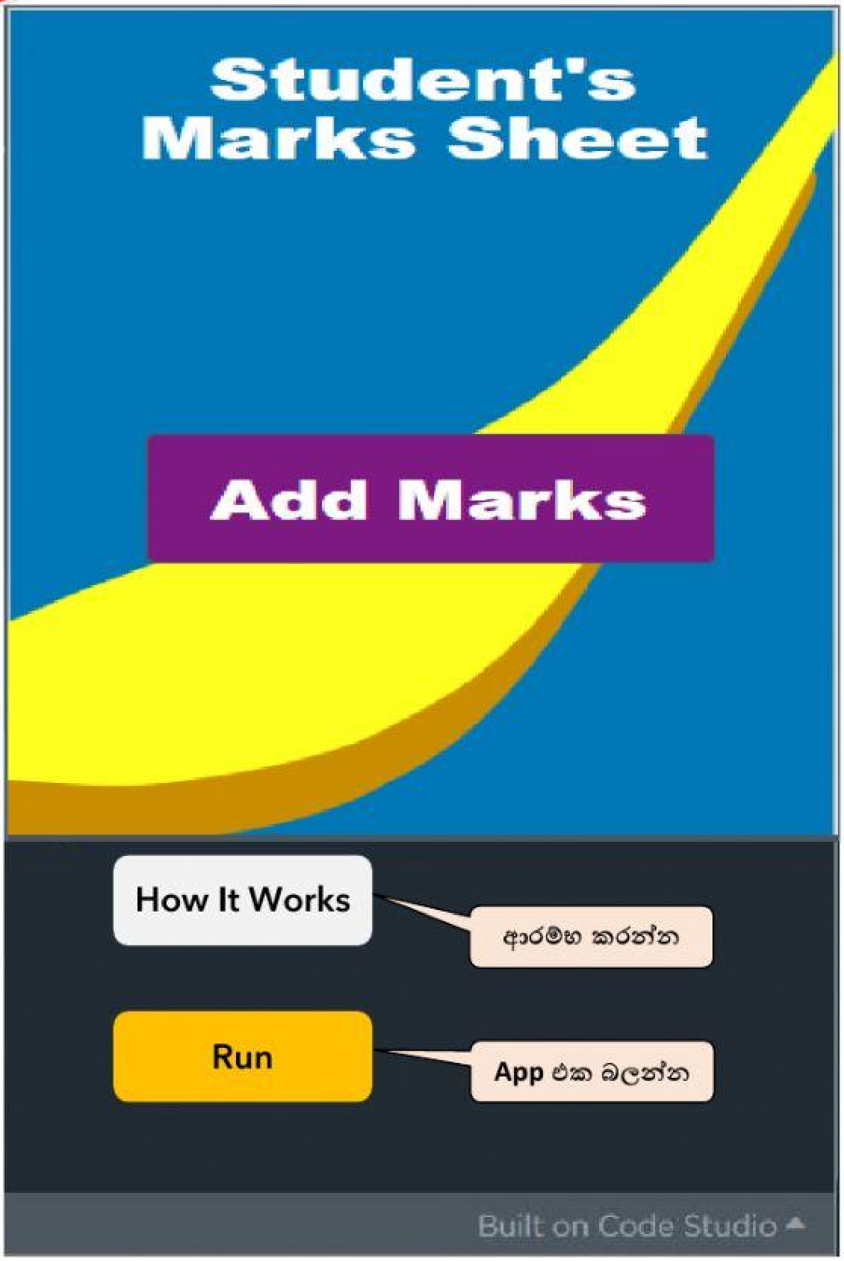
Project 91
91 වන ව්‍යාපෘතිය

DP EDUCATION

Coding School



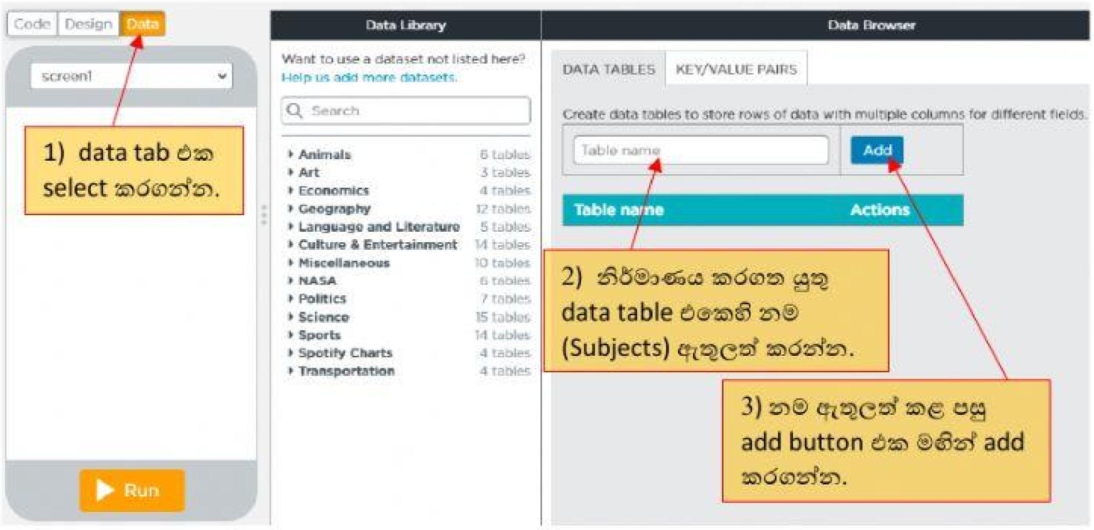
LIVEWORKSHEETS

❖ සිසුන්ගේ විෂයයන් සඳහා ලකුණු ඇතුළත් කිරීමට Report card එකක් නිර්මාණය කරමු.

❖ මෙහිදී data table එකක් තුළට ඇතුළත් කරනා ළමුන් කිහිපදෙනෙකුගේ නම, වසර, විෂය හා එම විෂයට අදාළ ලකුණු