

1. displaying items in a grid

a) create a screens folder and CategoriesScreen.js
And create a components folder and inside of it
CategoryGridTitle.js

b) In CategoriesScreen.js(to show the list of the items)

```
import { FlatList } from 'react-native';
import CategoryGridTile from '../components/CategoryGridTile';

import { CATEGORIES } from '../data/dummy-data';

function renderCategoryItem(itemData) {
  return (
    <CategoryGridTile title={itemData.item.title}
color={itemData.item.color} />
  );
}

function CategoriesScreen() {
  return (
    <FlatList
      data={CATEGORIES}
      keyExtractor={(item) => item.id}
      renderItem={renderCategoryItem}
      numColumns={2}
    />
  );
}

export default CategoriesScreen;
```

In CategoryGridTile.js(how to display the items)

```
import { Pressable, View, Text, StyleSheet, Platform } from
'react-native';

function CategoryGridTile({ title, color }) {
  return (
    <Pressable
      style={[Platform.OS === 'ios' ? styles.iosContainer : styles.androidContainer]}
      onPress={() => console.log('Category Grid Tile Pressed')}
    >
      <View style={styles.container}>
        <Text style={styles.title}>{title}</Text>
      </View>
    </Pressable>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    padding: 10,
    margin: 10,
    alignItems: 'center',
    justifyContent: 'center',
    backgroundColor: '#fff',
    shadowColor: '#000',
    shadowOffset: { width: 0, height: 2 },
    shadowRadius: 2,
    shadowOpacity: 0.2,
  },
  title: {
    color: 'black',
    fontSize: 18,
    fontWeight: 'bold',
  },
  iosContainer: {
    padding: 10,
  },
  androidContainer: {
    padding: 10,
  },
});
```

```
<View style={styles.gridItem}>
  <Pressable
    android_ripple={{ color: '#ccc' }}
    style={({ pressed }) => [
      styles.button,
      pressed ? styles.buttonPressed : null,
    ]}
  >
    <View style={[styles.innerContainer, { backgroundColor: color }]}>
      <Text style={styles.title}>{title}</Text>
    </View>
  </Pressable>
</View>
);

}

export default CategoryGridTile;

const styles = StyleSheet.create({
  gridItem: {
    flex: 1,
    margin: 16,
    height: 150,
    borderRadius: 8,
    elevation: 4,
    backgroundColor: 'white',
    shadowColor: 'black',
    shadowOpacity: 0.25,
    shadowOffset: { width: 0, height: 2 },
    shadowRadius: 8,
    overflow: Platform.OS === 'android' ? 'hidden' : 'visible',
  },
  button: {
    flex: 1,
  },
  buttonPressed: {
    opacity: 0.5,
  },
  innerContainer: {
    flex: 1,
```

```
padding: 16,
borderRadius: 8,
justifyContent: 'center',
alignItems: 'center',
},
title: {
fontWeight: 'bold',
fontSize: 18,
},
}) ;
```

In app.js

```
import { StatusBar } from 'expo-status-bar';
import { StyleSheet } from 'react-native';
import CategoriesScreen from './screens/CategoriesScreen';

export default function App() {
  return (
    <>
      <StatusBar style='light' />
      <CategoriesScreen />
    </>
  ) ;
}

const styles = StyleSheet.create({
  container: {},
});
```

In app.json

```
expo": {
  "name": "navigation",
  "slug": "navigation",
  "version": "1.0.0",
  "orientation": "portrait",
  "icon": "./assets/icon.png",
  "backgroundColor": "#24180f",
```

```
"userInterfaceStyle": "light",
"splash": {
  "image": "./assets/splash.png",
  "resizeMode": "contain",
  "backgroundColor": "#ffffff"
},
"up
```

Output:



2.React Navigation

a)install react navigvation

npm install @react-navigation/native

b)for expo users install

npx expo install react-native-screens react-native-safe-area-context

c)installing react-nativation stack

npm install @react-navigation/native-stack

npx expo install react-native-gesture-handler

d) In app.js

```
import { StatusBar } from 'expo-status-bar';
import { StyleSheet } from 'react-native';
import { NavigationContainer } from '@react-navigation/native';
import { createNativeStackNavigator } from
'@react-navigation/native-stack';

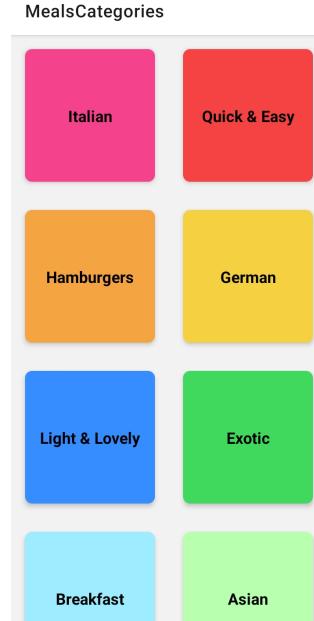
import CategoriesScreen from './screens/CategoriesScreen';

const Stack = createNativeStackNavigator();

export default function App() {
  return (
    <>
      <StatusBar style="dark" />
      <NavigationContainer>
        <Stack.Navigator>
          <Stack.Screen name="MealsCategories"
component={CategoriesScreen} />
        </Stack.Navigator>
      </NavigationContainer>
    </>
  ) ;
}

const styles = StyleSheet.create({
  container: {},
});
```

Output:



3.Implementing navigation between two screens

1)In MealsOverViewScreen.js(contains information after pressable button)

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

import { MEALS } from '../data/dummy-data';

function MealsOverviewScreen() {
  return (
    <View style={styles.container}>
      <Text>Meals Overview Screen</Text>
    </View>
  );
}

export default MealsOverviewScreen;

const styles = StyleSheet.create({
  container: {
    flex: 1,
```

```
        padding: 16,  
    },  
});
```

2)In categoriesscreen.js

```
import { FlatList } from 'react-native';  
import CategoryGridTile from '../components/CategoryGridTile';  
  
import { CATEGORIES } from '../data/dummy-data';  
  
function CategoriesScreen({ navigation }) {  
    function renderCategoryItem(itemData) {  
        function pressHandler() {  
            navigation.navigate('MealsOverview');  
        }  
  
        return (  
            <CategoryGridTile  
                title={itemData.item.title}  
                color={itemData.item.color}  
                onPress={pressHandler}  
            />  
        );  
    }  
  
    return (  
        <FlatList  
            data={CATEGORIES}  
            keyExtractor={(item) => item.id}  
            renderItem={renderCategoryItem}  
            numColumns={2}  
        />  
    );  
}  
  
export default CategoriesScreen;
```

3)in categorygridtitle.js

```
import { Pressable, View, Text, StyleSheet, Platform } from
'react-native';

function CategoryGridTile({ title, color, onPress }) {
  return (
    <View style={styles.gridItem}>
      <Pressable
        android_ripple={{ color: '#ccc' }}
        style={({ pressed }) => [
          styles.button,
          pressed ? styles.buttonPressed : null,
        ]}
        onPress={onPress}
      >
        <View style={[styles.innerContainer, { backgroundColor: color }]}>
          <Text style={styles.title}>{title}</Text>
        </View>
      </Pressable>
    </View>
  );
}

export default CategoryGridTile;

const styles = StyleSheet.create({
  gridItem: {
    flex: 1,
    margin: 16,
    height: 150,
    borderRadius: 8,
    elevation: 4,
    backgroundColor: 'white',
    shadowColor: 'black',
    shadowOpacity: 0.25,
    shadowOffset: { width: 0, height: 2 },
    shadowRadius: 8,
    overflow: Platform.OS === 'android' ? 'hidden' : 'visible',
  },
  button: {
    flex: 1,
```

```

} ,
buttonPressed: {
  opacity: 0.5,
},
innerContainer: {
  flex: 1,
  padding: 16,
  borderRadius: 8,
  justifyContent: 'center',
  alignItems: 'center',
},
title: {
  fontWeight: 'bold',
  fontSize: 18,
},
);
}

```

d)in app.js

