

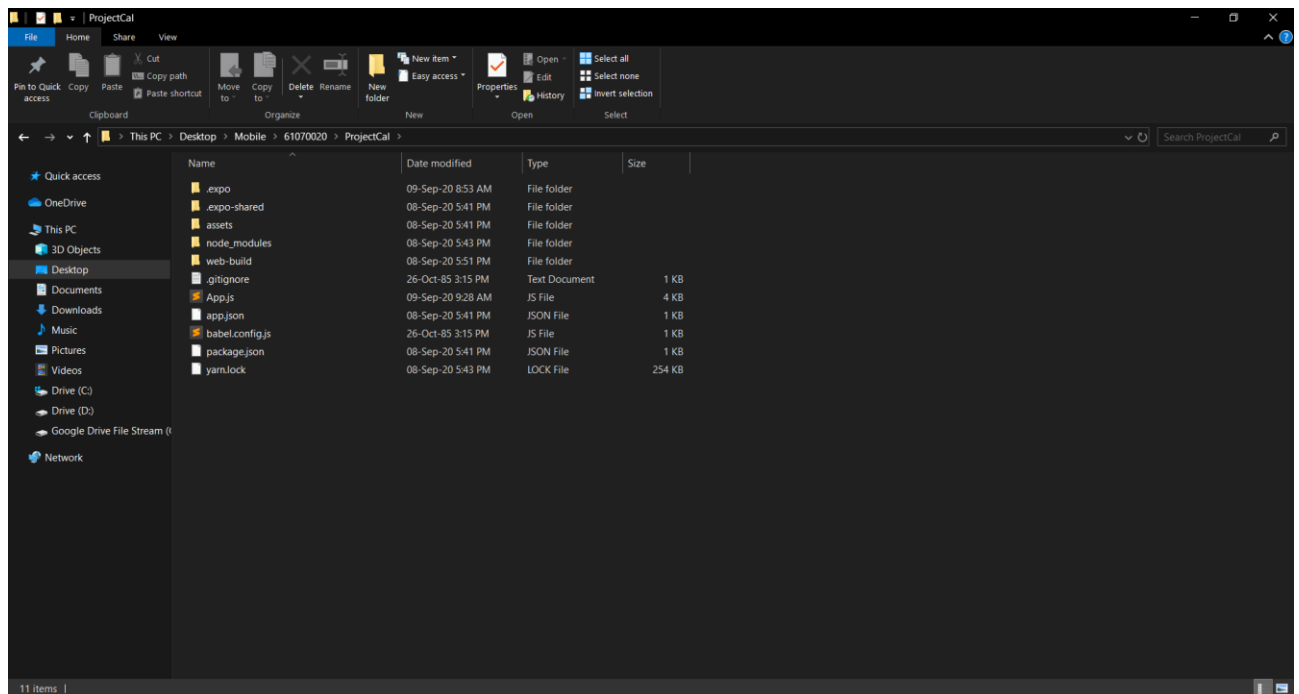
Lab 4 : JavaScript for Mobile Device Programming

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

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

บันทึกผลการทดลอง:

ทำการสร้างโปรเจกใหม่ชื่อ ProjectCal ในโฟลเดอร์ 61070020



2. ให้นักศึกษาทำการสร้างโปรแกรมเครื่องคิดเลขบนอุปกรณ์เคลื่อนที่ ดังแสดงตามรูปข้างล่างนี้



