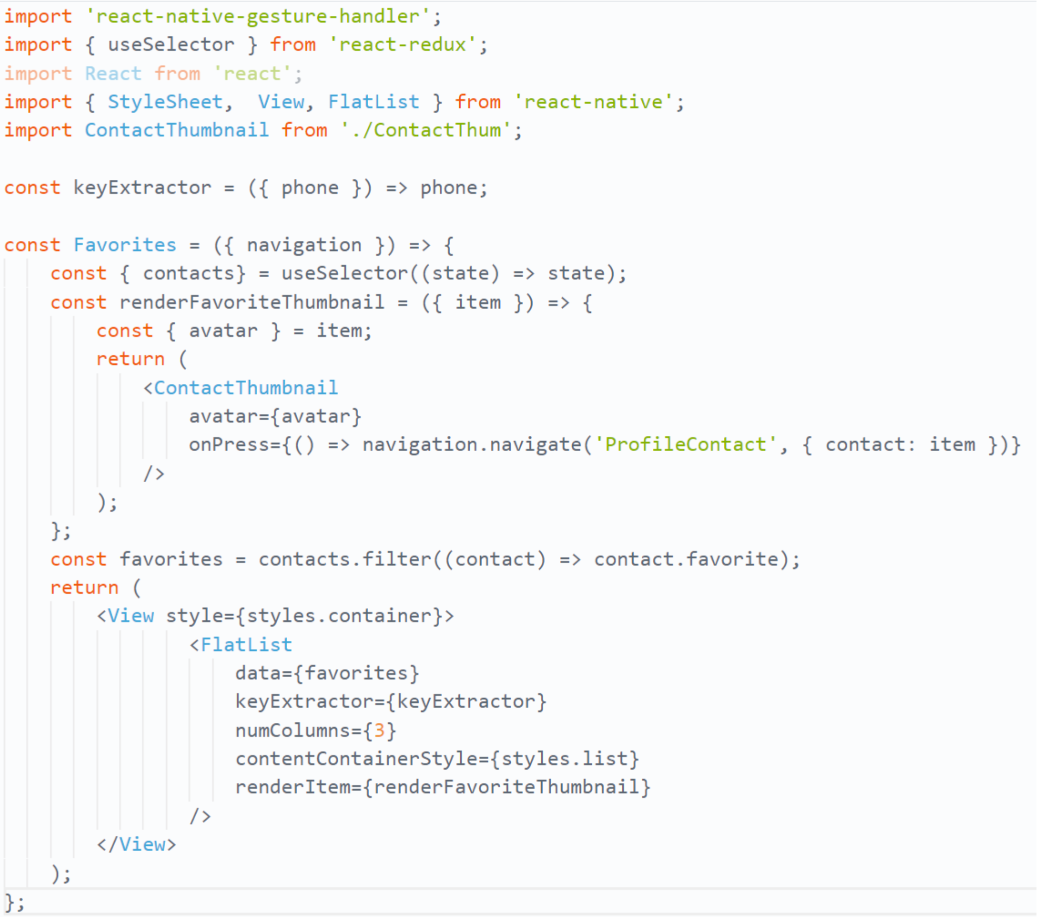


```
export default ContactThum;
ContactThum.propTypes = {
    name: PropTypes.string,
    avatar: PropTypes.string,
    phone: PropTypes.string,
    onPress: PropTypes.func,
}
ContactThum.defaultProps = {
    name: '',
    phone: '',
    textColor: 'white',
    onPress: null,
}
const styles = StyleSheet.create({
    container: {
        paddingVertical: 30,
        marginHorizontal: 15,
        alignItems: 'center',
        justifyContent: 'center',
    },
    avatar: {
        width: 90,
        height: 90,
        borderRadius: 45,
        borderColor: 'white',
        borderWidth: 2,
    },
    name: {
        fontSize: 20,
        marginTop: 24,
        marginBottom: 2,
        fontWeight: 'bold',
    },
    phoneSection: {
        flexDirection: 'row',
```

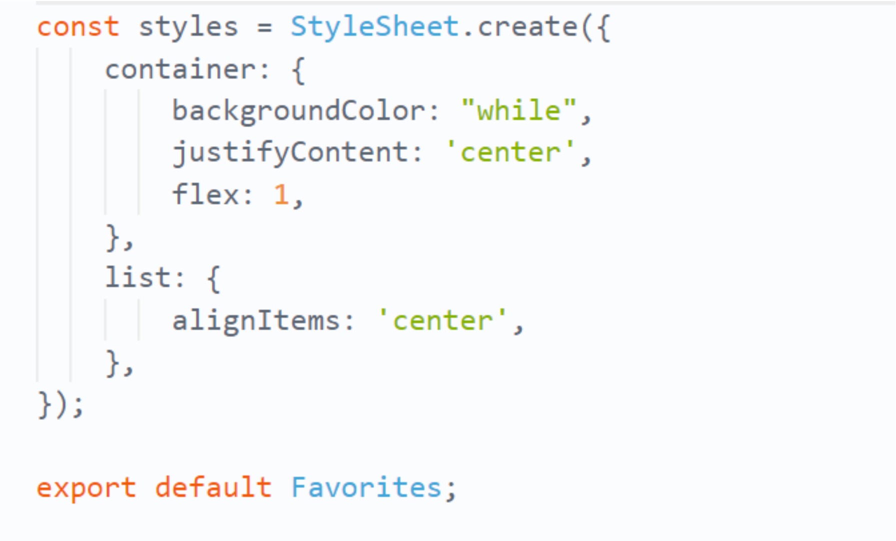


