|  |  |
| --- | --- |
| **Name and Registration No.** | **Fazeela Rehman (Fa20-Bcs-090)**  **Syed Muhammad Shaheer Ali Shah (Fa20-bcs079)** |
| **Section** | **A** |
| **Subject** | **Mobile Application Development** |
| **Lab Assignment** | **02** |
| **Teacher** | **Dr. Osman Khalid** |

**Question No.1**

<https://snack.expo.dev/@fazeelarehman410/context_api>

**App.js**

const Tab = createBottomTabNavigator();

function App() {

  return (

    <NavigationContainer>

      <Tab.Navigator>

        <Tab.Screen

          name="Student Info"

          component={BottomTabScreen1}

          options={{ headerShown: false }}

        />

        <Tab.Screen

          name="Location"

          component={BottomTabScreen2}

          options={{ headerShown: false }}

        />

      </Tab.Navigator>

    </NavigationContainer>

  );

}

export default App;

**BottomTabScreen1.js**

const Stack = createNativeStackNavigator();

const Context1 = React.createContext(null);

function BottomTabScreen1() {

  const [studentInfo, setStudentInfo] = useState({ name: 'Ali', regNo: 12 });

  const studentInfoSetters = { setName, setRegNo, setData };

  function setName(name) {

    const updatedInfo = { ...studentInfo, name };

    setStudentInfo(updatedInfo);

  }

  function setRegNo(regNo) {

    const updatedInfo = { ...studentInfo, regNo };

    setStudentInfo(updatedInfo);

  }

  function setData(name, regNo) {

    const updatedInfo = { ...studentInfo, name, regNo };

    setStudentInfo(updatedInfo);

  }

  return (

    <Context1.Provider value={{ ...studentInfo, ...studentInfoSetters }}>

      <Stack.Navigator>

        <Stack.Screen name="Screen1" component={Screen1} />

        <Stack.Screen name="Screen2" component={Screen2} />

      </Stack.Navigator>

    </Context1.Provider>

  );

}

export default BottomTabScreen1;

**Screen 1.js**

function Screen1({ navigation }) {

  const context = React.useContext(Context1);

  return (

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

      <Text>Student Name: {context.name}</Text>

      <Text>Registration No.{context.regNo}</Text>

      <Text>{'\n'}</Text>

      <TouchableOpacity

        style={styles.btn}

        onPress={() => {

          navigation.navigate('Screen2');

        }}>

        <Text style={styles.btntxt}>Go To Screen2</Text>

      </TouchableOpacity>

    </View>

  );

}

**More Screens** **BottomTabScreen1.js, Screen3.js, Screem4.js,Screen2.js**

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

**Question No.2**

<https://snack.expo.dev/@shaheershah079/labassignment2question2_asynchronousstorage>

**App.js**

function App() {

  return (

    <NavigationContainer>

      <Stack.Navigator initialRouteName="Student Section">

        <Stack.Screen

          name="Student Registration"

          component={StudentRegistraionScreen}

        />

        <Stack.Screen

          name="Search Student"

          component={SearchAdminStudentScreen}

        />

        <Stack.Screen

          name="Student Section"

          component={StudentSection}

          options={{ headerShown: false }}

        />

      </Stack.Navigator>

    </NavigationContainer>

  );

}

**StudentSection.js**

const StudentSection = () => {

  return (

    <Provider store={store}>

      <Stack.Navigator initialRouteName="LogIn">

        <Stack.Screen name="LogIn" component={LoginScreen} />

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

        <Stack.Screen

          name="Course Registration"

          component={CourseRegistration}

          options={({ route }) => ({ title: route.params?.name })}

        />

        <Stack.Screen

          name="Search Course"

          component={SearchCourse}

          options={({ route }) => ({ title: route.params?.name })}

        />

      </Stack.Navigator>

    </Provider>

  );

};

**LoginScreen.js**

const LoginScreen=({navigation})=>{

  const dispatch =useDispatch()

  const LogIn=async()=>{

    if(reg=='admin'&& password=='123'){

*// await AsyncStorage.removeItem('Students');*

        navigation.navigate('Student Registration')

      }

      else{

    const jsonValue=await AsyncStorage.getItem('Students')

    if(jsonValue!=null){

      const value=JSON.parse(jsonValue);

      var student=false;

      for(let i=0;i<value.length;i++){

        if(value[i].RegNo==reg.toUpperCase() && value[i].password==password){

          student=true;

           dispatch(setRegistration(value[i].RegNo))

          dispatch(setName(value[i].name))

        navigation.navigate("Student Home")