```

import { StatusBar } from 'expo-status-bar';
import { StyleSheet } from 'react-native';
import { NavigationContainer } from '@react-navigation/native';
import { createNativeStackNavigator } from
'@react-navigation/native-stack';

import CategoriesScreen from './screens/CategoriesScreen';
import MealsOverviewScreen from './screens/MealsOverviewScreen';

const Stack = createNativeStackNavigator();

export default function App() {
  return (
    <>
      <StatusBar style="dark" />
      <NavigationContainer>
        <Stack.Navigator>
          <Stack.Screen name="MealsCategories"
component={CategoriesScreen} />
          <Stack.Screen name="MealsOverview"
component={MealsOverviewScreen} />

```

```

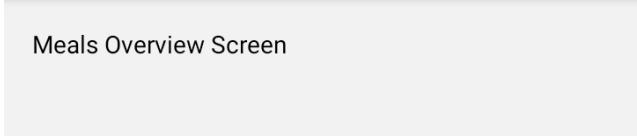
        </Stack.Navigator>
    </NavigationContainer>
</>
);
}

const styles = StyleSheet.create({
  container: {},
}) ;

```

Output:

← MealsOverview



Meals Overview Screen

3)Working with route parameters to pass data between screens

a)In categoriesScreen.js

```

import { FlatList } from 'react-native';
import CategoryGridTile from '../components/CategoryGridTile';

import { CATEGORIES } from '../data/dummy-data';

function CategoriesScreen({ navigation }) {
  function renderCategoryItem(itemData) {
    function pressHandler() {
      navigation.navigate('MealsOverview', {
        categoryId: itemData.item.id,
      });
    }

    return (
      <CategoryGridTile
        title={itemData.item.title}
    
```

```

        color={itemData.item.color}
        onPress={pressHandler}
      />
    ) ;
}

return (
<FlatList
  data={CATEGORIES}
  keyExtractor={(item) => item.id}
  renderItem={renderCategoryItem}
  numColumns={2}
/>
) ;
}

export default CategoriesScreen;

```

b)In MealsOverViewSCreen.js

```

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

import { MEALS } from '../data/dummy-data';

function MealsOverviewScreen({ route }) {
  const catId = route.params.categoryId;

  return (
    <View style={styles.container}>
      <Text>Meals Overview Screen - {catId}</Text>
    </View>
  ) ;
}

export default MealsOverviewScreen;

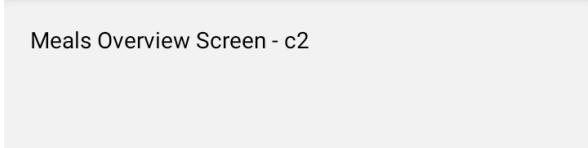
const styles = StyleSheet.create({
  container: {
    flex: 1,
    padding: 16,
  }
}

```

```
    } ,  
} );
```

Output:

← MealsOverview



Meals Overview Screen - c2

5) Displaying meals

In mealsoverviewscreen.js

```
import { View, FlatList, StyleSheet } from 'react-native';  
  
import MealItem from '../components/MealItem';  
import { MEALS } from '../data/dummy-data';  
  
function MealsOverviewScreen({ route }) {  
  const catId = route.params.categoryId;  
  
  const displayedMeals = MEALS.filter((mealItem) => {  
    return mealItem.categoryIds.indexOf(catId) >= 0;  
  });  
  
  function renderMealItem(itemData) {  
    const item = itemData.item;  
  
    const mealItemProps = {  
      title: item.title,  
      imageUrl: item.imageUrl,  
      affordability: item.affordability,  
      complexity: item.complexity,  
      duration: item.duration  
    };  
    return (  
      )  
  }  
  return (  
    )  
}
```

```

        <MealItem {...mealItemProps} />
    );
}

return (
    <View style={styles.container}>
        <FlatList
            data={displayedMeals}
            keyExtractor={(item) => item.id}
            renderItem={renderMealItem}
        />
    </View>
);
}

export default MealsOverviewScreen;

const styles = StyleSheet.create({
    container: {
        flex: 1,
        padding: 16,
    },
});

```

In mealitem.js(components folder)

```

import {
    View,
    Pressable,
    Text,
    Image,
    StyleSheet,
    Platform,
} from 'react-native';

function MealItem({ title, imageUrl, duration, complexity, affordability }) {
    return (
        <View style={styles.mealItem}>
            <Pressable

```

```
        android_ripple={{ color: '#ccc' }}
        style={({ pressed }) => (pressed ? styles.buttonPressed : null)}
    >
    <View style={styles.innerContainer}>
        <View>
            <Image source={{ uri: imageUrl }} style={styles.image} />
            <Text style={styles.title}>{title}</Text>
        </View>
        <View style={styles.details}>
            <Text style={styles.detailItem}>{duration}</Text>
            <Text
style={styles.detailItem}>{complexity.toUpperCase()}</Text>
            <Text
style={styles.detailItem}>{affordability.toUpperCase()}</Text>
        </View>
        </View>
    </Pressable>
</View>
);
}

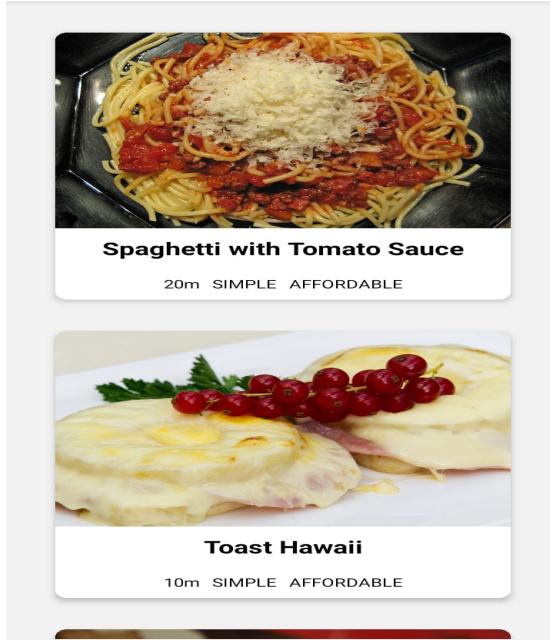
export default MealItem;

const styles = StyleSheet.create({
mealItem: {
    margin: 16,
    borderRadius: 8,
    overflow: Platform.OS === 'android' ? 'hidden' : 'visible',
    backgroundColor: 'white',
    elevation: 4,
    shadowColor: 'black',
    shadowOpacity: 0.25,
    shadowOffset: { width: 0, height: 2 },
    shadowRadius: 8,
},
buttonPressed: {
    opacity: 0.5,
},
innerContainer: {
    borderRadius: 8,
```

```
        overflow: 'hidden',
    },
    image: {
        width: '100%',
        height: 200,
    },
    title: {
        fontWeight: 'bold',
        textAlign: 'center',
        fontSize: 18,
        margin: 8,
    },
    details: {
        flexDirection: 'row',
        alignItems: 'center',
        justifyContent: 'center',
        padding: 8,
    },
    detailItem: {
        marginHorizontal: 4,
        fontSize: 12,
    },
}) ;
```

OUTPUT:

← MealsOverview



5) Styling Screen Headers & Backgrounds

In app.js

```
import { StatusBar } from 'expo-status-bar';
import { StyleSheet } from 'react-native';
import { NavigationContainer } from '@react-navigation/native';
import { createNativeStackNavigator } from
'@react-navigation/native-stack';

import CategoriesScreen from './screens/CategoriesScreen';
import MealsOverviewScreen from './screens/MealsOverviewScreen';

const Stack = createNativeStackNavigator();

export default function App() {
  return (
    <>
      <StatusBar style="dark" />
      <NavigationContainer>
        <Stack.Navigator
          screenOptions={{{
            headerStyle: { backgroundColor: '#351401' },

```

```

        headerTintColor: 'white',
        contentStyle: { backgroundColor: '#3f2f25' },
    } }
>
<Stack.Screen
    name="MealsCategories"
    component={CategoriesScreen}
    options={{
        title: 'All Categories',
    }}
/>
<Stack.Screen name="MealsOverview"
component={MealsOverviewScreen} />
</Stack.Navigator>
</NavigationContainer>
</>
);
}

const styles = StyleSheet.create({
    container: {},
}) ;

```

Output:



6) Styling navigation options dynamically

In mealsoverviewscreen.js