```
    phoneSection: {
        flexDirection: 'row',
        marginTop: 4,
        alignItems: 'center',
        justifyContent: 'center',
    },
    phone: {
        marginLeft: 4,
        fontSize: 16,
        fontWeight: 'bold',
    }
})
```

6. Design “Screen 3”: File Favorites.js

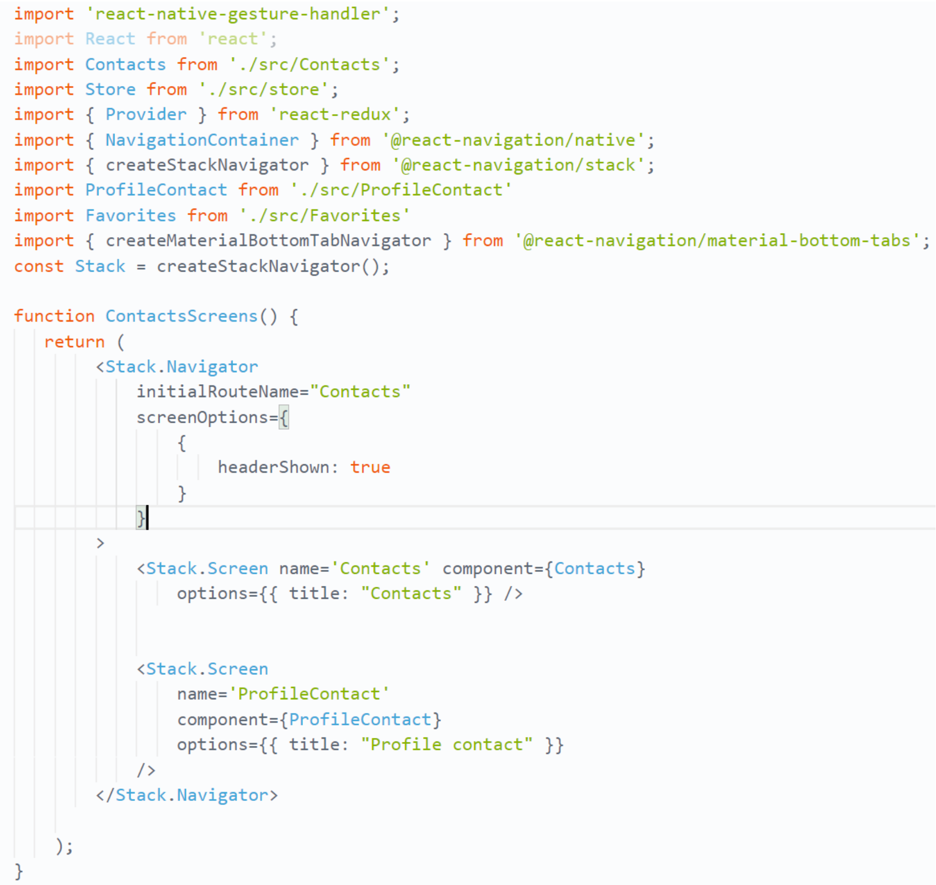


```
import 'react-native-gesture-handler';
import { useSelector } from 'react-redux';
import React from 'react';
import { StyleSheet,  View, FlatList } from 'react-native';
import ContactThumbnail from './ContactThum';

const keyExtractor = ({ phone }) => phone;

const Favorites = ({ navigation }) => {
    const { contacts} = useSelector((state) => state);
    const renderFavoriteThumbnail = ({ item }) => {
        const { avatar } = item;
        return (
            <ContactThumbnail
