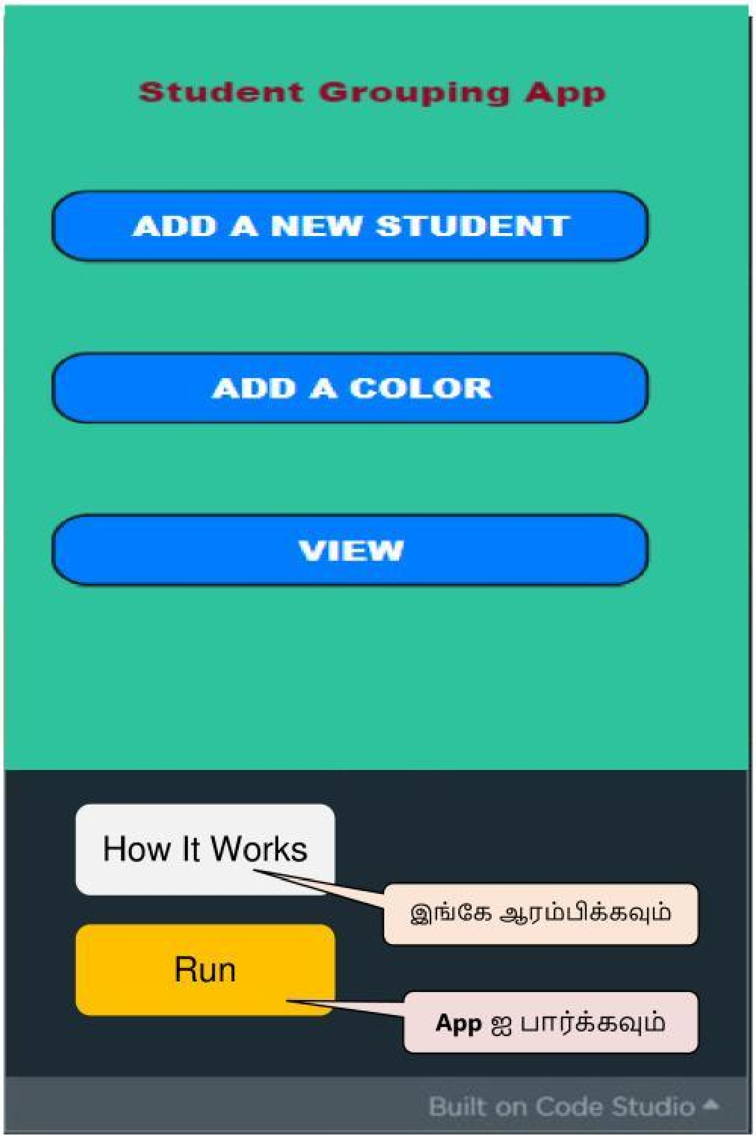
Homework Assignment
வீட்டு வேலை ஒப்படை

DP EDUCATION

Coding School



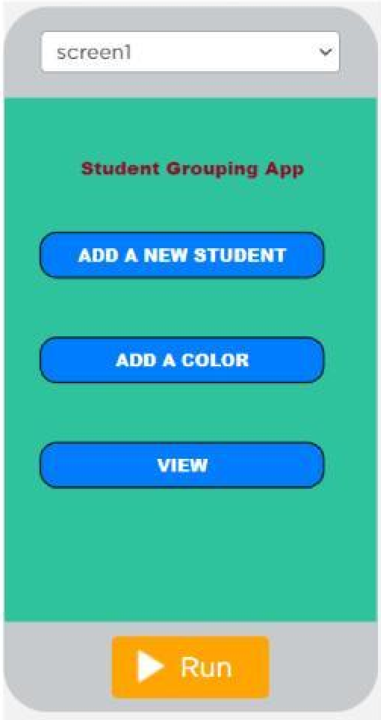
LIVEWORKSHEETS

❖ Array பயன்படுத்தி 20 மாணவர்கள் கொண்ட குழுவை 4 ஆக பிரிப்பதற்கு app ஒன்றை உருவாக்குவோம்.

20 மாணவர்கள் கொண்ட குழுவை 4 ஆகப் பிரிப்பதற்கு இந்த app பயன்படுத்தப்படும். 14 மாணவர்கள் array மூலம் பெற்று தருவதோடு எஞ்சிய மாணவர்களையும் array இற்கு add பண்ண முடியும்

மாணவர்களின் ID இற்கேற்ப 4 குழுக்களுக்கு பிரிக்க வேண்டும்.