```
import { useLayoutEffect } from 'react';
import { View, FlatList, StyleSheet } from 'react-native';

import MealItem from '../components/MealItem';
import { MEALS, CATEGORIES } from '../data/dummy-data';

function MealsOverviewScreen({ route, navigation }) {
  const catId = route.params.categoryId;

  const displayedMeals = MEALS.filter((mealItem) => {
    return mealItem.categoryIds.indexOf(catId) >= 0;
  });

  useLayoutEffect(() => {
    const categoryTitle = CATEGORIES.find(
      (category) => category.id === catId
    ).title;

    navigation.setOptions({
      title: categoryTitle,
    });
  }, [catId, navigation]);

  function renderMealItem(itemData) {
    const item = itemData.item;

    const mealItemProps = {
      title: item.title,
      imageUrl: item.imageUrl,
      affordability: item.affordability,
      complexity: item.complexity,
      duration: item.duration,
    };
    return <MealItem {...mealItemProps} />;
  }

  return (
    <FlatList
      data={displayedMeals}
      keyExtractor={(item) => item.id}
      renderItem={renderMealItem}
    />
  );
}

export default MealsOverviewScreen;
```

```

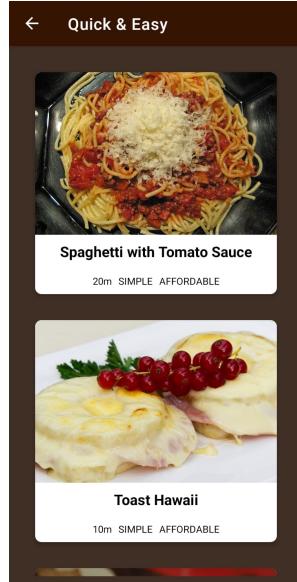
<View style={styles.container}>
  <FlatList
    data={displayedMeals}
    keyExtractor={(item) => item.id}
    renderItem={renderMealItem}
  />
</View>
);
}

export default MealsOverviewScreen;

const styles = StyleSheet.create({
  container: {
    flex: 1,
    padding: 16,
  },
});

```

Output:



8.Meal details screen

1)In List.js

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

```

function List({ data }) {
  return data.map((dataPoint) => (
    <View key={dataPoint} style={styles.listItem}>
      <Text style={styles.itemText}>{dataPoint}</Text>
    </View>
  )));
}

export default List;

const styles = StyleSheet.create({
  listItem: {
    borderRadius: 6,
    paddingHorizontal: 8,
    paddingVertical: 4,
    marginVertical: 4,
    marginHorizontal: 12,
    backgroundColor: '#e2b497',
  },
  itemText: {
    color: '#351401',
    textAlign: 'center',
  },
});

```

b) In subtitle.js

```

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

function Subtitle({children}) {
  return (
    <View style={styles.subtitleContainer}>
      <Text style={styles.subtitle}>{children}</Text>
    </View>
  );
}

export default Subtitle;

const styles = StyleSheet.create({

```

```

        subtitle: {
          color: '#e2b497',
          fontSize: 18,
          fontWeight: 'bold',
          textAlign: 'center',
        },
        subtitleContainer: {
          padding: 6,
          marginHorizontal: 12,
          marginVertical: 4,
          borderBottomColor: '#e2b497',
          borderBottomWidth: 2,
        },
      );
    }
  );
}

```

c)in MealDetailScreen.js

```

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

import List from '../components/MealDetail/List';
import Subtitle from '../components/MealDetail/Subtitle';
import MealDetails from '../components/MealDetails';
import { MEALS } from '../data/dummy-data';

function MealDetailScreen({ route }) {
  const mealId = route.params.mealId;

  const selectedMeal = MEALS.find((meal) => meal.id === mealId);

  return (
    <ScrollView style={styles.rootContainer}>
      <Image style={styles.image} source={{ uri: selectedMeal.imageUrl }} />
      <Text style={styles.title}>{selectedMeal.title}</Text>
      <MealDetails
        duration={selectedMeal.duration}
        complexity={selectedMeal.complexity}
        affordability={selectedMeal.affordability}
        textStyle={styles.detailText}
      />
      <View style={styles.listOuterContainer}>

```

```
        <View style={styles.listContainer}>
          <Subtitle>Ingredients</Subtitle>
          <List data={selectedMeal.ingredients} />
          <Subtitle>Steps</Subtitle>
          <List data={selectedMeal.steps} />
        </View>
      </View>
    </ScrollView>
  ) ;
}

export default MealDetailScreen;

const styles = StyleSheet.create({
  rootContainer: {
    marginBottom: 32
  },
  image: {
    width: '100%',
    height: 350,
  },
  title: {
    fontWeight: 'bold',
    fontSize: 24,
    margin: 8,
    textAlign: 'center',
    color: 'white',
  },
  detailText: {
    color: 'white',
  },
  listOuterContainer: {
    alignItems: 'center',
  },
  listContainer: {
    width: '80%',
  },
});

```

d)in mealdetails.js

```

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

function MealDetails({
  duration,
  complexity,
  affordability,
  style,
  textStyle,
}) {
  return (
    <View style={[styles.details, style]}>
      <Text style={[styles.detailItem, textStyle]}>{duration}m</Text>
      <Text style={[styles.detailItem, textStyle]}>
        {complexity.toUpperCase()}
      </Text>
      <Text style={[styles.detailItem, textStyle]}>
        {affordability.toUpperCase()}
      </Text>
    </View>
  );
}

export default MealDetails;

const styles = StyleSheet.create({
  details: {
    flexDirection: 'row',
    alignItems: 'center',
    justifyContent: 'center',
    padding: 8,
  },
  detailItem: {
    marginHorizontal: 4,
    fontSize: 12,
  },
});

```

d)in app.js

```

import { StatusBar } from 'expo-status-bar';
import { StyleSheet } from 'react-native';

```

```
import { NavigationContainer } from '@react-navigation/native';
import { createNativeStackNavigator } from
'@react-navigation/native-stack';

import CategoriesScreen from './screens/CategoriesScreen';
import MealsOverviewScreen from './screens/MealsOverviewScreen';
import MealDetailScreen from './screens/MealDetailScreen';

const Stack = createNativeStackNavigator();

export default function App() {
  return (
    <>
      <StatusBar style="light" />
      <NavigationContainer>
        <Stack.Navigator
          screenOptions={{
            headerStyle: { backgroundColor: '#351401' },
            headerTintColor: 'white',
            contentStyle: { backgroundColor: '#3f2f25' },
          }}>
          <Stack.Screen
            name="MealsCategories"
            component={CategoriesScreen}
            options={{
              title: 'All Categories',
            }}>
          </Stack.Screen>
          <Stack.Screen
            name="MealsOverview"
            component={MealsOverviewScreen}
            // options={({ route, navigation }) => {
            //   const catId = route.params.categoryId;
            //   return {
            //     title: catId,
            //   };
            // }}>
          </Stack.Screen>
          <Stack.Screen name="MealDetail" component={MealDetailScreen} />
        </Stack.Navigator>
      </NavigationContainer>
    </>
  );
}
```

```
        </Stack.Navigator>
    </NavigationContainer>
</>
);
}

const styles = StyleSheet.create({
  container: {},
}) ;
```

e)in mealitem.js

```
import {
  View,
  Pressable,
  Text,
  Image,
  StyleSheet,
  Platform,
} from 'react-native';
import { useNavigation } from '@react-navigation/native';

import MealDetails from './MealDetails';

function MealItem({
  id,
  title,
  imageUrl,
  duration,
  complexity,
  affordability,
}) {
  const navigation = useNavigation();

  function selectMealItemHandler() {
    navigation.navigate('MealDetail', {
      mealId: id,
    });
  }

  return (
    <Pressable
      style={styles.container}
      onPress={selectMealItemHandler}>
      <Image
        source={{uri: imageUrl}}
        style={styles.image} />
      <MealDetails
        title={title}
        duration={duration}
        complexity={complexity}
        affordability={affordability} />
    </Pressable>
  );
}
```