        }

      }

      if(!student){

       alert('Incorrect Registration Password')

      }

    }

    else{

      alert('User Not found')

    }

      }

  }

const [reg,setReg]=useState('')

const [password,setPassword]=useState('')

return(

  <View style={styles.container}>

    <TextInput label="Registration Number" mode="outlined" style={styles.textinput} onChangeText={setReg}/>

    <TextInput label="Password" mode="outlined" style={styles.textinput} onChangeText={setPassword}/>

    <TouchableOpacity style={styles.button}

    onPress={()=>{LogIn()}}>

      <Text>Login </Text>

    </TouchableOpacity>

  </View>

)

}

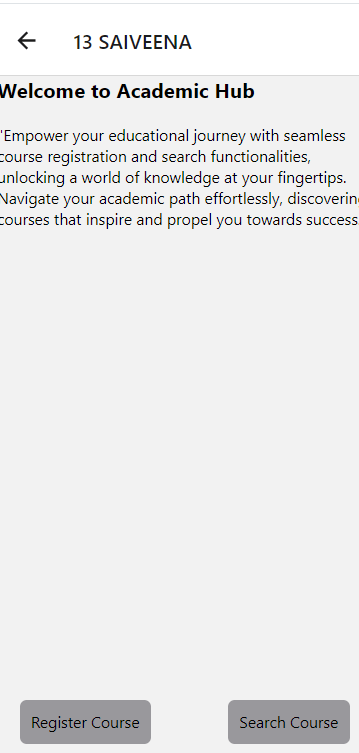
export default LoginScreen;

**More Screens StudentRegistration.js , StudentSlice.js, store.js ,SearchCourse.js, SearchStudent.js , StudentHomeScreen , CourseRegistration**

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

**Question No. 3**

<https://snack.expo.dev/@shaheershah079/labassignment2_question3_sqllitefazeela>

**App.js**

const Stack = createNativeStackNavigator();

function App() {

    useEffect(

    ()=>{

  createTable()

  insertRecord()

    },[]

  )

  return (

    <NavigationContainer>

      <Stack.Navigator>

        <Stack.Screen name="Record Search" component={SearchRecord} />

        <Stack.Screen name="View Record" component={ViewRecord} />

      </Stack.Navigator>

    </NavigationContainer>

  );

}

export default App;

**ViewRecord.js:**

function ViewRecord({route}) {

  id=route.params?.id

  useEffect(

    ()=>{

    getData()

    },[]

  )

  const[record,setRecord]=useState({})

  const getData=()=>{

    db.transaction(

      (tx)=>{

        tx.executeSql(

          "Select \* from RECORDS where ID=?",

          [id],

           (z,result)=>{

            var length=result.rows.length

            if(length!=0){

              console.log(result.rows.item(0))

              setRecord(result.rows.item(0))

            }

          },

          (e)=>{console.log(e)}

        )

      }

    )

  }

    return (

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

    <View style={{flexDirection:'column', alignItems:'center'}}>

    <View style={{flexDirection:'row', alignItems:'center'}}>

      <Text style={styles.txt}>ID:                  </Text>

      <Text style={styles.txtIn}>{record.ID}</Text>

      </View>

      </View>

      <Text>{"\n"}</Text>

      <View style={{flexDirection:'column', alignItems:'center'}}>

    <View style={{flexDirection:'row', alignItems:'center'}}>

      <Text style={styles.txt}>NAME:            </Text>

      <Text style={styles.txtIn}>{record.NAME}</Text>

      </View>

      </View>

      <Text>{"\n"}</Text>

      <View style={{flexDirection:'column', alignItems:'center'}}>

    <View style={{flexDirection:'row', alignItems:'center'}}>

      <Text style={styles.txt}>AGE:               </Text>

      <Text style={styles.txtIn}>{record.AGE}</Text>

      </View>

      </View>

      <Text>{"\n"}</Text>

      <View style={{flexDirection:'column'}}>

    <View style={{flexDirection:'row'}}>

      <Text style={styles.txt} >ADDRESS:      </Text>

      <Text style={styles.txtInAdd} >{record.ADDRESS}</Text>

      </View>

      </View>

    </View>

  );