APP එක මාර්ගයෙන් update කිරීම හෝ data table එක තුළ ඇති නමක් සඳහා අළුතින් ලකුණු ඇතුළත් කිරීම සිදු කරයි.

❖ මේ සඳහා පළමුව data table එක නිර්මාණය කරගන්නා ආකාරය බලමු.

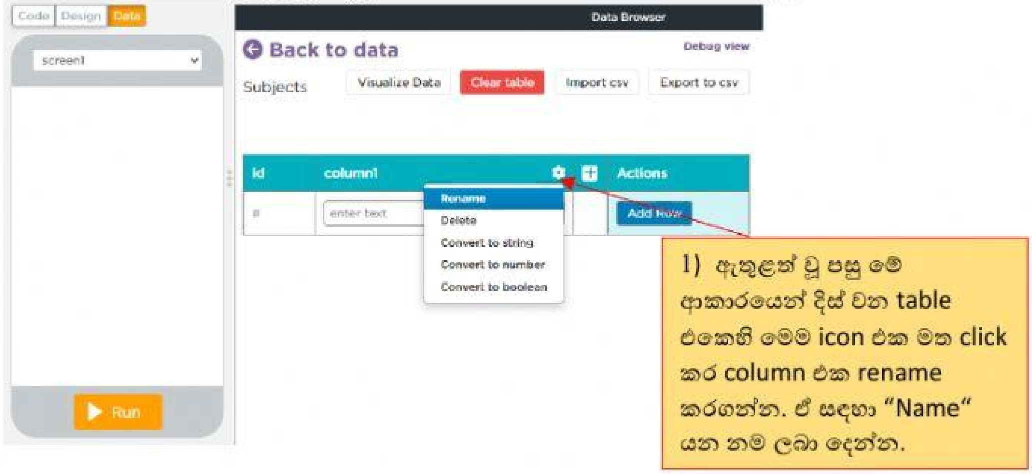
පියවර 1: data table එක සඳහා නම ලබාදීමේදී පළමුව විෂයයන් ඇතුළත් table එක නිර්මාණය කරගමු. ඒ සඳහා "Subjects" ලෙස නම ලබා දෙන්න.