```
<View style={styles.mealItem}>
  <Pressable
    android_ripple={{ color: '#ccc' }}
    style={({ pressed }) => (pressed ? styles.buttonPressed : null)}
    onPress={selectMealItemHandler}
  >
    <View style={styles.innerContainer}>
      <View>
        <Image source={{ uri: imageUrl }} style={styles.image} />
        <Text style={styles.title}>{title}</Text>
      </View>
      <MealDetails
        duration={duration}
        affordability={affordability}
        complexity={complexity}
      />
    </View>
  </Pressable>
</View>
) ;
}

export default MealItem;

const styles = StyleSheet.create({
  mealItem: {
    margin: 16,
    borderRadius: 8,
    overflow: Platform.OS === 'android' ? 'hidden' : 'visible',
    backgroundColor: 'white',
    elevation: 4,
    shadowColor: 'black',
    shadowOpacity: 0.25,
    shadowOffset: { width: 0, height: 2 },
    shadowRadius: 8,
  },
  buttonPressed: {
    opacity: 0.5,
  },
  innerContainer: {
```

```

        borderRadius: 8,
        overflow: 'hidden',
    },
    image: {
        width: '100%',
        height: 200,
    },
    title: {
        fontWeight: 'bold',
        textAlign: 'center',
        fontSize: 18,
        margin: 8,
    },
}) ;

```

e)in mealoverviewscreen.js

```

import { useLayoutEffect } from 'react';
import { View, FlatList, StyleSheet } from 'react-native';

import MealItem from '../components/MealItem';
import { MEALS, CATEGORIES } from '../data/dummy-data';

function MealsOverviewScreen({ route, navigation }) {
    const catId = route.params.categoryId;

    const displayedMeals = MEALS.filter((mealItem) => {
        return mealItem.categoryIds.indexOf(catId) >= 0;
    });

    useLayoutEffect(() => {
        const categoryTitle = CATEGORIES.find(
            (category) => category.id === catId
        ).title;

        navigation.setOptions({
            title: categoryTitle,
        });
    }, [catId, navigation]);

    function renderMealItem(itemData) {

```

```
const item = itemData.item;

const mealItemProps = {
  id: item.id,
  title: item.title,
  imageUrl: item.imageUrl,
  affordability: item.affordability,
  complexity: item.complexity,
  duration: item.duration,
};

return <MealItem {...mealItemProps} />;
}

return (
  <View style={styles.container}>
    <FlatList
      data={displayedMeals}
      keyExtractor={(item) => item.id}
      renderItem={renderMealItem}
    />
  </View>
);
}

export default MealsOverviewScreen;

const styles = StyleSheet.create({
  container: {
    flex: 1,
    padding: 16,
  },
});
```

Output:



9)Adding Header Buttons In app.js

```
import { StatusBar } from 'expo-status-bar';
import { StyleSheet, Button } from 'react-native';
import { NavigationContainer } from '@react-navigation/native';
import { createStackNavigator } from '@react-navigation/native-stack';

import CategoriesScreen from './screens/CategoriesScreen';
import MealsOverviewScreen from './screens/MealsOverviewScreen';
import MealDetailScreen from './screens/MealDetailScreen';

const Stack = createStackNavigator();

export default function App() {
  return (
    <>
      <StatusBar style="light" />
      <NavigationContainer>
        <Stack.Navigator
          screenOptions={{{
            headerStyle: { backgroundColor: '#351401' },
            headerTintColor: 'white',
            contentStyle: { backgroundColor: '#3f2f25' },
          }}>
```

```

        >
      <Stack.Screen
        name="MealsCategories"
        component={CategoriesScreen}
        options={{
          title: 'All Categories',
        }}
      />
      <Stack.Screen
        name="MealsOverview"
        component={MealsOverviewScreen}
        // options={({ route, navigation }) => {
        //   const catId = route.params.categoryId;
        //   return {
        //     title: catId,
        //   };
        // }}
      />
      <Stack.Screen name="MealDetail" component={MealDetailScreen} />
    </Stack.Navigator>
  </NavigationContainer>
</>
);
}

const styles = StyleSheet.create({
  container: {},
});

```

2)in mealdetailscreen.js

```

import { useEffect } from 'react';
import { View, Text, Image, StyleSheet, ScrollView, Button } from
'react-native';

import List from '../components/MealDetail/List';
import Subtitle from '../components/MealDetail/Subtitle';
import MealDetails from '../components/MealDetails';

```

```
import { MEALS } from '../data/dummy-data';

function MealDetailScreen({ route, navigation }) {
  const mealId = route.params.mealId;

  const selectedMeal = MEALS.find((meal) => meal.id === mealId);

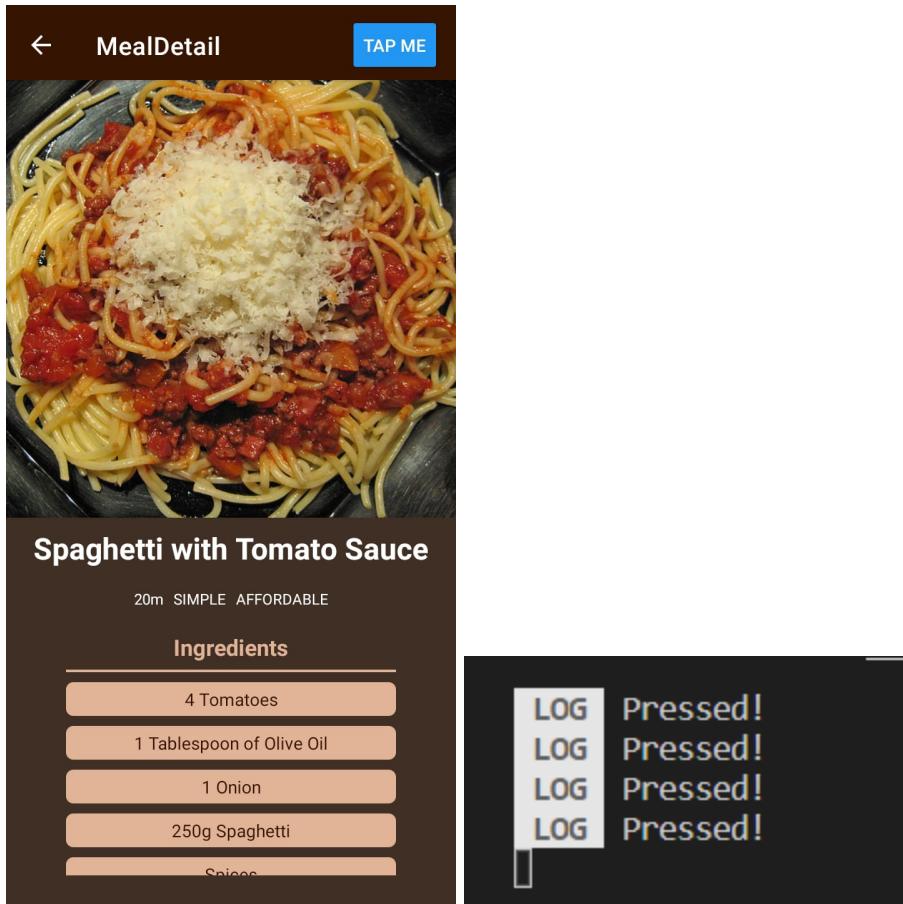
  function headerButtonPressHandler() {
    console.log('Pressed!');
  }

  useLayoutEffect(() => {
    navigation.setOptions({
      headerRight: () => {
        return <Button title='Tap me' onPress={headerButtonPressHandler}>
      />
        }
      });
    }, [navigation, headerButtonPressHandler]);

  return (
    <ScrollView style={styles.rootContainer}>
      <Image style={styles.image} source={{ uri: selectedMeal.imageUrl }} />
      <Text style={styles.title}>{selectedMeal.title}</Text>
      <MealDetails
        duration={selectedMeal.duration}
        complexity={selectedMeal.complexity}
        affordability={selectedMeal.affordability}
        textStyle={styles.detailText}
      />
      <View style={styles.listOuterContainer}>
        <View style={styles.listContainer}>
          <Subtitle>Ingredients</Subtitle>
          <List data={selectedMeal.ingredients} />
          <Subtitle>Steps</Subtitle>
          <List data={selectedMeal.steps} />
        </View>
      </View>
    </ScrollView>
  );
}
```

