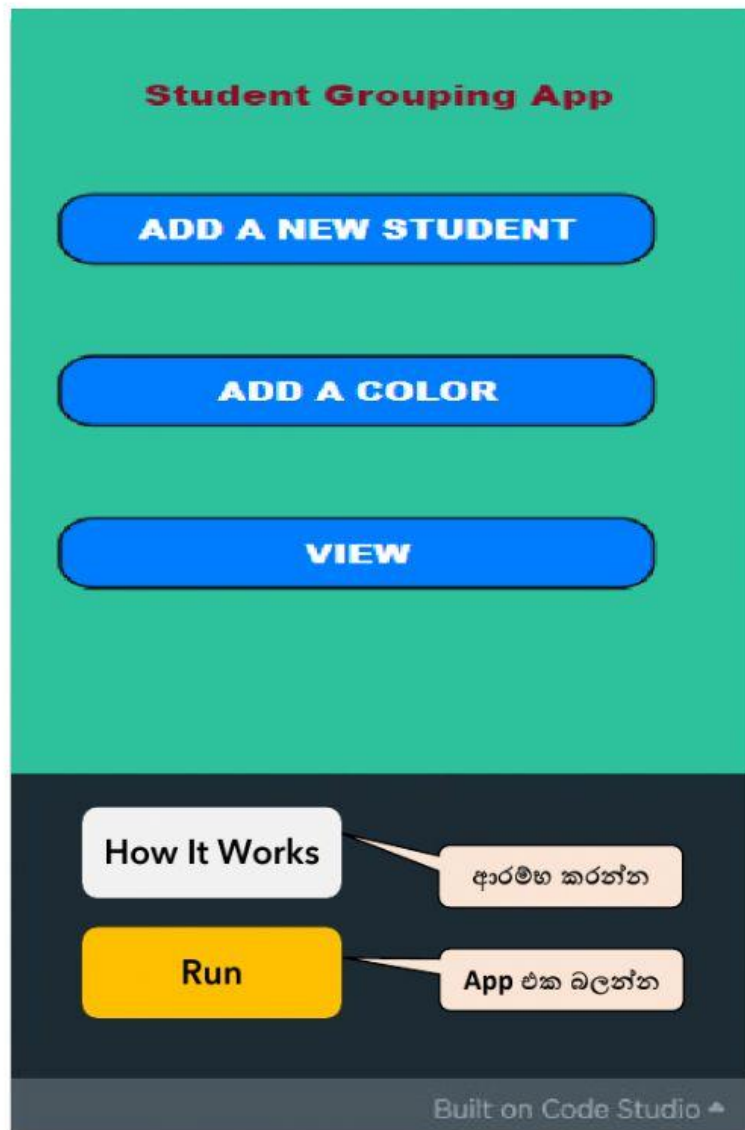


Project 83

83 වන ව්‍යාපෘතිය



Coding School

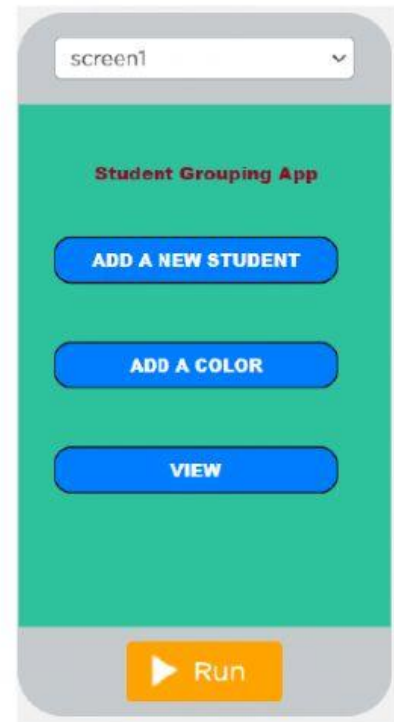


- ❖ Array භාවිතා කර සිසුන් 20 දෙනෙකු කණ්ඩායම් 4 කට බෙදීම සඳහා app එකක් නිර්මාණය කරමු.

සිසුන් 20 ක් දක්වා කණ්ඩායම් 4කට බෙදීම සඳහා මෙම app එක භාවිතා කරයි. සිසුන් 14ක් array එකක් මගින් ලබා දෙන අතර ඉතිරි සිසුන් array එකට add කිරීමේ හැකියාව පවතී.