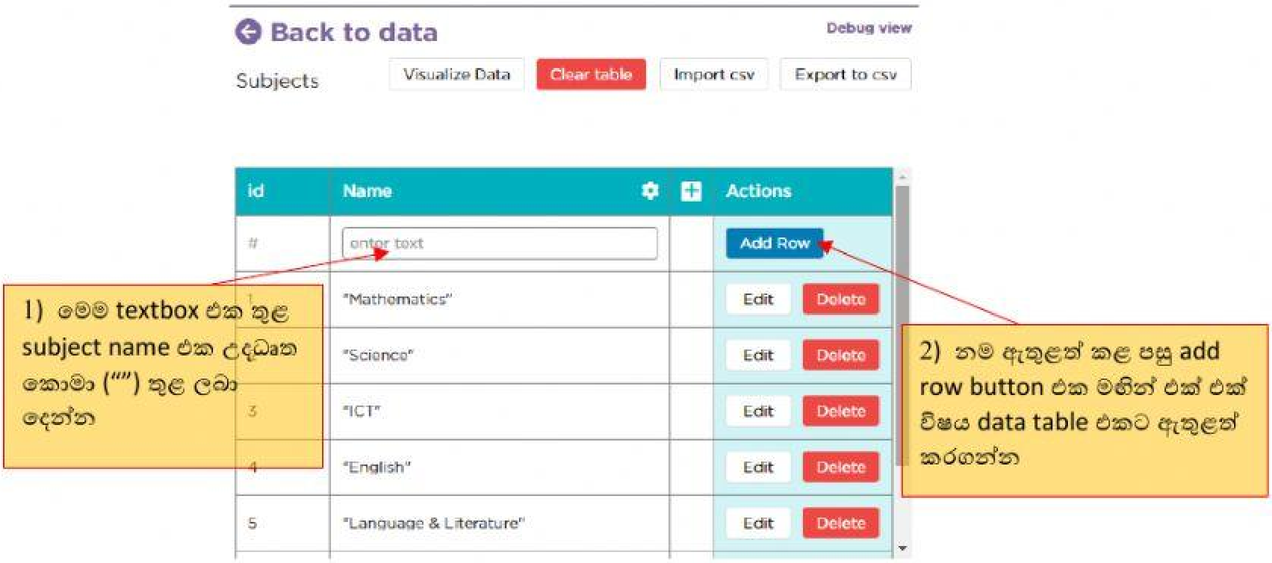
එවිට පහත පරිදි add කරගත් data table එක පහළින් දර්ශණය වේ. එහි නම මත click කිරීමෙන් table එක තුළට ඇතුල් විය හැකිය.

LIVEWORKSHEETS

පියවර 2: data table එකට ඇතුල්වූ පසු එහි columns නිර්මාණය කරගමු.



පියවර 3: column add කරගත් පසු table එකට rows add කරමින් data add කරගන්නා ආකාරය බලමු.

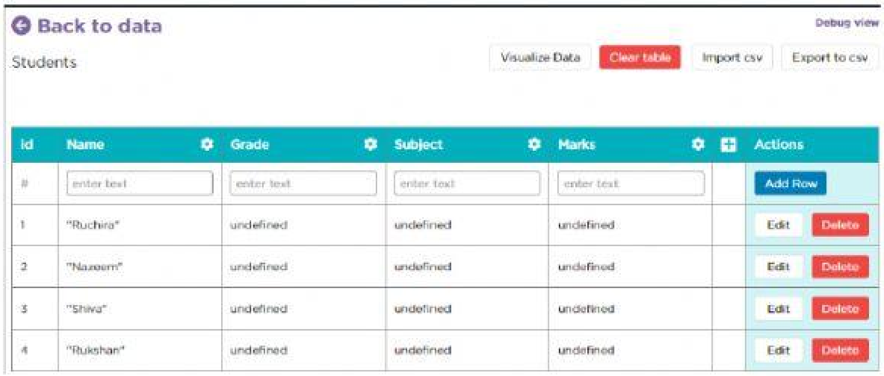


* ඔබට යම් record එකක් add කිරීමේදී යම් වැරදීමක් සිදු වුවහොත් edit button එක මගින් එය edit කිරීමට හෝ delete button එක මගින් එය delete කරගත හැකිය.