บันทึกผลการทดลอง :

```

JS Appjs X
61070020 > ProjectCal > JS Appjs > App > render
1 import { StatusBar } from 'expo-status-bar';
2 import React, { useState } from 'react';
3 import { StyleSheet, Text, View, TextInput, Button, TouchableOpacity } from 'react-native';
4
5 export default class App extends React.Component {
6   constructor() {
7     super();
8     this.state = {
9       resultText: '',
10      calculateText: ''
11    }
12  }
13
14  calculateResult = () => {
15    const text = this.state.calculateText
16    this.setState({ resultText: eval(text) })
17  }
18
19  checkLastChar = (char) => {
20    const lastChar = this.state.calculateText.slice(-1)
21    if (lastChar == char) {
22      // pass (กด เครื่องหมาย ซ้ำไม่ได้)
23    }
24    else if (lastChar == '+' || lastChar == '-' || lastChar == '*' || lastChar == '/') {
25      if (char == '.') {
26        // pass (กด เครื่องหมายอื่น แล้วเปลี่ยนเป็น . ไม่ได้)
27      } else {
28        this.setState({ calculateText: this.state.calculateText.slice(0, this.state.calculateText.length - 1) + char })
29      }
30    } else {
31      this.setState({ calculateText: this.state.calculateText + char })
32    }
33  }
    
```

```

JS Appjs X
61070020 > ProjectCal > JS Appjs > App > render
33 }
34 }
35
36 buttonPressed = (text) => {
37   if (text == 'DEL') {
38     this.setState({
39       // ลบตัวเลข
40       calculateText: this.state.calculateText.slice(0, this.state.calculateText.length - 1)
41       // ลบเครื่องหมาย
42       // calculateText: this.state.calculateText.slice(-1,0)
43     })
44   } else if (text == '=') {
45     this.calculateResult()
46   } else if (text == '+' || text == '-' || text == '*' || text == '/' || text == '.') {
47     this.checkLastChar(text)
48   } else {
49     this.setState({ calculateText: this.state.calculateText + text })
50   }
51 }
52
JS Appjs X
61070020 > ProjectCal > JS Appjs > styles > result > alignItems
50 }
51 }
52
53 render() {
54   let rows = []
55   let operate = []
56   let nums = [[1, 2, 3], [4, 5, 6], [7, 8, 9], ['.', 0, '=']]
57   let operations = ['DEL', '+', '-', '*', '/', '=']
58   for (let i = 0; i < 4; i++) {
59     let row = []
60     for (let j = 0; j < 3; j++) {
61       row.push(<TouchableOpacity onPress={() => this.buttonPressed(nums[i][j])} style={styles.btn}><Text style={styles.btnText}>{nums[i][j]}</Text></TouchableOpacity>)
62     }
63     rows.push(<View style={styles.row}>{row}</View>)
64   }
65   for (let i = 0; i < 5; i++) {
66     operate.push(<View style={styles.row}><TouchableOpacity onPress={() => this.buttonPressed(operations[i])} style={styles.btn}><Text style={styles.btnText}>{operations[i]}</Text></TouchableOpacity></View>)
67   }
68
69   return (
70     <View style={styles.container}>
71       <View style={styles.textCal}>
72         <Text style={styles.calculateText}>{this.state.calculateText}</Text>
73       </View>
74       <View style={styles.result}>
75         <Text style={styles.resultText}>{this.state.resultText}</Text>
76       </View>
77       <View style={styles.buttons}>
78         <View style={styles.numbers}>
79           {rows}
80         </View>
81         <View style={styles.operation}>
82           {operate}
83         </View>
84       </View>
85     </View>
86   )
87 }
88
JS Appjs X
61070020 > ProjectCal > JS Appjs > ...
88 }
89
90 const styles = StyleSheet.create({
91   container: {
92     paddingTop: 30,
93     flex: 1
94   },
95   result: {
96     flex: 1,
97     backgroundColor: 'white',
98     justifyContent: 'center',
99     alignItems: 'flex-end'
100   },
101   resultText: {
102     fontSize: 30,
103     paddingRight: 10
104   },
105   textCal: {
106     flex: 1,
107     backgroundColor: 'white',
108     justifyContent: 'center',
109     alignItems: 'flex-end'
110   },
111   calculateText: {
112     fontSize: 30,
113     paddingRight: 10
114   },
115   row: {
116     flexDirection: 'row',
117     flex: 1
118   },

```

```

118 },
119 buttons: {
120   flex: 7,
121   flexDirection: 'row'
122 },
123 numbers: {
124   flex: 3,
125   backgroundColor: '#1d2325'
126 },
127 operation: {
128   flex: 1,
129   backgroundColor: '#444e54',
130   justifyContent: 'space-around',
131   alignItems: 'stretch'
132 },
133 btn: {
134   flex: 1,
135   alignItems: 'center',
136   justifyContent: 'center'
137 },
138 btnText: {
139   fontSize: 30,
140   color: 'white'
141 }
142 })