සිසුන්ගේ ID එක අනුව කණ්ඩායම් 4කට බෙදිය යුතුය.

එක් එක් කණ්ඩායමට වෙන වෙනම screen එක බැගින් ඇති අතර එම screen එකක් තුළ labels 5ක් ඇත. එම labels වල වර්ණය වෙනස් කිරීම සඳහා තවත් array එකක් use කර ඇත.



- ❖ App එකට අදාළ ව design සියල්ල ම ඔබට ලබා දී ඇත.
- ❖ සිසුන්ගේ ID හා name සඳහා වෙන වෙනම arrays දෙකක දෙන්න. තවද color සඳහා ද array එකක් නිර්මාණය කරන්න. Color array එකේ color code වලින් සමන්විත වේ. len හා lenColors ලෙස variables දෙකක් නිර්මාණය කර එවාට පිළිවලින් studentId array එකේ හා colors array එකේ length එක ලබා දෙන්න.

```
var studentsId = [8, 9, 9, 0, 9, 0, 0, 0, 0, 10, 11, 12, 13, 14];
var studentsName = ["Amila", "Amal", "Anura", "Binura", "Bavantha", "Chamera", "Chemika", "Devin", "Darsana", "Denusha", "Dhan", "Ishara", "Sumitha", "Dilki"];
var colors = ['#801616', '#00891b', '#c0cb10', '#1910b6', '#801616', '#00891b', '#c0cb10', '#1910b6', '#801616', '#00891b', '#c0cb10', '#1910b6', '#801616', '#00891b'];
var len = studentsId.length;
var lenColors = colors.length;
```

❖ Screen අතර සම්බන්ධතාව නිර්මාණය කරන ආකාරය බලමු.

```
onEvent(▼"buttonNextA", ▼"click", function(○) {  
    setScreen(▼"screenB");  
});  
onEvent(▼"buttonNextB", ▼"click", function(○) {  
    setScreen(▼"screenC");  
});  
onEvent(▼"buttonNextC", ▼"click", function(○) {  
    setScreen(▼"screenD");  
});  
onEvent(▼"buttonNextD", ▼"click", function(○) {  
    setScreen(▼"screenA");  
});  
onEvent(▼"btnHome", ▼"click", function(○) {  
    setScreen(▼"screen1");  
});
```

❖ “ADD A NEW STUDENT” button click kala විට screen2 වෙත යා යුතු අතර එහි ඇති error text labels show වී ඇත්නම් hide කළ යුතුය.

```
onEvent(▼"btnAddNewStudent", ▼"click", function(○) {  
    hideElement(▼"errorMsg1");  
    hideElement(▼"errorMsg2");  
    setScreen(▼"screen2");  
});
```

❖ “ADD A COLOR” button click කළ විට screen3 වෙත යා යුතුය.

```
onEvent(▼"btnAddColor", ▼"click", function(○) {  
    setScreen(▼"screen3");  
});
```


Screen2

- ❖ Screen2 මගින් new student කෙනකු add කිරීම සඳහා යොදා ගනී. “studentsId” array එකට හා “studentsName” array එකට new value එකක් add කිරීම සිදු කරයි.
- ❖ Back button click කළ විට නැවත screen1 වෙත ගමන් කළ යුතුය.

```
onEvent(▼ "buttonBack1", ▼ "click", function(●) {  
    setScreen(▼ "screen1");  
});
```

(screen2)

- ❖ Add button එක click කළ විට ID එක “studentsId” array එකට හා name එක “studentsName” array එකට add විය යුතුය.
- ❖ “newId” හා “newName” ලෙස variables 2ක් නිර්මාණය කරන්න. ඉන්පසු “studentsId” array එකේ length එක 20 ට සමාන වේ නම් “errorMsg2” label එක show විය යුතුය. මන්ද මෙම app එක උපරිම සිසුන් 20 ක් සඳහා නිර්මාණය කරන බැවිනි.

```
onEvent(▼ "buttonAddNewStudent", ▼ "click", function(●) {  
    var newId;  
    var newName;  
    if(len == 20){  
        showElement(▼ "errorMsg1");  
    }
```

- ❖ Else part එක තුළ “idCheck” ලෙස variable එකක් create කර එහි අගය 1 ලෙස ලබා දෙන්න.
- ❖ “newId” හා “newName” variables දෙකකට පිළිවලත් new student ගේ id එක හා name එක assign කර ගන්න.