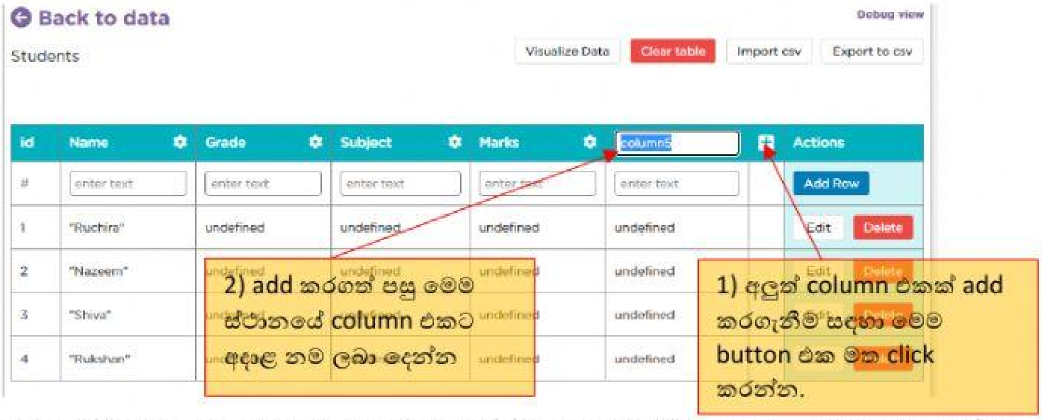
සියලු records add කරගත් පසු back to data මගින් back විය හැකිය. නැවත තවත් records add කරගැනීමට අවශ්‍ය නම් ඉහත ආකාරයටම table name එක මත click කර නැවත table එක view කරගත හැකිය.

LIVEWORKSHEETS

❖ මේ ආකාරයටම තවත් අලුත් table එකක් “Students” ලෙස පහත පරිදී නිර්මාණය කරගන්න.



මෙය නිර්මාණයේදී column 4ක් භාවිතා කර ඇත. Name, Grade, Subject, Marks ලෙස. එසේ අලුත් column add කරගැනීමට පහත පියවර අනුගමනය කරන්න.



❖ App එක නිර්මාණයට අවශ්‍ය වන data table දෙකම නිර්මාණය කරගත අවසන් වූ පසු app එක නිර්මාණය කිරීම ආරම්භ කරමු. ඔබ සඳහා අවශ්‍ය වන screens ලබා දී ඇත. Code කිරීමට පටන්ගනිමු.

❖ පළමුව search box එක තුළට input කරනා නම store කර ගැනීමට හා ඇතුළත් කළ නම data table එක තුළ අන්තර්ගත වේ දැයි පරීක්ෂා කරගැනීමට පහත පරිදී variable දෙකක් නිර්මාණය කරගන්න.

```
var stdName = "";
var register = false;
```

LIVEWORKSHEETS

❖ ඉන්පසුව Add Marks button එක click කළ විට ලකුණු add කරන screen එකට මාරු වීමට නිර්මාණය කරන්න.

❖ මෙහි subject නැමැති drop down එක සඳහා options ලබාදීමට ඉහතදී නිර්මාණය කරන ලද subjects නැමැති data table එකෙහි ඇති subjects භාවිතා කරමු.

```
readRecords("Subjects", {}, function(records) {
  var options = [];
  for (var i = 0; i < records.length; i++) {
    options.push(records[i].Name);
  }
  setProperty("drpSubjects", "options", options);
});
```