ஒவ்வொரு குழுவுக்கும் வேறு வேறாக screen ஒவ்வொன்று இருப்பதோடு அந்த screen இற்கு labels 5 உண்டு. அந்த labels களின் நிறத்தை மாற்றுவதற்கு மேலும் array ஒன்று use பண்ணப்பட்டுள்ளது.



❖ App இற்கேற்ப design முழுவதும் உங்களுக்கும் பெற்றுத்தரப்பட்டுள்ளது.

❖ மாணவர்களின் ID , name இற்கு வேறு வேறு arrays இரண்டு தரவும். மேலும் color இற்கு array ஒன்று உருவாக்கவும். Color array , ஒன்று color code உடன் கூடியதாகும். len மற்றும் lenColors என variables இரண்டு உருவாக்கி அதற்கு முறையாக studentId array மற்றும் colors array இன் length ஐ பெற்றுக் கொடுக்க.

```
var studentsId = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14];
var studentsName = ["Amila", "Amal", "Amara", "Binura", "Bawantha", "Chamara", "Chamika", "Dasun", "Darshana", "Danushka", "Eshan", "Isuru", "Sumudu", "Dilki"];
var colors = ['#801616', '#08891b', '#c0cb10', '#1810b6', '#801616', '#08891b', '#c0cb10', '#1810b6', '#801616', '#08891b', '#c0cb10', '#1810b6', '#801616', '#08891b'];
var len = studentsId.length;
var lenColors = colors.length;
```

LIVEWORKSHEETS

❖ Screen இற்கிடையில் தொடர்பை உருவாக்கும் முறையை பார்ப்போம்.

```
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 செய்யும் போது screen2 இற்கு செல்ல வேண்டியதோடு அங்குள்ள error text labels show

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

ஆகியிருந்தால் hide செய்ய வேண்டும்.

❖ "ADD A COLOR" button click செய்யும்போது screen3 இற்கு செல்ல வேண்டும்.

LIVEWORKSHEETS

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

Screen2

(screen2)

- Screen2 new student ஒருவரை add பண்ணுவதற்காக பயன்படுத்தப்படும். “studentsId” array இற்கு மற்றும் “studentsName” array இற்கு new value add செய்வது நடைபெறும்.

- Back button click செய்யும்போது மீண்டும் screen1 இற்கு நகர வேண்டும்.

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

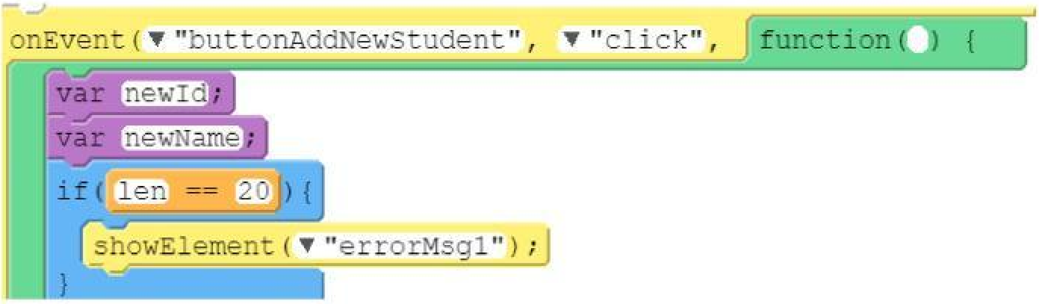
- Add button ஐ click செய்யும் போது ID யானது “studentsId” array இற்கு மற்றும் name “studentsName” array இற்கு add ஆக வேண்டும்.

- “newId” மற்றும் “newName” என variables 2 இரண்டு உருவாக்குக. அதன் பின் “studentsId” array இன் length 20 இற்கு சமனாகும் போது "errorMsg2" label show ஆக வேண்டும்

LIVEWORKSHEETS

ஏனென்றால் அந்த app அதிகூடிய மாணவர்கள் 20 பேருக்காக உருவாக்கப்பட்டதாகும்.

❖ Else part இற்குள் “idCheck” என variable ஒன்றை create செய்து



அதன் பெறுமானம் 1 என பெற்றுக் கொடுக்க.

❖ “newId” மற்றும் “newName” variables இரண்டிற்கு ஒழுங்கு முறையாக new student இன் id மற்றும் name ஐ assign பண்ணிக் கொள்க.

❖ For loop ஒன்றை உருவாக்கி அதற்குள் new student இன் id ஐ, “studentsId” array இற்குள் முதலில் store பண்ணப்பட்டுள்ளதா என check செய்யவும்.

❖ அவ்வாறிருந்தால் “idCheck” variable இன் பெறுமானம் 0 என பெற்றுக் கொடுக்க.

❖ தற்போது if condition இற்குள் “idCheck” இன் பெறுமானம் 0 ஆக இருந்தால் "errorMsg2" label ஐ show செய்யவும்.

❖ New student இன் id முதலில் நாம் பெற்றுக் கொடுத்த array இற்குள் இருக்கின்றதா என்று இதன் மூலம் check பண்ணப்படும். அவ்வாறு இருந்தால் error massage ஐ சிவப்பு நிறத்தால் display செய்வது "errorMsg2" label மூலம் நடைபெறும்.

LIVEWORKSHEETS

- புதிதாக add பண்ணிய student முன் இருந்த array இற்குள் இல்லை என்றால் அந்த new student array அற்கு add பண்ண முடியும். Else part இற்குள் அதற்குரிய coding செய்யவும்.