- ❖ For loop එකක් නිර්මාණය කර එය තුළ new student ගේ id එක, "studentsId" array එක තුළ කලින් store කර තිබේ දැයි check කරන්න.
- ❖ එසේ තිබේ නම් "idCheck" variable එකෙහි අගය 0 ලෙස ලබා දෙන්න.
- ❖ දැන් if condition එකක් තුළ "idCheck" හි අගය 0 වේ නම් "errorMsg2" label එක show කරන්න.
- ❖ New student ගේ id එක පෙර අප ලබා දුන් array එක තුළ තිබේ දැයි මෙමගින් check කරයි. එසේ තිබේ නම් error message එකක් රතු පාටින් display කිරීම "errorMsg2" label එක මගින් සිදු කරයි.
- ❖ අලුතින් add කරන student පෙර array එක තුළ නොමැති නම් එම new student array එකට add කළ හැකිය. Else part එක තුළ එයට ආදාල coding සිදු කරන්න.

```

else{
    var idCheck = 1;
    newId = getText(▼"inputNewStudenID");
    newName = getText(▼"inputNewStuName");
    for( var i = 0; i < len; i++){
        if( studentsId[i] == newId ){
            idCheck = 0;
        }
    }
    if( idCheck == 0 ){
        showElement(▼"errorMsg2");
    }
    else{
        hideElement(▼"errorMsg2");
        appendItem(studentsId, newId);
        appendItem(studentsName, newName);
        len = studentsId.length;
        setScreen(▼"screen1");
    }
}
};

```

Screen3

- ❖ Screen3 මගින් “colors” array එකට new color එකක් add කිරීම සිදු කරයි.
- ❖ Back button click කළ විට නැවත screen1 වෙත ගමන් කළ යුතුය.

```
onEvent(▼"buttonBack2", ▼"click", function( ) {  
    setScreen(▼"screen1");  
});
```

(screen3)

- ❖ Add button එක click කළ විට new color එක, “colors” array එකේ Index එකට ලබා දෙන ස්ථානයට add විය යුතුය.
- ❖ මෙහිදී new color එක array එකට add වන්නේ අප ලබා දෙන ස්ථානයට වේ. ඒ සඳහා Tool box හි variables වල ඇති code block එක භාවිතා කරනු ලැබේ.

```
insertItem(list, 2, "c");
```

Value එක add විය යුතු index එක

Array එකේ name එක.

අලුත් add කරන value එක

```
insertItem(list, 2, "c");
```

- ❖ “newIndex” හා “newColor” ලෙස variables 2ක් නිර්මාණය කරන්න. ඉන්පසු “colors” array එකේ length එක 20 ට සමාන වේ නම් “errorMsg3” label එක show විය යුතුය. මන්ද මෙම app එක උපරිම සිසුන් 20 ක් සඳහා නිර්මාණය කරන බැවින්ය.
- ❖ Else part එකෙහි “newIndex” හා “newColor” variables දෙකකට පිළිවලත් index එක හා color එකේ name/code එක assign කර ගන්න.
- ❖ ලබා දෙන index එකට අනුව new color එක, “colors” array එකට assign කරන්න.


```

onEvent (▼ "buttonAddNewColor", ▼ "click", function() {
    var newIndex;
    var newColor;
    if (len == 20) {
        showElement (▼ "errorMsg3");
    }
    else {
        newIndex = getText (▼ "inputIndex");
        newColor = getText (▼ "inputColorName");
        insertItem (colors, newIndex, newColor);
        setScreen (▼ "screen1");
    }
});

```

- ❖ View button එක click කළ විට student ගේ id එක අනුව anuwa group 4 කට ළමුන් ව group කළ යුතුය. තවද labels වලට color එකක් set කළ යුතුය.

Student Id එක අනුව group කරන ආකාරය සලකමු.

- ❖ මෙහිදී students id එක අනුව ළමුන් ව group 4 කට බෙදනු ලැබේ. Id එක 4 න් බෙදා reminder එක අනුව group 4 කට බෙදනු ලැබේ.

Reminder = 1 වේ නම් group A ද,
 Reminder = 2 වේ නම් group B ද,
 Reminder = 3 වේ නම් group C හා
 Reminder = 0 වේ නම් group D ලෙස

ළමුන් ව group 4 කට බෙදිය යුතුය.