- readRecord block එක මඟින් Subjects ලෙස අප නිර්මාණය කරගත් data table එක access කරයි.
- මෙහිදී option ලෙස නම් කර array එකක් නිර්මාණය කරගන්න. එය data table එකෙහි දත්ත ගබඩා කිරීමට භාවිතා කරයි.
- For loop එක මඟින් සියලුම records හරහා ගොස් එක් එක් record එකෙහි Name column එකෙහි ඇති අගය ඉහත නිර්මාණය කරගත් "options" array එක තුළ ගබඩා කරයි.

```
options.push(records[i].Name);
```

- මේ සඳහා arrays වල ඇති push නැමැති function එක භාවිතා කරයි. එය මඟින් array එකකට නව අගයක් ඇතුළත් කරගත හැකිය.

```
options.push
```

- ඉන්පසු එසේ ඇතුළත් කරගත් options array එක dropdown එක සඳහා ලබා දීමට මෙම block එක භාවිතා කර ඇත.

```
setProperty("drpSubjects", "options", options);
```

LIVEWORKSHEETS

- ළමයකුගේ විස්තර update කිරීම සඳහා ප්‍රථමයෙන් ළමයාගේ නම search box එක තුළ type කර search button එක click කර එම නම students table එක තුළ ඇතුළත් දැයි බැලිය යුතුය. ඒ සඳහා පහත පරිදි code කරන්න.

```
onEvent("btnSearchAdd", "click", function( ) {
  stdName = getText("txtSearchAdd");
  readRecords("Students", {}, function(records) {
    for (var i =0; i < records.length; i++) {
      var recodsLc = records[i].Name;
      if(recodsLc == stdName)
      {
        register = true;
        setText("txtName", records[i].Name);
        setText("txtGrade", records[i].Grade);
        setText("drpSubjects", records[i].Subject);
        setText("txtMarksAdd", records[i].Marks);
      }
    }
    if(register == false)
    {
      setProperty("lblSubmit", "text-color", "red");
      setText("lblSubmit", "Record didn not found !");
      setTimeout(function( ) {
        hideElement("lblSubmit");
      }, 1000);
    }
  });
});
```

Search box එක තුළට input කරනා අගය ඉහත නිර්මාණය කරගත් variable එකට දමාගැනීම

Student table එකෙහි ඇති records read කර එම එක් එක් record එකෙහි Name column එකේ අගය for loop එකේ එක් එක එක් loop වීමකදී recordLc නැමැති variable එකට දමාගනී.

ඉන්පසුව එම variable එකෙහි අගය search box එක තුළට ඇතුළත් කළ අගයට සමාන දැයි if block එක මඟින් check කරයි. එසේ සමාන වන record එකක් ඇත් නම් එහි එක් එක් column එකේ values txtName, txtGrade, drpSubjects, txtMarksAdd නැමැති fields වලට set කරයි.

ඉහත check කළ condition එක false වන්නේ නම්, එනම් input කළ name එක data table එකෙහි ඇතුළත් නොවේ නම් not found message එකක් දර්ශණය වී තත්පරයකට පසු එය නැති වී යාමට මෙම codes භාවිතා කරන්න. “lblSubmit” ලෙස label එකක් මේ සඳහා design කර ඇත. එය භාවිතා කරන්න

- ❖ මෙම ආකාරයට code කරගත් පසු data table එක තුළ search කරන නම තිබේ නම් text field තුළ එම record එකෙහි අගයන් පෙන්වයි.
- ❖ ඉන්පසුව එසේ search කරගත් record එකක් update කිරීම සඳහා ලකුණු add කිරීමේ දී plus button එක හා minus button එකේ ක්‍රියාකාරීත්වය සඳහා පහත පරිදි code කරන්න.

```
onEvent("btnPlus", "click", function( ) {
  var marks = getNumber("txtMarksAdd")
  setProperty("txtMarksAdd", "value", marks+1);
});
```

LIVEWORKSHEETS

Marks ලෙස අලුත් variable එකක් නිර්මාණය කරගෙන "txtMarksAdd" නැමැති ලකුණු add කරනා text box එකෙහි අගය getNumber මගින් ලබාගන store කරගන්න. getNumber මගින් එම textbox එකෙහි අගය number එකක් ලෙස ලබාගන variable එකෙහි store කරයි.