                avatar={avatar}
                onPress={() => navigation.navigate('ProfileContact', { contact: item })}
            />
        );
    };
    const favorites = contacts.filter((contact) => contact.favorite);
    return (
        <View style={styles.container}>
                <FlatList
                    data={favorites}
                    keyExtractor={keyExtractor}
                    numColumns={3}
                    contentContainerStyle={styles.list}
                    renderItem={renderFavoriteThumbnail}
                />
        </View>
    );
};
```

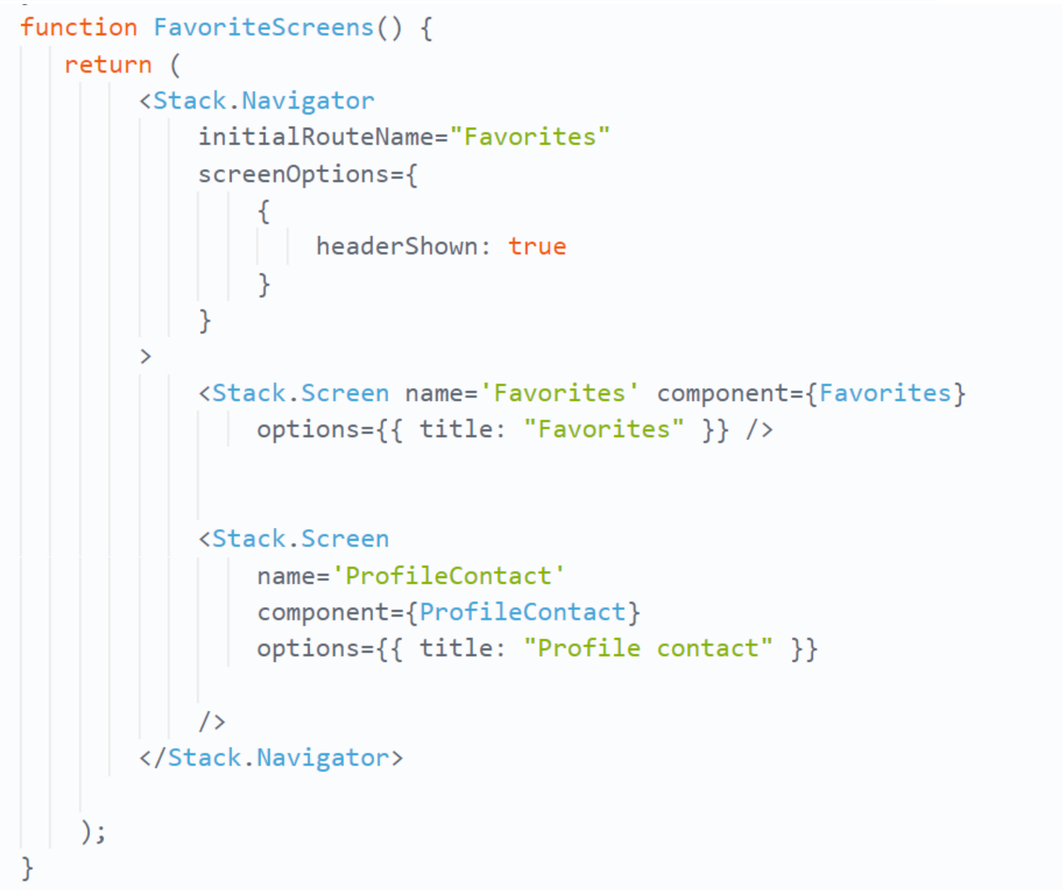


```
const styles = StyleSheet.create({
    container: {
        backgroundColor: "while",
        justifyContent: 'center',
        flex: 1,
    },
    list: {
        alignItems: 'center',
    },
});

export default Favorites;
```