```
) ;  
}  
  
export default MealDetailScreen;  
  
const styles = StyleSheet.create({  
  rootContainer: {  
    marginBottom: 32  
  },  
  image: {  
    width: '100%',  
    height: 350,  
  },  
  title: {  
    fontWeight: 'bold',  
    fontSize: 24,  
    margin: 8,  
    textAlign: 'center',  
    color: 'white',  
  },  
  detailText: {  
    color: 'white',  
  },  
  listOuterContainer: {  
    alignItems: 'center',  
  },  
  listContainer: {  
    width: '80%',  
  },  
});
```

Output:



10)adding an icon button

In iconbutton.js

```
import { Pressable, StyleSheet } from 'react-native';
import { Ionicons } from '@expo/vector-icons';

function IconButton({ icon, color, onPress }) {
  return (
    <Pressable
      onPress={onPress}
      style={({ pressed }) => pressed && styles.pressed}
    >
      <Ionicons name={icon} size={24} color={color} />
    </Pressable>
  );
}
```

```
export default IconButton;

const styles = StyleSheet.create({
  pressed: {
    opacity: 0.7,
  },
});
```

In mealdetailscreen.js

```
import { useLayoutEffect } from 'react';
import { View, Text, Image, StyleSheet, ScrollView } from 'react-native';

import IconButton from '../components/IconButton';
import List from '../components/MealDetail/List';
import Subtitle from '../components/MealDetail/Subtitle';
import MealDetails from '../components/MealDetails';
import { MEALS } from '../data/dummy-data';

function MealDetailScreen({ route, navigation }) {
  const mealId = route.params.mealId;

  const selectedMeal = MEALS.find((meal) => meal.id === mealId);

  function headerButtonPressHandler() {
    console.log('Pressed!');
  }

  useLayoutEffect(() => {
    navigation.setOptions({
      headerRight: () => {
        return (
          <IconButton
            icon="star"
            color="white"
            onPress={headerButtonPressHandler}
          />
        );
      },
    });
  });
}
```

```
, [navigation, headerButtonPressHandler]) ;
```

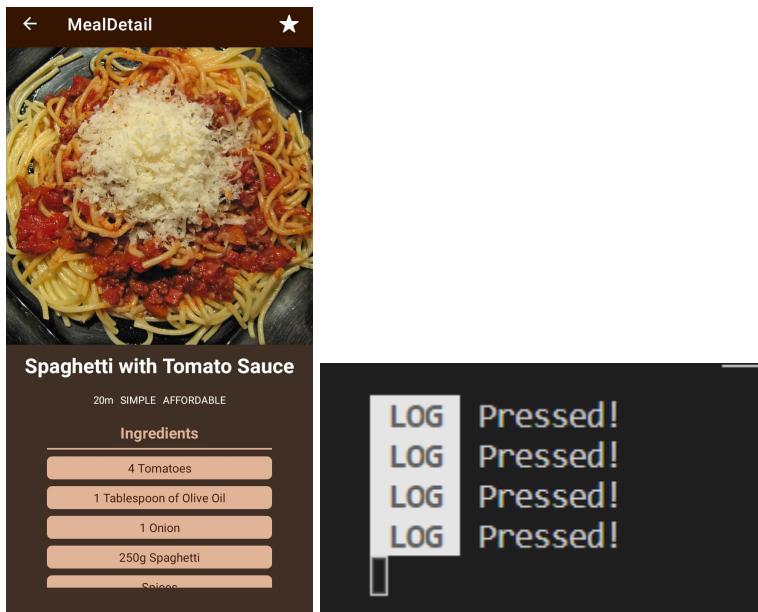
```
return (
  <ScrollView style={styles.rootContainer}>
    <Image style={styles.image} source={{ uri: selectedMeal.imageUrl }} />
    <Text style={styles.title}>{selectedMeal.title}</Text>
    <MealDetails
      duration={selectedMeal.duration}
      complexity={selectedMeal.complexity}
      affordability={selectedMeal.affordability}
      textStyle={styles.detailText}
    />
    <View style={styles.listOuterContainer}>
      <View style={styles.listContainer}>
        <Subtitle>Ingredients</Subtitle>
        <List data={selectedMeal.ingredients} />
        <Subtitle>Steps</Subtitle>
        <List data={selectedMeal.steps} />
      </View>
    </View>
  </ScrollView>
);
}
```

```
export default MealDetailScreen;
```

```
const styles = StyleSheet.create({
  rootContainer: {
    marginBottom: 32,
  },
  image: {
    width: '100%',
    height: 350,
  },
  title: {
    fontWeight: 'bold',
    fontSize: 24,
    margin: 8,
    textAlign: 'center',
  },
});
```

```
        color: 'white',
    },
    detailText: {
        color: 'white',
    },
    listOuterContainer: {
        alignItems: 'center',
    },
    listContainer: {
        width: '80%',
    },
},
});
```

Output:



11)nesting navigators

a)install reanimated and drawer-navigation

```
expo install react-native-reanimated
```

```
b) npm i @react-navigation/drawer
```

c)in app.js

```
import { StatusBar } from 'expo-status-bar';
import { StyleSheet, Button } from 'react-native';
import { NavigationContainer } from '@react-navigation/native';
import { createNativeStackNavigator } from
'@react-navigation/native-stack';
import { createDrawerNavigator } from '@react-navigation/drawer';

import CategoriesScreen from './screens/CategoriesScreen';
import MealsOverviewScreen from './screens/MealsOverviewScreen';
import MealDetailScreen from './screens/MealDetailScreen';
import FavoritesScreen from './screens/FavoritesScreen';

const Stack = createNativeStackNavigator();
const Drawer = createDrawerNavigator();

function DrawerNavigator() {
  return (
    <Drawer.Navigator
      screenOptions={{
        headerStyle: { backgroundColor: '#351401' },
        headerTintColor: 'white',
        sceneContainerStyle: { backgroundColor: '#3f2f25' },
      } {}
    >
      <Drawer.Screen
        name="Categories"
        component={CategoriesScreen}
        options={{ title: 'All Categories' }}
      />
      <Drawer.Screen name="Favorites" component={FavoritesScreen} />
    </Drawer.Navigator>
  );
}

export default function App() {
  return (
    <>
      <StatusBar style="light" />
    </>
  );
}
```

```

<NavigationContainer>
  <Stack.Navigator
    screenOptions={(
      headerStyle: { backgroundColor: '#351401' },
      headerTintColor: 'white',
      contentStyle: { backgroundColor: '#3f2f25' },
    )}
  >
    <Stack.Screen
      name="Drawer"
      component={DrawerNavigator}
      options={({
        headerShown: false,
      })}>
    />
    <Stack.Screen name="MealsOverview"
      component={MealsOverviewScreen} />
    <Stack.Screen
      name="MealDetail"
      component={MealDetailScreen}
      options={({
        title: 'About the Meal',
      })}>
    />
  </Stack.Navigator>
</NavigationContainer>
</>
) ;
}

const styles = StyleSheet.create({
  container: {},
}) ;

```

In favoritescreen.js

```

import { Text } from 'react-native';

function FavoritesScreen() {
  return <Text>The favorites screen!</Text>;