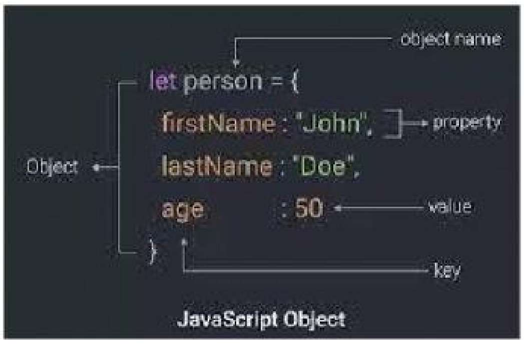
Plus button එක click කළ විට textbox එකෙහි අගය එකකින් වැඩි වීම සඳහා මෙසේ code කර ඇත.

```
setProperty("txtMarksAdd", "value", marks+1);
```

- ❖ මෙ ආකාරයෙන්ම minus button එක මත click කළ විට text box එකෙහි අගයෙන් එකක් අඩු වීමට නිර්මාණය කරන්න.
- ❖ ඉන්පසු search කර update කරගත් record එකක් submit කර data table එකේ save කරගන්නා ආකාරය බලමු.
- ❖ Submit නැමැති button එක click කළ විට එසේ update වීමට onEventClick block එක පහත පරිදි භාවිතා කරන්න.

```
onEvent("btnSubmit", "click", function( ) {
  var name = getText("txtName");
  var grade = getText("txtGrade");
  var subject = getText("drpSubjects");
  var marks = getText("txtMarksAdd");
```

- ❖ name,grade,subject හා marks ලෙස variable 4ක් create කර getText මගින් එක් එක textbox එකෙහි අගයන් variable වල store කරගනී.
- ❖ ඉන්පසු search කරනා නම data table එක තුළ ඇති නම් ඉහතදී register variable එක true කරගනී. මෙහිදී එය true නම් එම නමට අදාළ record එක readRecord block එක මගින් පහත පරිදි ලබාගෙන එහි අගයන් temp නැමැති object එකක ගබඩා කරගනී.
- ❖ Object එකක් යනු විවිධ වර්ග වල data type සහිත properties අඩංගු entity එකකි. පහත දැක්වෙන්නේ එසේ person ලෙස create කරන ලද object එකකි. Object එකක් තුළ එක් property එකක් සඳහා key එකක් හා value එකක් අඩංගු වේ. මෙසේ properties කිහිපයකින් object එකක් සමන්විත වේ.



LIVEWORKSHEETS

```
readRecords("Students", {Name:stdName}, function(records) {
  var temp = {};
  temp.id = (records[0].id);
  temp.Name = name;
  temp.Grade = grade;
  temp.Subject = subject;
  temp.Marks = marks;
```

search box එක තුළ enter කරන නමට අදාළ නම සහිත record එක data table එක තුළින් ලබා ගැනීමට මේ ආකාරයට ලබා දෙන්න. (Name:stdName) Name යනු Students table එකේ name column එකයි.

temp ලෙස create කරගත් object එක. එහි id, Name, Grade, Subject, Marks ලෙස properties අඩංගු වේ. මෙම object එකේ properties නම් කිරීමේ දී data table එකෙහි column names ම ලබා දෙන්න.

- මෙහිදී text box වලට enter කරන ලද අගයන් object එකෙහි properties තුළ store කරයි. මෙම object එක data table එක update වීම සඳහා පහතදී pass කිරීම සිදු කරයි.

```
updateRecord("Students", temp, function(record, success) {
  if(success)
  {
    setText("lblSubmit", "successfully updated the record ! ");
    setTimeout(function() {
      hideElement("lblSubmit");
    }, 1000);
  }
  else{
    setProperty("lblSubmit", "text-color", "red");
    setText("lblSubmit", "Record didn't updated !");
  }
});
```

ඉහත නිර්මාණය කරගත් object එක updateRecord එක සඳහා ලබා දීම

- Update වීම නිවැරදිව සිදු වූයේ නම් function එක තුළට success parameter එක pass වීම සිදු වේ. එය if block එක තුළදී check කර success නම්

```
setText("lblSubmit", "successfully updated the record ! ");
setTimeout(function() {
  hideElement("lblSubmit");
}, 1000);
```