- ❖ Group A හි ළමුන් screenA වලද, group B හි ළමුන් screenB වලද, group C හි ළමුන් screenC වලද හා group D හි ළමුන් screenD වලද, display කළ යුතුය.

- ❖ screenA, screenB, screenC හා screenD වල labels 5 ක් බැගින් ඇති අතර එවාගේ id පහත පරිදි වේ.

screenA: labelA1, labelA2, labelA3, labelA4 හා labelA5
 screenB: labelB1, labelB2, labelB3, labelB4 හා labelB5
 screenC: labelC1, labelC2, labelC3, labelC4 හා labelC5
 screenD: labelD1, labelD2, labelD3, labelD4 හා labelD5

(screenA)

- ❖ “viewGrouping” function එක create karamu.
- ❖ Labels වල id සඳහා ගැනීම සඳහා a, b, c, d ලෙස variables 4 ක් නිර්මාණය කරන්න. එහි value එක ලෙස 1 ලබා දෙන්න.
- ❖ Function එක තුළ for loop එකක් නිර්මාණය කර “studentsId” array එකේ element එකින් එක check කිරීම සඳහා for loop එකේ i variable එක i = 0 සිට “studentsId” array එකේ length එකේ size දක්වා එකින් එක වැඩි කරන්න. For loop එක තුළ “studentsId[i]” හි අගය 4 න් බෙදා එහි remainder එක ලබා ගන්න.
- ❖ Reminder එකේ අගය අනුව group වලට බෙදන්න.
- ❖ setProperty block එක භාවිතා කර labels වල text එකට values set කරන්න.

```
function viewGrouping() {
  var a = 1;
  var b = 1;
  var c = 1;
  var d = 1;
  for (var i = 0; i < len; i++) {
    if (studentsId[i] % 4 == 1) {
      setProperty("labelA" + a, "text", "ID : " + studentsId[i] + " Name : " + studentsName[i]);
      a++;
    }
    else if (studentsId[i] % 4 == 2) {
      setProperty("labelB" + b, "text", "ID : " + studentsId[i] + " Name : " + studentsName[i]);
      b++;
    }
    else if (studentsId[i] % 4 == 3) {
      setProperty("labelC" + c, "text", "ID : " + studentsId[i] + " Name : " + studentsName[i]);
      c++;
    }
    else {
      setProperty("labelD" + d, "text", "ID : " + studentsId[i] + " Name : " + studentsName[i]);
      d++;
    }
  }
}
```


Array එක අනුව labels color කරන ආකාරය සලකමු.

❖ “colorRow” function එක create karamu

❖ Labels වල id සදා ගැනීම සඳහා a, b, c, d ලෙස variables 4 ක් නිර්මාණය කරන්න.
එහි value එක ලෙස 1 ලබා දෙන්න.

❖ Function එක තුළ for loop එකක් නිර්මාණය කර “colors” array එකේ element එකින් එක check කිරීම සඳහා for loop එකේ i variable එක i = 0 සිට “colors” array එකේ length එකේ size දක්වා එකින් එක වැඩි කරන්න. For loop එක තුළ “colors[i]” හි අගය 4 න් බෙදා එහි remainder එක ලබා ගන්න.

❖ setProperty block එක භාවිතා කර labels වල background-color එකට values set කරන්න.

```
function colorRow() {  
  var a = 1;  
  var b = 1;  
  var c = 1;  
  var d = 1;  
  for (var i = 0; i < lenColors; i++) {  
    if (i % 4 == 1) {  
      setProperty("labelA" + a, "background-color", colors[i]);  
      a++;  
    }  
    else if (i % 4 == 2) {  
      setProperty("labelB" + b, "background-color", colors[i]);  
      b++;  
    }  
    else if (i % 4 == 3) {  
      setProperty("labelC" + c, "background-color", colors[i]);  
      c++;  
    }  
    else {  
      setProperty("labelD" + d, "background-color", colors[i]);  
      d++;  
    }  
  }  
}
```

❖ View button එක click කළ විට “viewGrouping” function එක හා “colorRow” function එකට call කරන්න.

```
onEvent("btnView", "click", function() {  
  viewGrouping();  
  colorRow();  
  setScreen("screenA");  
});
```