# MOBILE PROGRAMMING

## LAB 3

## Contents

**Create github Lab3 with Lab3 folder on computer. Submit assignments to the moodle system.**

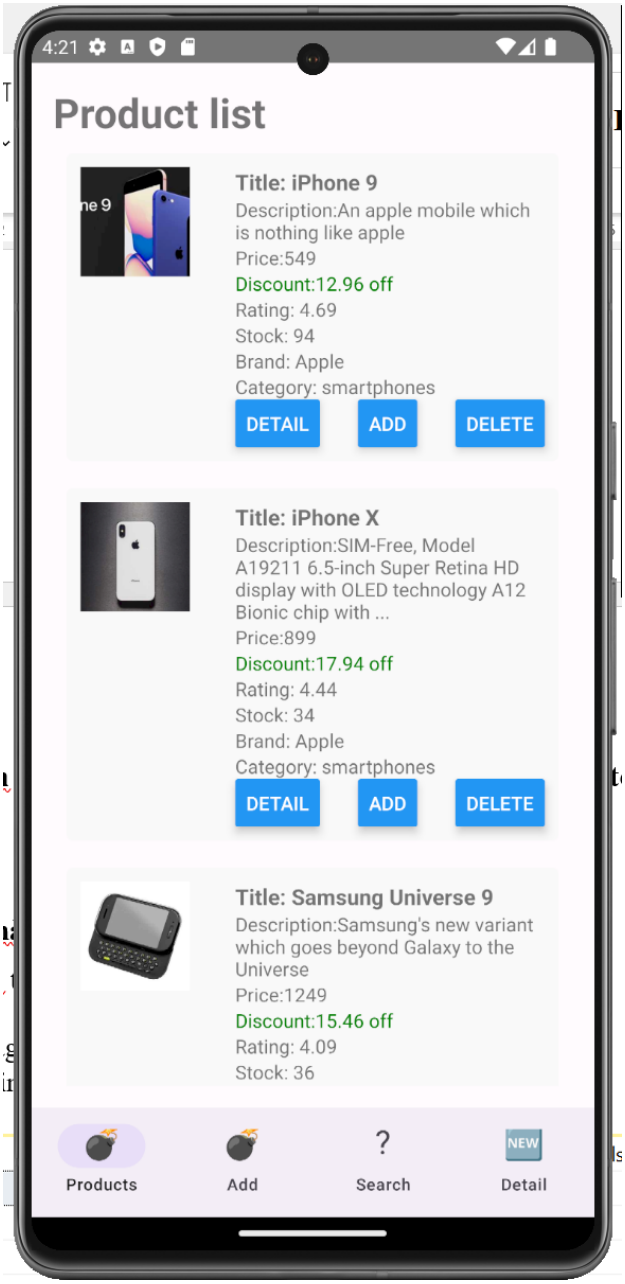
**Create project Lab3, Create a Products folder**

**Install libraries:**

*npm install react-native-safe-area-context*

*npm install react-native-paper*

**Question 1. Develop the ProductList screen as below.**

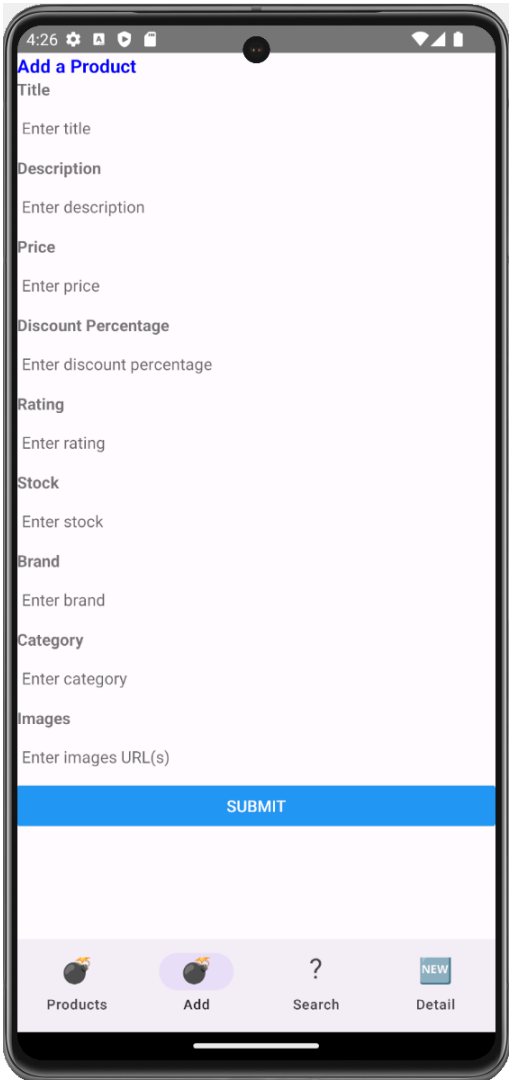


Note: Detail, add, delete buttons do not handle.