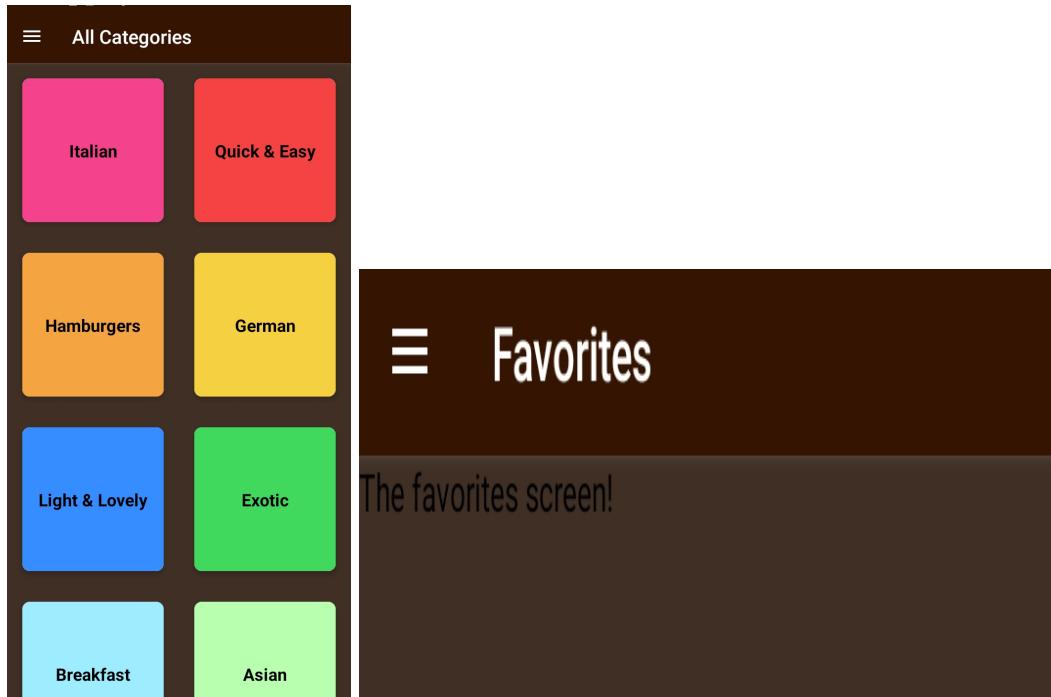
```

```
}
```



```
export default FavoritesScreen;
```

Output:



16)adding bottom tabs

```
import { StatusBar } from 'expo-status-bar';
import { StyleSheet, Button } from 'react-native';
import { NavigationContainer } from '@react-navigation/native';
import { createStackNavigator } from
'@react-navigation/native-stack';
import { createDrawerNavigator } from '@react-navigation/drawer';
import { Ionicons } from '@expo/vector-icons';

import CategoriesScreen from './screens/CategoriesScreen';
import MealsOverviewScreen from './screens/MealsOverviewScreen';
import MealDetailScreen from './screens/MealDetailScreen';
import FavoritesScreen from './screens/FavoritesScreen';
```

```
const Stack = createNativeStackNavigator();
const Drawer = createDrawerNavigator();

function DrawerNavigator() {
  return (
    <Drawer.Navigator
      screenOptions={{
        headerStyle: { backgroundColor: '#351401' },
        headerTintColor: 'white',
        sceneContainerStyle: { backgroundColor: '#3f2f25' },
        drawerContentStyle: { backgroundColor: '#351401' },
        drawerInactiveTintColor: 'white',
        drawerActiveTintColor: '#351401',
        drawerActiveBackgroundColor: '#e4baa1',
      }}>
      <Drawer.Screen
        name="Categories"
        component={CategoriesScreen}
        options={{ title: 'All Categories',
          drawerIcon: ({ color, size }) => (
            <Ionicons name="list" color={color} size={size} />
          ),
        }}>
      />
      <Drawer.Screen
        name="Favorites"
        component={FavoritesScreen}
        options={{ drawerIcon: ({ color, size }) => (
          <Ionicons name="star" color={color} size={size} />
        ),
        }}>
      />
    </Drawer.Navigator>
  );
}

export default function App() {
```

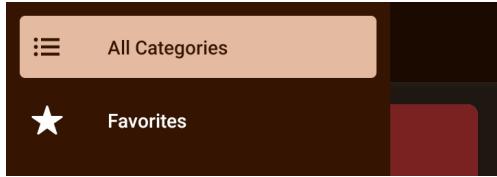
```

    return (
      <>
        <StatusBar style="light" />
        <NavigationContainer>
          <Stack.Navigator
            screenOptions={ {
              headerStyle: { backgroundColor: '#351401' },
              headerTintColor: 'white',
              contentStyle: { backgroundColor: '#3f2f25' },
            } }
          >
            <Stack.Screen
              name="Drawer"
              component={DrawerNavigator}
              options={ {
                headerShown: false,
              } }
            />
            <Stack.Screen name="MealsOverview"
              component={MealsOverviewScreen} />
            <Stack.Screen
              name="MealDetail"
              component={MealDetailScreen}
              options={ {
                title: 'About the Meal',
              } }
            />
          </Stack.Navigator>
        </NavigationContainer>
      </>
    ) ;
  }

  const styles = StyleSheet.create({
    container: {},
  });

```

Output:



17) context-api

React Context is a method to pass props from parent to child component(s), by storing the props in a store(similar in Redux) and using these props from the store by child component(s) without actually passing them manually at each level of the component tree.

1) create a store folder and inside of it create a context folder and inside of it create a

In favorites-context.js

```
import { createContext } from 'react';

export const FavoritesContext = createContext({
  ids: [],
  addFavorite: (id) => {},
  removeFavorite: (id) => {}
});

function FavoritesContextProvider({children}) {
  return <FavoritesContext.Provider>{children}</FavoritesContext.Provider>
}

export default FavoritesContextProvider;
```

In app.js

```
import { StatusBar } from 'expo-status-bar';
import { StyleSheet, Button } from 'react-native';
import { NavigationContainer } from '@react-navigation/native';
import { createNativeStackNavigator } from
  '@react-navigation/native-stack';
import { createDrawerNavigator } from '@react-navigation/drawer';
import { Ionicons } from '@expo/vector-icons';

import CategoriesScreen from './screens/CategoriesScreen';
```

```
import MealsOverviewScreen from './screens/MealsOverviewScreen';
import MealDetailScreen from './screens/MealDetailScreen';
import FavoritesScreen from './screens/FavoritesScreen';
import FavoritesContextProvider from './store/context/favorites-context';

const Stack = createNativeStackNavigator();
const Drawer = createDrawerNavigator();

function DrawerNavigator() {
  return (
    <Drawer.Navigator
      screenOptions={{
        headerStyle: { backgroundColor: '#351401' },
        headerTintColor: 'white',
        sceneContainerStyle: { backgroundColor: '#3f2f25' },
        drawerContentStyle: { backgroundColor: '#351401' },
        drawerInactiveTintColor: 'white',
        drawerActiveTintColor: '#351401',
        drawerActiveBackgroundColor: '#e4baa1',
      }}
    >
      <Drawer.Screen
        name="Categories"
        component={CategoriesScreen}
        options={{ title: 'All Categories',
          drawerIcon: ({ color, size }) => (
            <Ionicons name="list" color={color} size={size} />
          ),
        }}
      />
      <Drawer.Screen
        name="Favorites"
        component={FavoritesScreen}
        options={{ drawerIcon: ({ color, size }) => (
          <Ionicons name="star" color={color} size={size} />
        ),
      }}
    />
  
```

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

export default function App() {
    return (
        <>
            <StatusBar style="light" />
            <FavoritesContextProvider>
                <NavigationContainer>
                    <Stack.Navigator
                        screenOptions={{
                            headerStyle: { backgroundColor: '#351401' },
                            headerTintColor: 'white',
                            contentStyle: { backgroundColor: '#3f2f25' },
                        }}>
                        <Stack.Screen
                            name="Drawer"
                            component={DrawerNavigator}
                            options={{{
                                headerShown: false,
                            }}}>
                        />
                        <Stack.Screen
                            name="MealsOverview"
                            component={MealsOverviewScreen}>
                        />
                        <Stack.Screen
                            name="MealDetail"
                            component={MealDetailScreen}
                            options={{{
                                title: 'About the Meal',
                            }}}>
                        />
                    </Stack.Navigator>
                </NavigationContainer>
            </FavoritesContextProvider>
        </>
    ) ;
}
```

```
}
```



```
const styles = StyleSheet.create({
  container: {},
});
```