```

Source Code:

```

import { StatusBar } from 'expo-status-bar';
import React, { useState } from 'react';
import { StyleSheet, Text, View, TextInput, Button, TouchableOpacity } from 'react-native';

export default class App extends React.Component {
  constructor() {
    super()
    this.state = {
      resultText: '',
      calculateText: ''
    }
  }

  calculateResult = () => {
    const text = this.state.calculateText
    this.setState({ resultText: eval(text) })
  }

  checkLastChar = (char) => {
    const lastChar = this.state.calculateText.slice(-1)
    if (lastChar == char) {
      // pass (กด เครื่องหมาย ซ้ำไม่ได้)
    }
    else if (lastChar == '+' || lastChar == '-'
    || lastChar == '*' || lastChar == '/') {
      if (char == '.') {
        // pass (กด เครื่องหมายอื่น แล้วเปลี่ยนเป็น . ไม่ได้)
      } else {

```

```

        this.setState({ calculateText: this.state.calculateText.slice(0, this.state
.calculateText.length - 1) + char })
    }
}
else {
    this.setState({ calculateText: this.state.calculateText + char })
}
}

buttonPressed = (text) => {
    if (text == 'DEL') {
        this.setState({
            // ลบตัวสุดท้าย
            calculateText: this.state.calculateText.slice(0, this.state.calculateText.l
ength - 1)
            // ลบหมด
            // calculateText: this.state.calculateText.slice(-1,0)
        })
    } else if (text == '=') {
        this.calculateResult()
    } else if (text == '+' || text == '-'
|| text == '*' || text == '/' || text == '.') {
        this.checkLastChar(text)
    } else {
        this.setState({ calculateText: this.state.calculateText + text })
    }
}

render() {
    let rows = []
    let operate = []
    let nums = [[1, 2, 3], [4, 5, 6], [7, 8, 9], ['.', 0, '=']]
    let operations = ['DEL', '+', '-', '*', '/']
    for (let i = 0; i < 4; i++) {
        let row = []
        for (let j = 0; j < 3; j++) {
            row.push(<TouchableOpacity onPress={() => this.buttonPressed(nums[i][j])} s
tyle={styles.btn}><Text style={styles.btnText}>{nums[i][j]}</Text></TouchableOpacit
y>)
        }
        rows.push(<View style={styles.row}>{row}</View>)
    }
    for (let i = 0; i < 5; i++) {

```

```

        operate.push(<View style={styles.row}><TouchableOpacity onPress={() => this.buttonPressed(operations[i])} style={styles.btn}><Text style={styles.btnText}>{operations[i]}</Text></TouchableOpacity></View>)
    }

    return (
      <View style={styles.container}>
        <View style={styles.textCal}>
          <Text style={styles.calculateText}>{this.state.calculateText}</Text>
        </View>
        <View style={styles.result}>
          <Text style={styles.resultText}>{this.state.resultText}</Text>
        </View>
        <View style={styles.buttons}>
          <View style={styles.numbers}>
            {rows}
          </View>
          <View style={styles.operation}>
            {operate}
          </View>
        </View>
      </View>
    )
  }
}

const styles = StyleSheet.create({
  container: {
    paddingTop: 30,
    flex: 1
  },
  result: {
    flex: 1,
    backgroundColor: 'white',
    justifyContent: 'center',
    alignItems: 'flex-end'
  },
  resultText: {
    fontSize: 30,
    paddingRight: 10
  },
  textCal: {
    flex: 1,
    backgroundColor: 'white',
    justifyContent: 'center',

```

```
      alignItems: 'flex-end'
    },
    calculateText: {
      fontSize: 30,
      paddingRight: 10
    },
    row: {
      flexDirection: 'row',
      flex: 1
    },
    buttons: {
      flex: 7,
      flexDirection: 'row'
    },
    numbers: {
      flex: 3,
      backgroundColor: '#1d2325'
    },
    operation: {
      flex: 1,
      backgroundColor: '#444e54',
      justifyContent: 'space-around',
      alignItems: 'stretch'
    },
    btn: {
      flex: 1,
      alignItems: 'center',
      justifyContent: 'center'
    },
    btnText: {
      fontSize: 30,
      color: 'white'
    }
  })
})
```