1. Create Products.js file
2. Use Flatlist to present the list.
3. The source code that reads the Products data returns Json data.

```
const [data,setData] = useState([])
const filePath = 'https://dummyjson.com/products/';
useEffect(() => {
    //  Alert.alert(filePath);
    fetch(filePath)
        .then((response) => {
            if (!response.ok) {
                throw new Error('Network response was not ok');
            }
            return response.json();
        })
        .then((d) => {
            setData(d.products);
        })
        .catch((error) => {
            console.error('Error fetching data:', error);
        });
});
```

**Question 2. Develop the "Add a Product" screen as below.**




1. Create Product_Add.js file
2. The source code adds data when the user clicks submit.

```
const [title,setTitle] = useState('');
const [description,setDescription] = useState('');
const [price,setPrice] = useState('');
const [discountPercentage,setDiscountPercentage] = useState('');
const [rating,setRating] = useState('');
const [stock,setStock] = useState('');
const [brand,setBrand] = useState('');
const [category,setCategory] = useState('');
const [images,setImages] = useState('');
handleSubmit = () => {
  fetch('https://dummyjson.com/products/add', {
   method: 'POST',
   headers: { 'Content-Type': 'application/json' },
   body: JSON.stringify({
     title: title,
     description: description,
     price: price,
     discountPercentage: discountPercentage,
     rating: rating,
     stock: stock,
     brand: brand,
     category: category,
     images: images,
   }),
  })
    .then((res) => res.json())
    .then(console.log);
  Alert.alert("Add sucessfull")
};
```

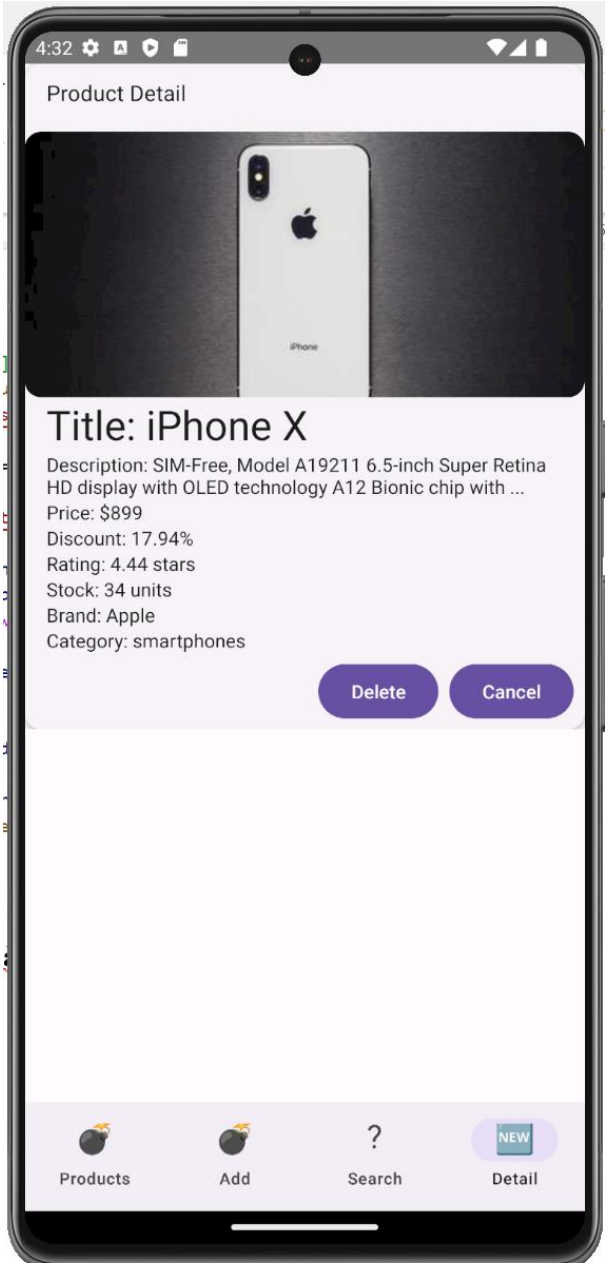
**Question 3. Develop the Search products screen as below.**