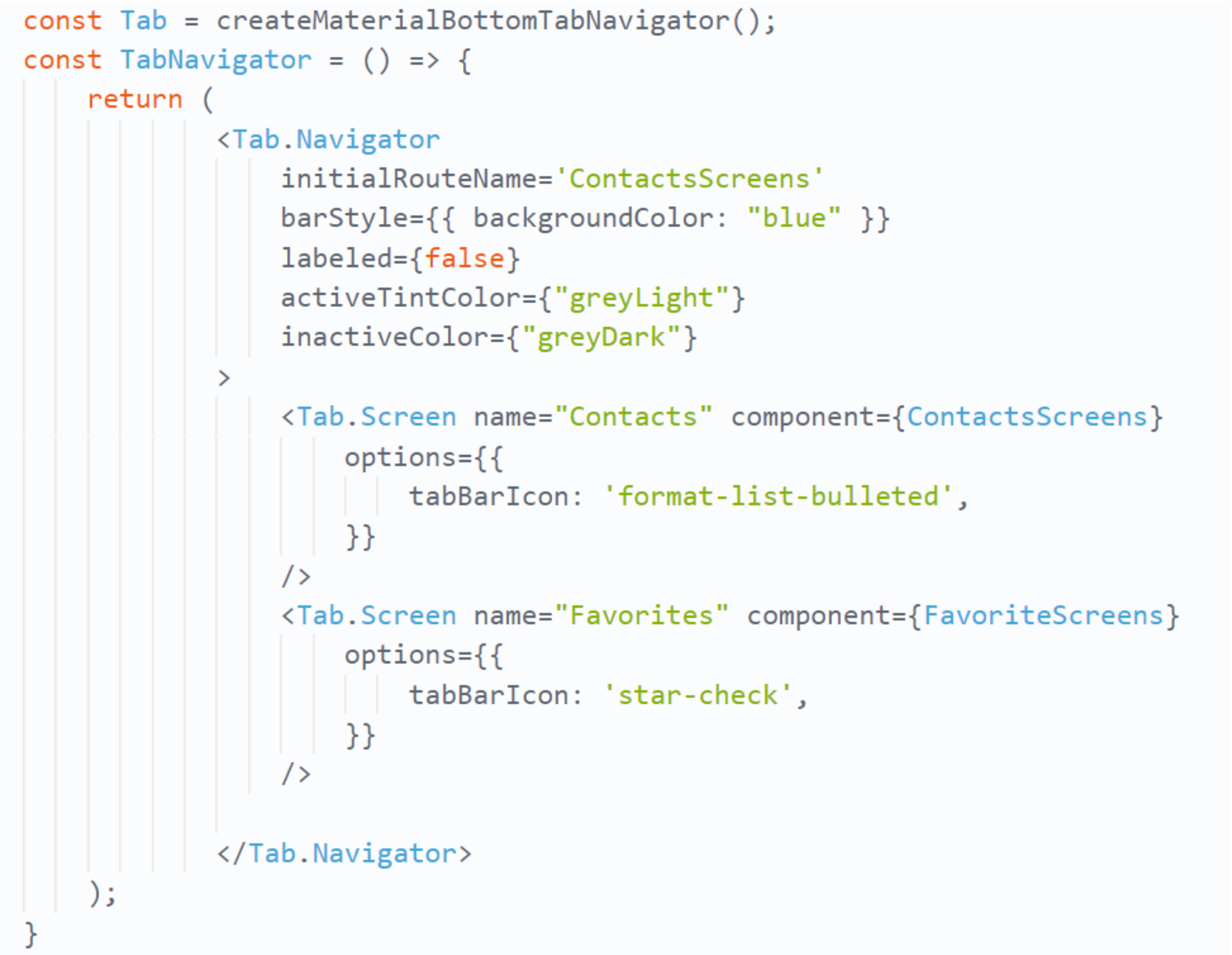
7. File App.js with Navigation.



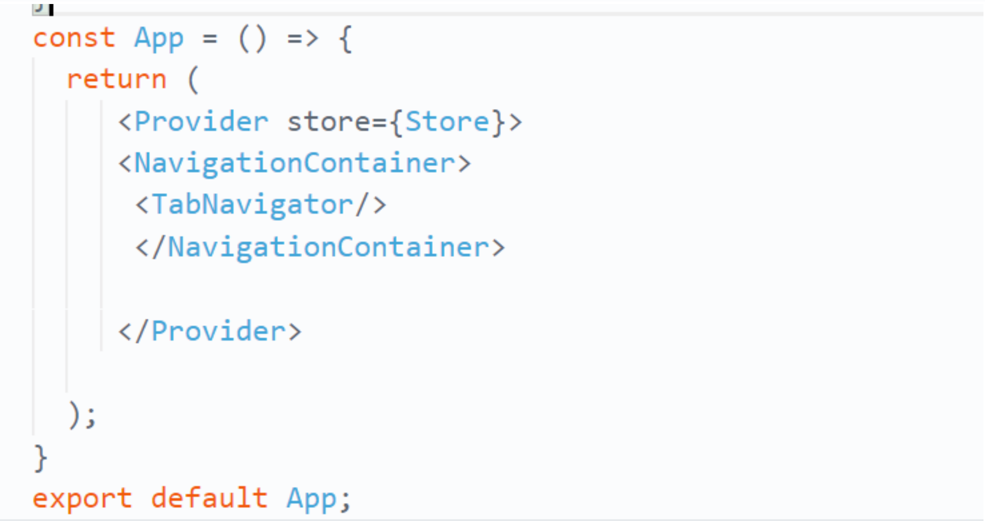
```
import 'react-native-gesture-handler';
import React from 'react';
import Contacts from './src/Contacts';
import Store from './src/store';
import { Provider } from 'react-redux';
import { NavigationContainer } from '@react-navigation/native';
import { createStackNavigator } from '@react-navigation/stack';
import ProfileContact from './src/ProfileContact'
import Favorites from './src/Favorites'
import { createMaterialBottomTabNavigator } from '@react-navigation/material-bottom-tabs';
const Stack = createStackNavigator();

function ContactsScreens() {
   return (
        <Stack.Navigator
            initialRouteName="Contacts"
            screenOptions={
                {
                    headerShown: true
                }
            }
        >
            <Stack.Screen name='Contacts' component={Contacts}
                options={{ title: "Contacts" }} />


            <Stack.Screen
                name='ProfileContact'
                component={ProfileContact}
                options={{ title: "Profile contact" }}
            />
        </Stack.Navigator>

    );
}
```



```
function FavoriteScreens() {
   return (
        <Stack.Navigator
            initialRouteName="Favorites"
            screenOptions={
                {
                    headerShown: true
                }
            }
        >
            <Stack.Screen name='Favorites' component={Favorites}
                options={{ title: "Favorites" }} />


            <Stack.Screen
                name='ProfileContact'
                component={ProfileContact}
                options={{ title: "Profile contact" }}

            />
        </Stack.Navigator>

    );
}
```



```
const Tab = createMaterialBottomTabNavigator();
const TabNavigator = () => {
    return (
            <Tab.Navigator
                initialRouteName='ContactsScreens'
                barStyle={{ backgroundColor: "blue" }}
                labeled={false}
                activeTintColor={"greyLight"}
                inactiveColor={"greyDark"}
            >
                <Tab.Screen name="Contacts" component={ContactsScreens}
                    options={{
                        tabBarIcon: 'format-list-bulleted',
                    }}
                />
                <Tab.Screen name="Favorites" component={FavoriteScreens}
                    options={{
                        tabBarIcon: 'star-check',
                    }}
                />

            </Tab.Navigator>
    );
}
```



```
const App = () => {
  return (
     <Provider store={Store}>
     <NavigationContainer>
      <TabNavigator/>
      </NavigationContainer>

     </Provider>

  );
}
export default App;
```

8. Replace Tab Navigation with Drawer Navigation and Redux toolkit with AsyncStorage.