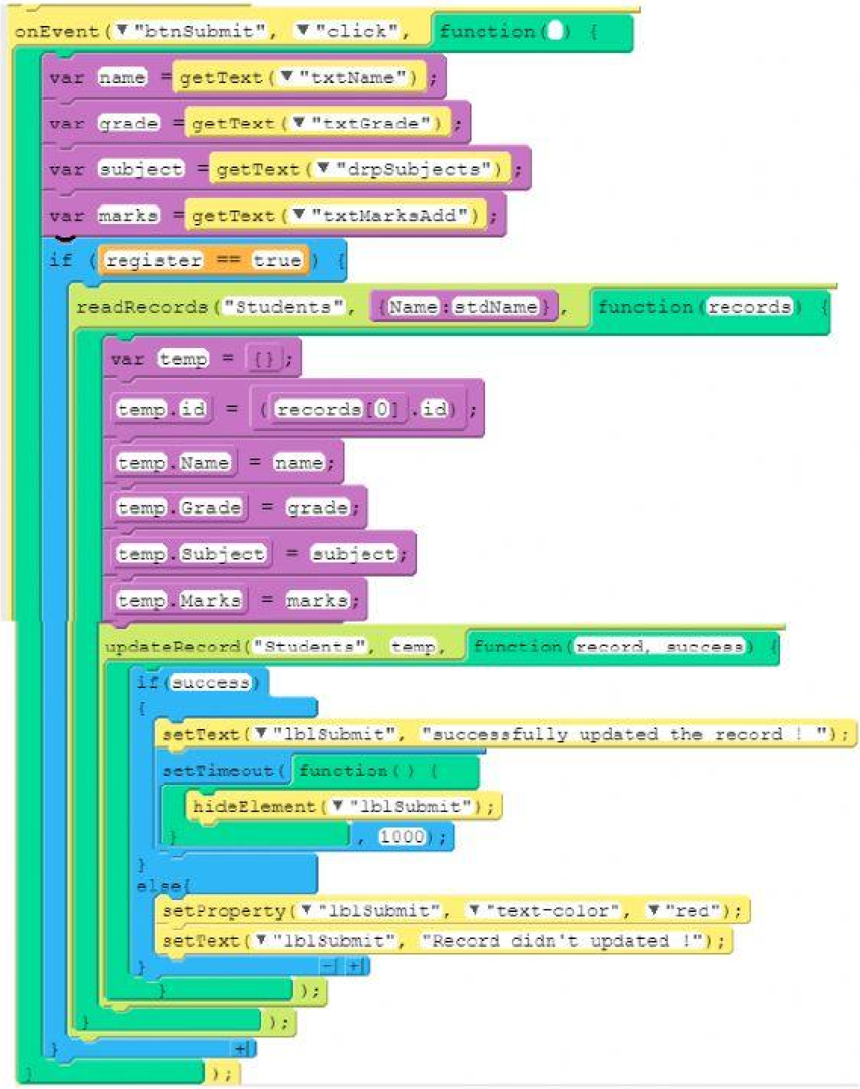
- මෙම block එක display වී තත්පරයකින් නැති වී යයි.

  Success නොවේ නම් පහත blocks display වේ.

```
setProperty("lblSubmit", "text-color", "red");
setText("lblSubmit", "Record didn't updated !");
```

- Submit button එක click කළ විට සිදු විය යුතු දේ සඳහා අදාළ සම්පූර්ණ code එක පහත දැක්වේ.

LIVEWORKSHEETS



❖ මේ ආකාරයෙන් app එක නිර්මාණය කර app එක මඟින් record update කර නැවත data table එකට ගොස් එය update වී ඇත් දැයි බලන්න. නිවැරදිව app එක නිර්මාණය කර ඇත් නම් එසේ update කරන විට data table එක update විය යුතුය.

LIVEWORKSHEETS