1. Create Product_Search.js file
2. Using CARD in react-native-paper library, Flatlist
3. The source code when the user clicks Search.

```
const [data, setData] = useState([]);
const [value, setValue] = useState('');
let filePath = 'https://dummyjson.com/products';

const searchProduct = () => {
    if(value!='')
       filePath = 'https://dummyjson.com/products/search?q=' + value;
    fetch(filePath)
        .then((response) => {
            if (!response.ok) {
                throw new Error('Network response was not ok');
            }
            return response.json();
        })
        .then((d) => {
            setData(d.products)
        })
        .catch((error) => {
            console.error('Error fetching data:', error);
        });
};
```

## Question 4. Develop the Detail screen as below.



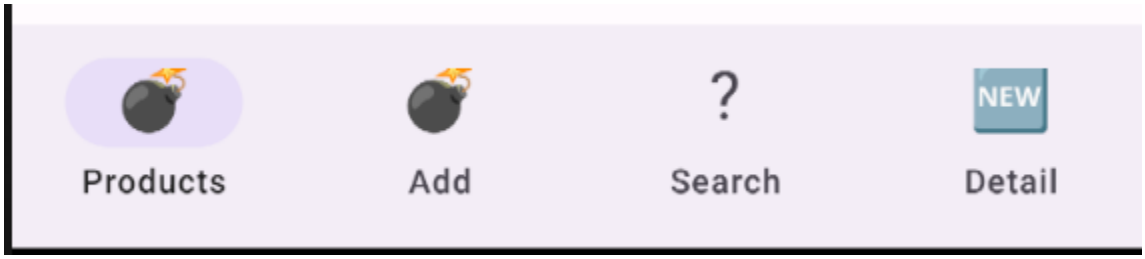

Note: Do not handle delete, cancel buttons

1. Create Product_Detail.js file
2. Use CARD, button in react-native-paper library
3. Source code when a user retrieves a product

```
const [data,setData] = useState([])
 const filePath = 'https://dummyjson.com/products/2';

 useEffect(() => {
    //  Alert.alert(filePath);
    fetch(filePath)
      .then((response) => {
        if (!response.ok) {
          throw new Error('Network response was not ok');
        }
        return response.json();
      })
      .then((d) => {
        setData(d);
      })
      .catch((error) => {
        console.error('Error fetching data:', error);
      });
 });
```

**Question 5. Develop Navigation as shown below.**




1. Choose your own icon.
2. Use BottomNavigation in react-native-paper library, SafeAreaProvider in react-native-safe-area-context library.
   Botton navigation: https://callstack.github.io/react-native-paper/docs/components/BottomNavigation/
   Icon: https://callstack.github.io/react-native-paper/docs/guides/icons/

3. Source code in App.tsx file.

```jsx
import React, { useState } from 'react'

import ProductList from './Products/Products';
import Product_Add from './Products/Product_Add';
import ProductDetail from './Products/Product_Detail';
import ProductSearch from './Products/Product_Search';
import { BottomNavigation, Text } from 'react-native-paper'
import { SafeAreaProvider } from 'react-native-safe-area-context';


export default App = () => {
  const [index, setIndex] = useState(0);
  const [routes] = useState([
    { key: 'ProductList', title: 'Products', focusedIcon: 'folder'},
    { key: 'Product_Add', title: 'Add', focusedIcon: 'folder' },
    { key: 'ProductSearch', title: 'Search', focusedIcon: 'find' },
    { key: 'Product_Detail', title: 'Detail', focusedIcon: 'calendar' },
  ]);

  const renderScene = BottomNavigation.SceneMap({
    ProductList: ProductList,
    Product_Add: Product_Add,
    ProductSearch: ProductSearch,
    Product_Detail: ProductDetail,
  });

  return (
    <SafeAreaProvider>
      <BottomNavigation
        navigationState={{ index, routes }}
        onIndexChange={setIndex}
        renderScene={renderScene}
      />
    </SafeAreaProvider>
  );
};
```