```
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");
  }
}
});
```

LIVEWORKSHEETS

Screen3

- Screen3 மூலம் “colors” array இற்கு new color ஒன்றை add பண்ணுதல் நடைபெறும்.

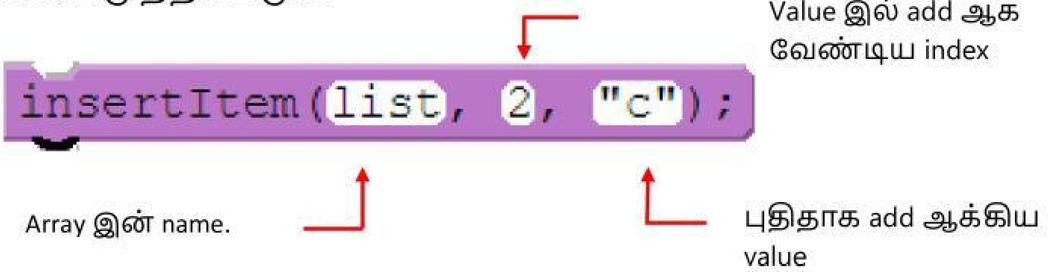
- Back button click செய்யும்போது மீண்டும் screen1 இற்கு நகர வேண்டும்.

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

(screen3)

- Add button ஐ click செய்யும் போது new color, “colors” array இன் Index இற்கு பெற்றுத் தரும் இடத்திற்கு add ஆக வேண்டும்.

- இங்கு new color array இற்கு add ஆவது நாங்கள் பெற்றுத்தரும் `insertItem(list, 2, "c");` இடத்திற்காகும் Tool box இன் variables இலுள்ள code block பயன்படுத்தப்படும்



- “newIndex” மற்றும் “newColor” என variables இரண்டு உருவாக்குக. அதன் பின் “colors” array இன் length 20 இற்கு சமனானால் "errorMsg3" label. show ஆக வேண்டும். இந்த app ஆனது அதிகூடிய 20 மாணவர்களுக்காக உருவாக்க பட்டதாகும்.

LIVEWORKSHEETS

❖ Else part இன் “newIndex” மற்றும் “newColor” variables இரண்டிற்கு முறையாக index மற்றும் color இன் name/code ஐ assign செய்து கொள்க.

❖ பெற்றுத் தரும் index இற்கேற்ப new color, “colors” array இற்கு assign பண்ணுக.

```
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 ஐ click ஐ செய்யும் போது student இன் id இற்கு ஏற்ப group 4 விற்கு பிள்ளைகளை group செய்ய வேண்டும். மேலும் labels களுக்கு color ஒன்று set செய்ய வேண்டும்.

Student Id இற்கேற்ப group செய்யும் முறையை பார்ப்போம்.

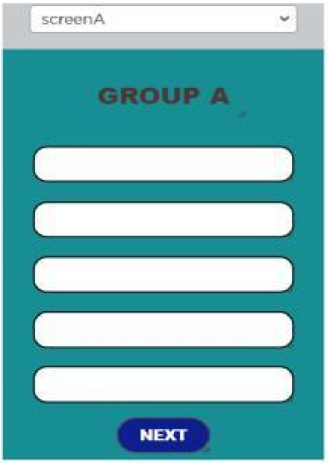
❖ இங்கு students id இற்கேற்ப பிள்ளைகளை group 4 பிரிக்க வேண்டும். Id ஐ 4 ஆல் பிரித்து reminder இற்கேற்ப group 4 ஆக பிரிக்க வேண்டும்.

