

Lab 4 : JavaScript for Mobile Device Programming

1. ให้นักศึกษาทำการสร้าง New Project ใหม่ ชื่อ ProjectCal ใน Folder Mobile\<รหัสนักศึกษา>\ProjectCal แบบ Blank Project โดยใช้คำสั่ง

Expo init <path\folder\StudentID\Project name>

```
C:\Mobile\61070218>expo init ProjectCal --npm
? Choose a template: expo-template-blank
√ Downloaded and extracted project files.
√ Installed JavaScript dependencies.

✓ Your project is ready!

To run your project, navigate to the directory and run one of the following npm commands.

- cd ProjectCal
- npm start # you can open iOS, Android, or web from here, or run them directly with the commands below.
- npm run android
- npm run ios # requires an iOS device or macOS for access to an iOS simulator
- npm run web
```

2. ให้นักศึกษาทำการสร้างโปรแกรมเครื่องคิดเลขบนอุปกรณ์เคลื่อนที่ ดังแสดงตามรูปข้างล่างนี้



บันทึกผลการทดลอง :

```
import React, { useState } from "react";
import { StyleSheet, Text, View } from "react-native";

import MyButton from "../components/MyButton";

export default function App() {
  const [expressionText, setExpressionText] = useState("");
  const [resultText, setResultText] = useState("");

  const buttonPressHandler = (text) => {
    switch (text) {
      case "=":
        try {
          setResultText(eval(expressionText));
        } catch (err) {
          setResultText("Bad expression");
        }
        break;
      case "DEL":
        setExpressionText((prevExpressionText) =>
          prevExpressionText.substring(0, prevExpressionText.length - 1)
        );
        break;
      default:
        setExpressionText((prevExpressionText) => prevExpressionText + text);
    }
  };
};
```

```

return (
  <View style={styles.screen}>
    <View style={styles.displayContainer}>
      <Text style={styles.expressionText}>{expressionText}</Text>
    </View>
    <View style={styles.displayContainer}>
      <Text style={styles.resultText}>{resultText}</Text>
    </View>
    <View style={styles.buttonContainer}>
      <View style={styles.buttonColumn}>
        <MyButton label="1" onTap={buttonPressHandler} />
        <MyButton label="4" onTap={buttonPressHandler} />
        <MyButton label="7" onTap={buttonPressHandler} />
        <MyButton label="." onTap={buttonPressHandler} />
      </View>
      <View style={styles.buttonColumn}>
        <MyButton label="2" onTap={buttonPressHandler} />
        <MyButton label="5" onTap={buttonPressHandler} />
        <MyButton label="8" onTap={buttonPressHandler} />
        <MyButton label="0" onTap={buttonPressHandler} />
      </View>
      <View style={styles.buttonColumn}>
        <MyButton label="3" onTap={buttonPressHandler} />
        <MyButton label="6" onTap={buttonPressHandler} />
        <MyButton label="9" onTap={buttonPressHandler} />
        <MyButton label="=" onTap={buttonPressHandler} />
      </View>
      <View style={styles.buttonColumn}>
        <MyButton label="DEL" color="#444D54" onTap={buttonPressHandler} />
        <MyButton label="+" color="#444D54" onTap={buttonPressHandler} />
        <MyButton label="-" color="#444D54" onTap={buttonPressHandler} />
        <MyButton label="*" color="#444D54" onTap={buttonPressHandler} />
        <MyButton label="/" color="#444D54" onTap={buttonPressHandler} />
      </View>
    </View>
  </View>
);

```

```
const styles = StyleSheet.create({
  screen: {
    flex: 1,
  },
  displayContainer: {
    flex: 1,
    justifyContent: "flex-end",
    alignItems: "flex-end",
  },
  buttonContainer: {
    flex: 4,
    flexDirection: "row",
  },
  buttonColumn: {
    flex: 1,
  },
  expressionText: {
    fontSize: 25,
    paddingHorizontal: 10,
  },
  resultText: {
    fontSize: 35,
    paddingVertical: 15,
    paddingHorizontal: 15,
  },
});
```

```
import React from "react";
import { Text, TouchableOpacity, StyleSheet } from "react-native";

const MyButton = (props) => {
  return (
    <TouchableOpacity
      style={{
        ...styles.button,
        backgroundColor: props.color ? props.color : "#1E2326",
      }}
      onPress={() => {
        props.onTap(props.label);
      }}
    >
      <Text style={styles.label}>{props.label}</Text>
    </TouchableOpacity>
  );
};

const styles = StyleSheet.create({
  button: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
    marginBottom: -1,
  },
  label: {
    fontSize: 30,
    color: "white",
  },
});

export default MyButton;
```