2. managing app wide state with context

In favoritescontext.js

```
import { createContext, useState } from 'react';

export const FavoritesContext = createContext({
  ids: [],
  addFavorite: (id) => {},
  removeFavorite: (id) => {},
});

function FavoritesContextProvider({ children }) {
  const [favoriteMealIds, setFavoriteMealIds] = useState([]);

  function addFavorite(id) {
    setFavoriteMealIds((currentFavIds) => [...currentFavIds, id]);
  }

  function removeFavorite(id) {
    setFavoriteMealIds((currentFavIds) =>
      currentFavIds.filter((mealId) => mealId !== id)
    );
  }

  const value = {
    ids: favoriteMealIds,
    addFavorite: addFavorite,
    removeFavorite: removeFavorite,
  };
}

return (
  <FavoritesContext.Provider value={value}>
```

```

        {children}
      </FavoritesContext.Provider>
    ) ;
}

export default FavoritesContextProvider;

```

3) using context-api in mealscreen.js

In mealdetailscreen.js

```

import { useContext, useLayoutEffect } from 'react';
import { View, Text, Image, StyleSheet, ScrollView } from 'react-native';

import IconButton from '../components/IconButton';
import List from '../components/MealDetail/List';
import Subtitle from '../components/MealDetail/Subtitle';
import MealDetails from '../components/MealDetails';
import { MEALS } from '../data/dummy-data';
import { FavoritesContext } from '../store/context/favorites-context';

function MealDetailScreen({ route, navigation }) {
  const favoriteMealsCtx = useContext(FavoritesContext);

  const mealId = route.params.mealId;
  const selectedMeal = MEALS.find((meal) => meal.id === mealId);

  const mealIsFavorite = favoriteMealsCtx.ids.includes(mealId);

  function changeFavoriteStatusHandler() {
    if (mealIsFavorite) {
      favoriteMealsCtx.removeFavorite(mealId);
    } else {
      favoriteMealsCtx.addFavorite(mealId);
    }
  }

  useLayoutEffect(() => {
    navigation.setOptions({
      headerRight: () => {

```

```

        return (
          <IconButton
            icon={mealIsFavorite ? 'star' : 'star-outline'}
            color="white"
            onPress={changeFavoriteStatusHandler}
          />
        );
      },
    );
  },
  [navigation, changeFavoriteStatusHandler]);
}

return (
  <ScrollView style={styles.rootContainer}>
    <Image style={styles.image} source={{ uri: selectedMeal.imageUrl }} />
    <Text style={styles.title}>{selectedMeal.title}</Text>
    <MealDetails
      duration={selectedMeal.duration}
      complexity={selectedMeal.complexity}
      affordability={selectedMeal.affordability}
      textStyle={styles.detailText}
    />
    <View style={styles.listOuterContainer}>
      <View style={styles.listContainer}>
        <Subtitle>Ingredients</Subtitle>
        <List data={selectedMeal.ingredients} />
        <Subtitle>Steps</Subtitle>
        <List data={selectedMeal.steps} />
      </View>
    </View>
  </ScrollView>
);
}

export default MealDetailScreen;

const styles = StyleSheet.create({
  rootContainer: {
    marginBottom: 32,
  },

```

```
image: {
  width: '100%',
  height: 350,
},
title: {
  fontWeight: 'bold',
  fontSize: 24,
  margin: 8,
  textAlign: 'center',
  color: 'white',
},
detailText: {
  color: 'white',
},
listOuterContainer: {
  alignItems: 'center',
},
listContainer: {
  width: '80%',
},
});
});
```

Output:



4.managing favorite meals

1) in mealList.js(components)

```
import { View, FlatList, StyleSheet } from 'react-native';

import MealItem from './MealItem';

function MealsList({items}) {
  function renderMealItem(itemData) {
    const item = itemData.item;

    const mealItemProps = {
      id: item.id,
      title: item.title,
      imageUrl: item.imageUrl,
      affordability: item.affordability,
      complexity: item.complexity,
      duration: item.duration,
    };
    return <MealItem {...mealItemProps} />;
  }

  return (
    <View style={styles.container}>
      <FlatList
        data={items}
        keyExtractor={(item) => item.id}
        renderItem={renderMealItem}
      />
    </View>
  );
}

export default MealsList;

const styles = StyleSheet.create({
  container: {
    flex: 1,
    padding: 16,
  },
});
```

```
});
```

In mealsoverviewscreen.js

```
import { useLayoutEffect } from 'react';

import MealsList from '../components/MealsList/MealsList';
import { MEALS, CATEGORIES } from '../data/dummy-data';

function MealsOverviewScreen({ route, navigation }) {
  const catId = route.params.categoryId;

  const displayedMeals = MEALS.filter((mealItem) => {
    return mealItem.categoryIds.indexOf(catId) >= 0;
  });

  useLayoutEffect(() => {
    const categoryTitle = CATEGORIES.find(
      (category) => category.id === catId
    ).title;

    navigation.setOptions({
      title: categoryTitle,
    });
  }, [catId, navigation]);

  return <MealsList items={displayedMeals} />;
}

export default MealsOverviewScreen;
```

c) in favoritescreen.js

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

import MealsList from '../components/MealsList/MealsList';
import { FavoritesContext } from '../store/context/favorites-context';
import { MEALS } from '../data/dummy-data';
```

```
function FavoritesScreen() {
  const favoriteMealsCtx = useContext(FavoritesContext);

  const favoriteMeals = MEALS.filter((meal) =>
    favoriteMealsCtx.ids.includes(meal.id)
  );

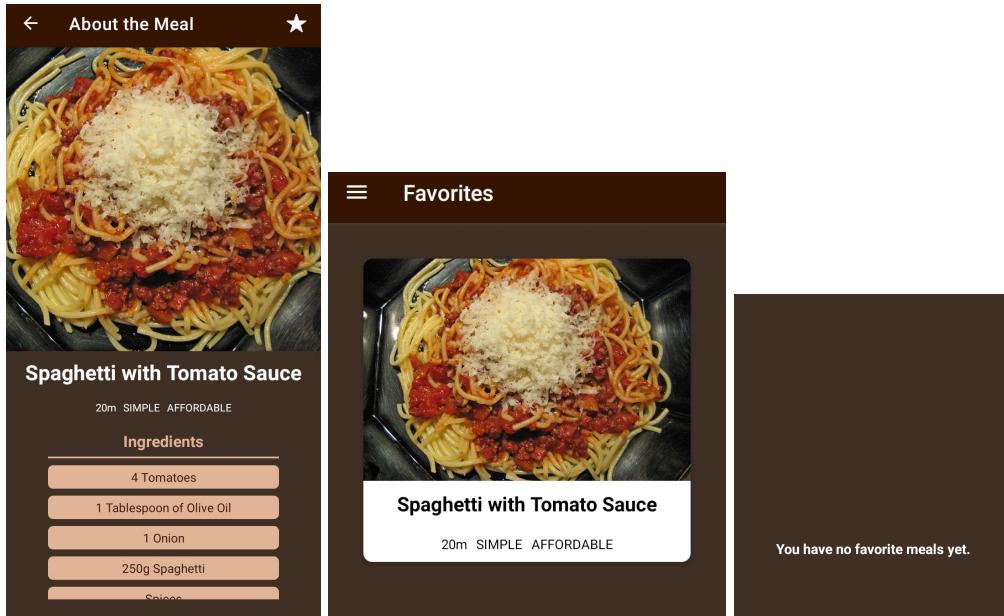
  if (favoriteMeals.length === 0) {
    return (
      <View style={styles.rootContainer}>
        <Text style={styles.text}>You have no favorite meals yet.</Text>
      </View>
    );
  }

  return <MealsList items={favoriteMeals} />;
}

export default FavoritesScreen;

const styles = StyleSheet.create({
  rootContainer: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
  },
  text: {
    fontSize: 18,
    fontWeight: 'bold',
    color: 'white',
  },
});
```

Output:



5.) Getting started with redux

a) Install redux