}

**SearchRecord.js:1**

function SearchRecord({navigation}) {

  const[id,setId]=useState("")

  return (

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

<View style={{flexDirection:'row'}}>

     <View style={{flexDirection:'column', alignItems:'center'}}>

      <Text style={styles.txt}>Enter ID:      </Text>

      </View>

      <View style={{flexDirection:'column'}}>

        <TextInput style={styles.txtIn} value={id} onChangeText={setId}></TextInput>

       <Text>{"\n"}</Text>

      <TouchableOpacity style={styles.btn} onPress={()=>{

        navigation.navigate('View Record',{'id':id})

        }}><Text style={styles.btntxt}>Search</Text></TouchableOpacity>

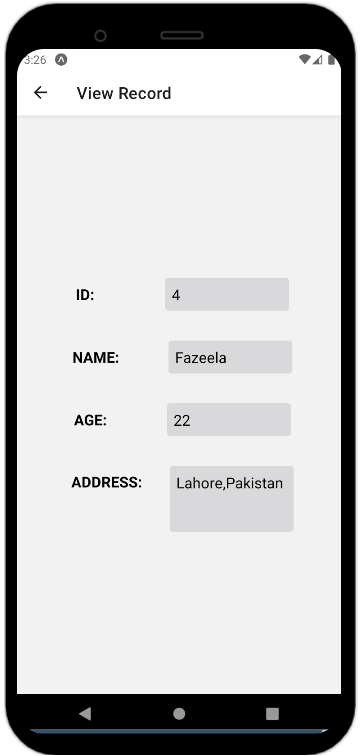
    </View>

    </View>

    </View>

  );

}

A screen shot of a computer

Description automatically generated

**Question 4:**

<https://github.com/ShaheerShah079/LabAssignment2Q4php>

**App.js:**

const Stack = createNativeStackNavigator();

export const ipAdrr='192.168.19.13';

function App() {

  return (

    <NavigationContainer>

      <Stack.Navigator initialRouteName='LogIn'>

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

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

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

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

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

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

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

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

      </Stack.Navigator>

    </NavigationContainer>

  );

}

**ViewProduct.js:**

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

    async function getProduct(){

        try{

            console.log('http:/'+ipAdrr+'//GetProducts.php')

            const response=await fetch('http:/'+ipAdrr+'//GetProducts.php');

            const res=await response.json()

            if(res.length==0){

                alert('No Products Available')

            }

            else{

                console.log(res)

                setProduct(res)

            }

          }

          catch(e){

            console.log('error')

          }

    }

    useEffect(

        ()=>{

            getProduct()

        },[]

    )

    const [product,setProduct]=useState([]);

    return (

<View

      style={{

        flex: 1,

        alignItems: 'center',

        justifyContent: 'center',

        backgroundColor: '#ffc266',

      }}>

      <ScrollView>

        <FlatList

          data={product}

          renderItem={({ item }) => (

            <TouchableOpacity

            style={{backgroundColor: '#ffffff',margin:10,padding:20,borderRadius:6}}

              onPress={() =>

                navigation.navigate('ProductDetail', {

                  id: item.ProductID,

                  name: item.ProductName,

                  description: item.ProductDescription,

                  price: item.ProductPrice,

                  category: item.ProductCategory,

                })

              }>

              <Text style={{color: '#ff8000',fontSize:18}}>

                {'Item Name : ' +

                  item.ProductName +

                  ' Item Price : ' +

                  item.ProductPrice}

              </Text>

            </TouchableOpacity>

          )}

        />

        <View style={{alignItems:'center'}}>

        <TouchableOpacity style={styles.btn} onPress={()=>navigation.navigate('PlaceOrders',

            {'customerId':route.params?.customerId})}>

          <Text style={{fontWeight:'bold',fontSize:18,color:'#ff8000'}}>Cart</Text>

        </TouchableOpacity>

        </View>

      </ScrollView>

    </View>

      )

    }

**GetProducts.php**

<?php

try{

include('opendb.php');

$query = "select \* from products";

$stmt = $conn->prepare($query);

$stmt->execute();

$resultJSON = json\_encode($stmt->fetchAll(PDO::FETCH\_ASSOC));

echo $resultJSON;

}

catch(PDOException $e)

{

echo $e -> getMessage();

}

?>

**More Screen Login.js,** **SignUp.js ProductDetai.jsl, PlaceOrder.js , ConfirmOrder.js , AddProduct.js,ReviewOrder.js,**

**AddProduct.php,GetOrder.php,GetProduct.php,GetReviewOrder.php,insertOrder.php,Login.phpopendb.php,SignUp.php,PlaceInorderTable.php,UpdateOrderStatus**