Reminder = 1 ஆயின் group A யும்,
Reminder = 2 ஆயின் group B யும்,
Reminder = 3 ஆயின் group C மற்றும்
Reminder = 0 ஆயின் group D என

பிள்ளைகளை 4 group களாக பிரிக்க வேண்டும்.

LIVEWORKSHEETS

❖ Group A பிள்ளைகள் screenA இற்கும், group B பிள்ளைகள் screenB இற்கும், group C இன் பிள்ளைகள் screenC இற்கும் மற்றும் group D பிள்ளைகள் screenD இற்கும், display செய்ய வேண்டும்.



(screenA)

❖ screenA, screenB, screenC மற்றும் screenD களில் labels 5 இருப்பதால் அவற்றின் id பின்வருமாறு அமையும்.

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

❖ "viewGrouping" function ஐ create செய்யவும்.

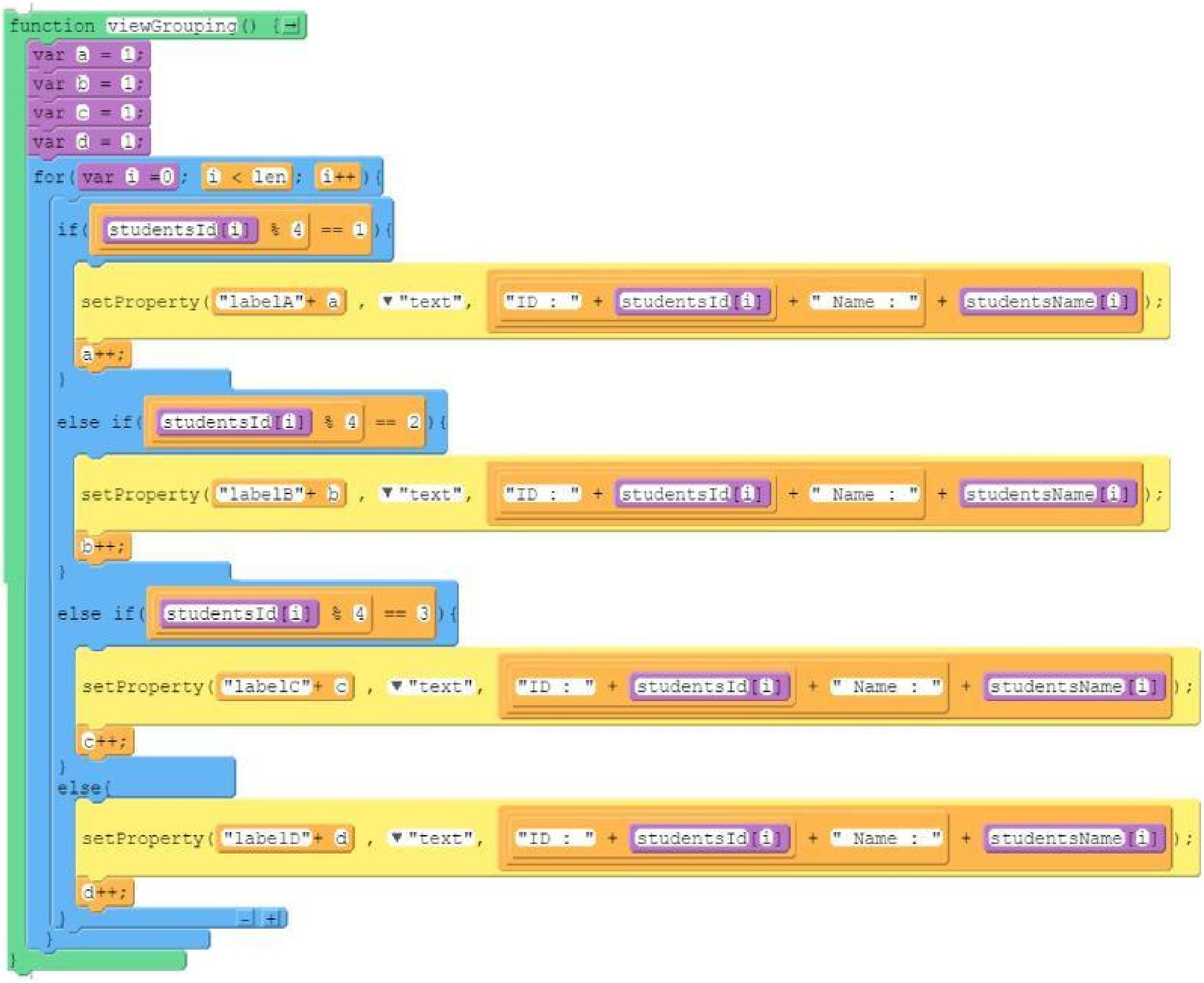
❖ Labels களின் id ஐ உருவாக்குவதற்கு a, b, c, d என variables 4 உருவாக்குக. அதன் value ஐ 1 என பெற்றுக் கொடுக்க.

❖ Function இற்குள் for loop ஒன்றை உருவாக்கி "studentsId" array இன் element ஒன்றையொன்று check பண்ணுவதற்கு for loop என் i variable ஐ i = 0 இலிருந்து "studentsId" array இன் length இன் size வரை ஒன்றையொன்று அதிகரிக்க. For loop இற்குள் "studentsId[i]" இன் பெறுமானத்தை 4 ஆல் பிரித்து அதன் reminder ஐ பெற்றுக் கொடுக்க.

❖ Reminder இன் பெறுமானத்துக்கேற்ப group பிரிக்க.

❖ setProperty block ஐ பாவித்து labels களின் text இற்கு values set செய்யவும்.

LIVEWORKSHEETS



Array இற்கேற்ப labels color பண்ணும் முறையை பார்ப்போம்.

- “colorRow” function ஐ create செய்வோம்

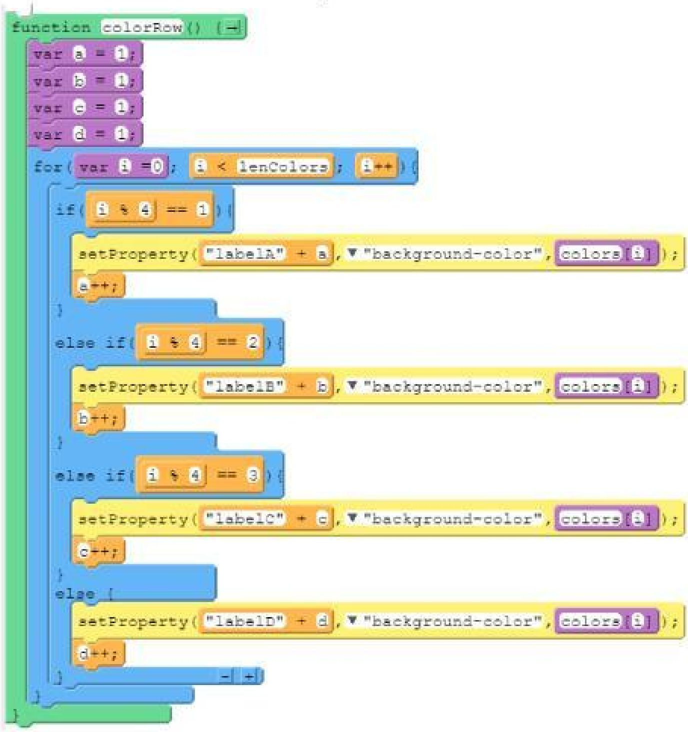
- Labels களின் id உருவாக்க a, b, c, d என variables 4 உருவாக்குக. அதன் value 1 எனப் பெற்றுக் கொடுக்க.

- Function இற்குள் for loop ஒன்றை உருவாக்குக “colors” array இன் element ஒன்றையொன்று check செய்வதற்கு for loop இன் i variable i = 0 இலிருந்து “colors” array இன் length இன் size வரை ஒன்றையொன்று அதிகரிக்கவும். For loop இற்குள் “colors[i]”

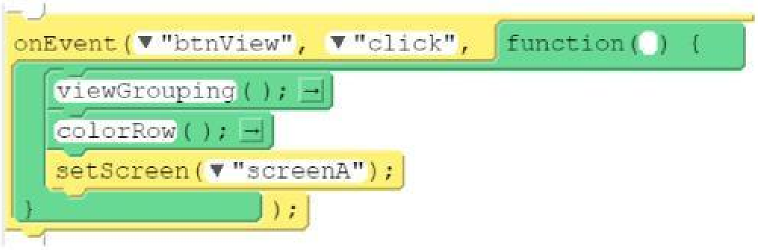
LIVEWORKSHEETS

இன் பெறுமானத்தை 4 ஆல் பிரித்து அதன் reminder பெற்றுக் கொடுக்க.

❖ setProperty block ஐ பயன்படுத்தி labels களின் background-color இற்கு values set செய்யவும்.



❖ View button ஐ click செய்து “viewGrouping” function மற்றும் “colorRow” function இற்கு call செய்யவும்.



LIVEWORKSHEETS