```
npm install @reduxjs/toolkit
```

b) create a redux folder and store.js

In store.js

```
import { configureStore } from '@reduxjs/toolkit';

export const store = configureStore({
  reducer: {}
});
```

In app.js

```
import { StatusBar } from 'expo-status-bar';
import { StyleSheet, Button } from 'react-native';
import { NavigationContainer } from '@react-navigation/native';
import { createNativeStackNavigator } from
  '@react-navigation/native-stack';
import { createDrawerNavigator } from '@react-navigation/drawer';
import { Ionicons } from '@expo/vector-icons';
import { Provider } from 'react-redux';

import CategoriesScreen from './screens/CategoriesScreen';
```

```
import MealsOverviewScreen from './screens/MealsOverviewScreen';
import MealDetailScreen from './screens/MealDetailScreen';
import FavoritesScreen from './screens/FavoritesScreen';
// import FavoritesContextProvider from
'./store/context/favorites-context';
import { store } from './store/redux/store';

const Stack = createNativeStackNavigator();
const Drawer = createDrawerNavigator();

function DrawerNavigator() {
  return (
    <Drawer.Navigator
      screenOptions={{
        headerStyle: { backgroundColor: '#351401' },
        headerTintColor: 'white',
        sceneContainerStyle: { backgroundColor: '#3f2f25' },
        drawerContentStyle: { backgroundColor: '#351401' },
        drawerInactiveTintColor: 'white',
        drawerActiveTintColor: '#351401',
        drawerActiveBackgroundColor: '#e4baa1',
      }}
    >
    <Drawer.Screen
      name="Categories"
      component={CategoriesScreen}
      options={{
        title: 'All Categories',
        drawerIcon: ({ color, size }) => (
          <Ionicons name="list" color={color} size={size} />
        ),
      }}
    />
    <Drawer.Screen
      name="Favorites"
      component={FavoritesScreen}
      options={{
        drawerIcon: ({ color, size }) => (
          <Ionicons name="star" color={color} size={size} />
        ),
      }},
    >
  
```

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

export default function App() {
return (
<>
<StatusBar style="light" />
{/* <FavoritesContextProvider> */}
<Provider store={store}>
<NavigationContainer>
<Stack.Navigator
screenOptions={{
headerStyle: { backgroundColor: '#351401' },
headerTintColor: 'white',
contentStyle: { backgroundColor: '#3f2f25' },
}}
>
<Stack.Screen
name="Drawer"
component={DrawerNavigator}
options={{
headerShown: false,
}}
/>
<Stack.Screen
name="MealsOverview"
component={MealsOverviewScreen}
/>
<Stack.Screen
name="MealDetail"
component={MealDetailScreen}
options={{
title: 'About the Meal',
}}
/>
</Stack.Navigator>
</NavigationContainer>
```

```

        </Provider>
        {/* </FavoritesContextProvider> */}
    </>
);
}

const styles = StyleSheet.create({
  container: {},
});

```

6) working with redux slices

Create a file called **favorites.js** in **redux** folder.

In **favorites.js**

```

import { createSlice } from '@reduxjs/toolkit';

const favoritesSlice = createSlice({
  name: 'favorites',
  initialState: {
    ids: []
  },
  reducers: {
    addFavorite: (state, action) => {
      state.ids.push(action.payload.id);
    },
    removeFavorite: (state, action) => {
      state.ids.splice(state.ids.indexOf(action.payload.id), 1);
    }
  }
);

export const addFavorite = favoritesSlice.actions.addFavorite;
export const removeFavorite = favoritesSlice.actions.removeFavorite;
export default favoritesSlice.reducer;

```

b) in **store.js**

```

import { configureStore } from '@reduxjs/toolkit';

import favoritesReducer from './favorites';

export const store = configureStore({
  reducer: {
    favoriteMeals: favoritesReducer
  }
});

```

7) managing redux state In mealdetailscreen.js

```

import { useLayoutEffect } from 'react';
import { View, Text, Image, StyleSheet, ScrollView } from 'react-native';
import { useDispatch, useSelector } from 'react-redux';

import IconButton from '../components/IconButton';
import List from '../components/MealDetail/List';
import Subtitle from '../components/MealDetail/Subtitle';
import MealDetails from '../components/MealDetails';
import { MEALS } from '../data/dummy-data';
import { addFavorite, removeFavorite } from '../store/redux/favorites';
// import { FavoritesContext } from '../store/context/favorites-context';

function MealDetailScreen({ route, navigation }) {
  // const favoriteMealsCtx = useContext(FavoritesContext);
  const favoriteMealIds = useSelector((state) => state.favoriteMeals.ids);
  const dispatch = useDispatch();

  const mealId = route.params.mealId;
  const selectedMeal = MEALS.find((meal) => meal.id === mealId);

  const mealIsFavorite = favoriteMealIds.includes(mealId);

  function changeFavoriteStatusHandler() {
    if (mealIsFavorite) {
      // favoriteMealsCtx.removeFavorite(mealId);
      dispatch(removeFavorite({ id: mealId }));
    }
  }
}

```

```
    } else {
      // favoriteMealsCtx.addFavorite(mealId);
      dispatch(addFavorite({ id: mealId }));
    }
  }

useLayoutEffect(() => {
  navigation.setOptions({
    headerRight: () => {
      return (
        <IconButton
          icon={mealIsFavorite ? 'star' : 'star-outline'}
          color="white"
          onPress={changeFavoriteStatusHandler}
        />
      );
    },
  });
}, [navigation, changeFavoriteStatusHandler]);

return (
  <ScrollView style={styles.rootContainer}>
    <Image style={styles.image} source={{ uri: selectedMeal.imageUrl }} />
    <Text style={styles.title}>{selectedMeal.title}</Text>
    <MealDetails
      duration={selectedMeal.duration}
      complexity={selectedMeal.complexity}
      affordability={selectedMeal.affordability}
      textStyle={styles.detailText}
    />
    <View style={styles.listOuterContainer}>
      <View style={styles.listContainer}>
        <Subtitle>Ingredients</Subtitle>
        <List data={selectedMeal.ingredients} />
        <Subtitle>Steps</Subtitle>
        <List data={selectedMeal.steps} />
      </View>
    </View>
  </ScrollView>
)
```

```
) ;  
}  
  
export default MealDetailScreen;  
  
const styles = StyleSheet.create({  
  rootContainer: {  
    marginBottom: 32,  
  },  
  image: {  
    width: '100%',  
    height: 350,  
  },  
  title: {  
    fontWeight: 'bold',  
    fontSize: 24,  
    margin: 8,  
    textAlign: 'center',  
    color: 'white',  
  },  
  detailText: {  
    color: 'white',  
  },  
  listOuterContainer: {  
    alignItems: 'center',  
  },  
  listContainer: {  
    width: '80%',  
  },  
});
```

Output:



8) using redux state in components In favoritesscreen.js

```
import { View, Text, StyleSheet } from 'react-native';
import { useSelector } from 'react-redux';
// import { useContext } from 'react';

import MealsList from '../components/MealsList/MealsList';
// import { FavoritesContext } from '../store/context/favorites-context';
import { MEALS } from '../data/dummy-data';

function FavoritesScreen() {
    // const favoriteMealsCtx = useContext(FavoritesContext);
    const favoriteMealIds = useSelector((state) => state.favoriteMeals.ids);

    const favoriteMeals = MEALS.filter((meal) =>
        favoriteMealIds.includes(meal.id)
    );

    if (favoriteMeals.length === 0) {
        return (
            <View style={styles.rootContainer}>
                <Text style={styles.text}>You have no favorite meals yet.</Text>
            </View>
        );
    }
}
```

```

        }

      return <MealsList items={favoriteMeals} />;
    }

export default FavoritesScreen;

const styles = StyleSheet.create({
  rootContainer: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
  },
  text: {
    fontSize: 18,
    fontWeight: 'bold',
    color: 'white',
  },
});

```

Output:

