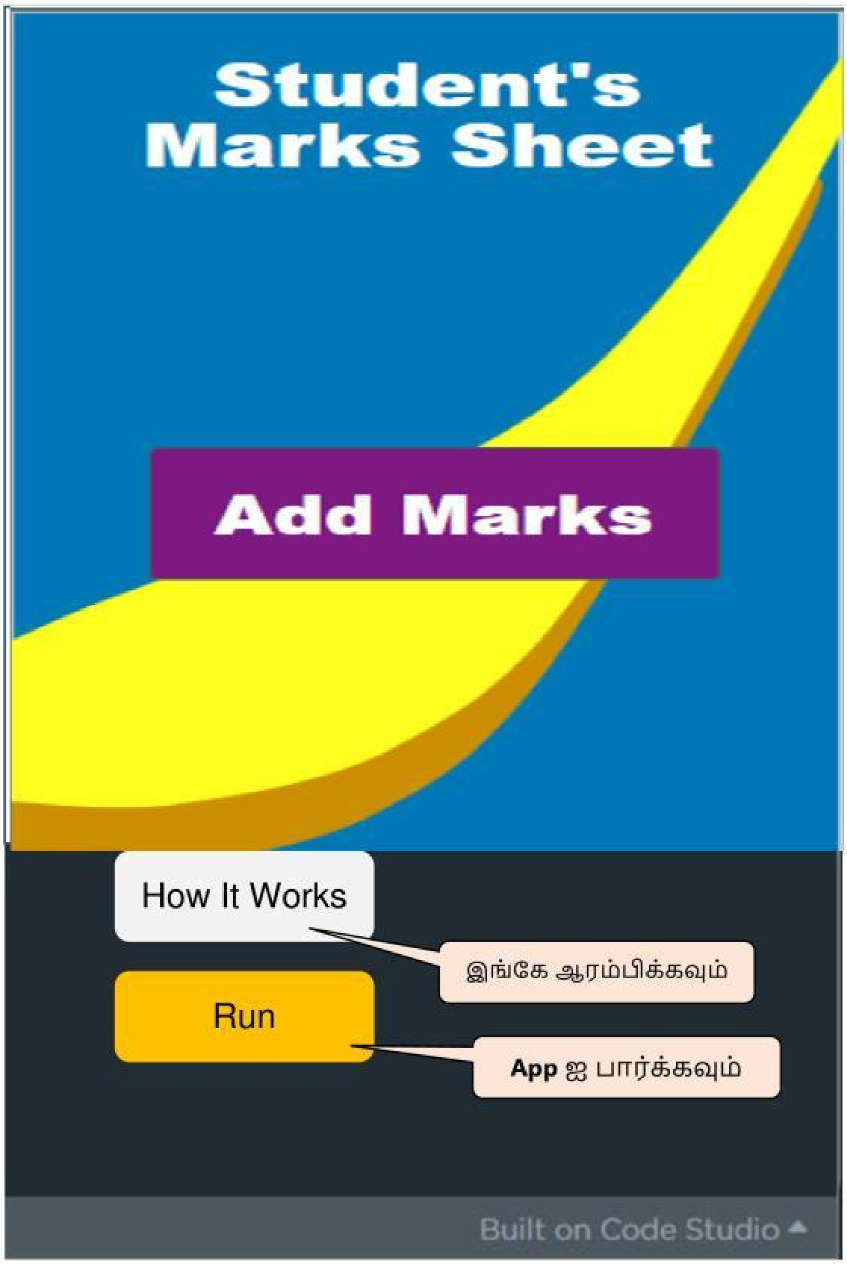
Homework Assignment
வீட்டு வேலை ஒப்படை

DP EDUCATION

Coding School

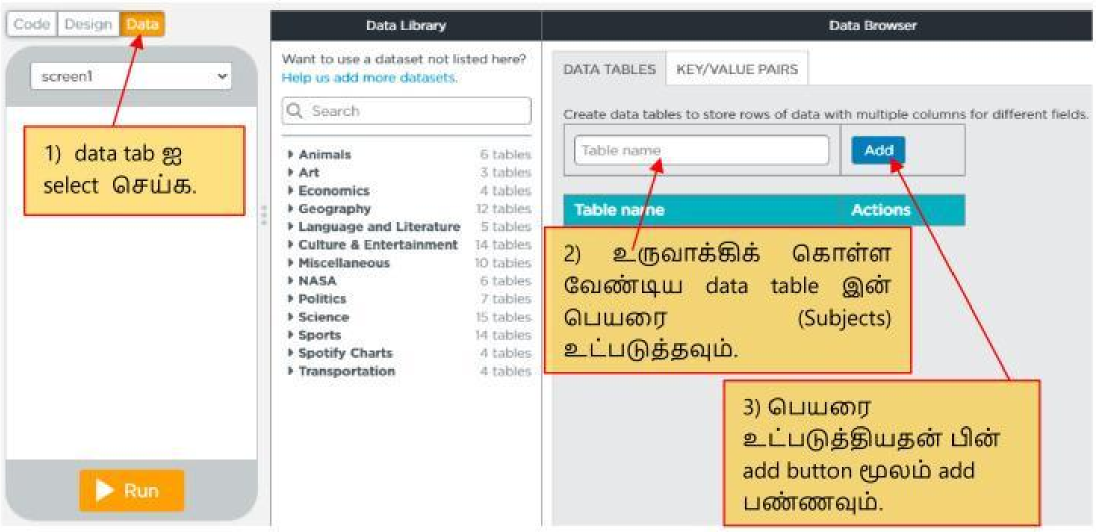


LIVEWORKSHEETS

- மாணவர்களின் பாடங்களின் புள்ளிகளை உட்படுத்துவதற்கு Report card ஒன்றை உருவாக்குவோம்.
- இங்கு data table இற்குள் உட்படுத்திய சில மாணவர்களின் பெயர், வகுப்பு, பாடம் மற்றும் அப்பாடத்துக்குரிய புள்ளிகளை APP மூலம் update செய்வதற்கு அல்லது data table இற்குள் உள்ள பெயர்களிற்கு புதிதாக புள்ளிகளை உட்படுத்தல் நடைபெறும்.

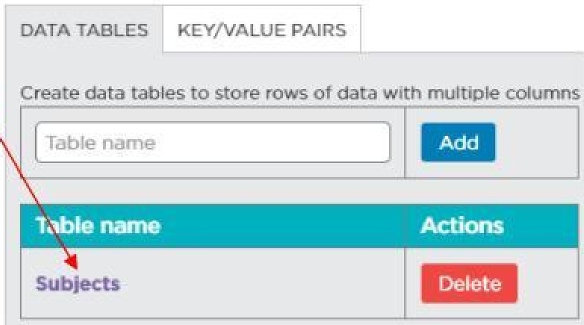
- அதற்காக முதலில் data table உருவாக்கும் முறையை பார்ப்போம்.

படிமுறை 1: data table இற்கு பெயர் பெற்று கொடுக்கப்படும் போது முதலில் பாடங்களை உட்படுத்திய table ஐ உருவாக்குவோம். அதற்காக "Subjects" என பெயர் பெற்றுக் கொடுக்க.



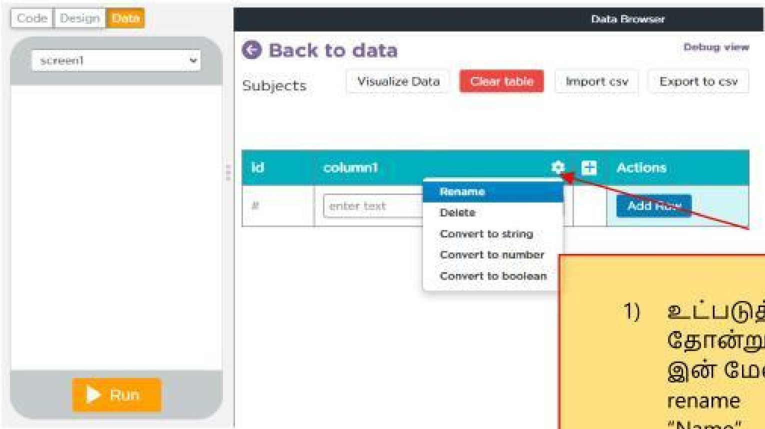
LIVEWORKSHEETS

அப்போது பின்வருமாறு add பண்ணிய data table இற்கு கீழ் காட்சிப்படுத்தப்பட்டுள்ளது. அங்கு பெயரின் மேல் click செய்வதன் மூலம் table இற்கு உட்பட முடியும்.



படிமுறை 2: data table இற்குள் உட்படுத்திய பின் அங்கு

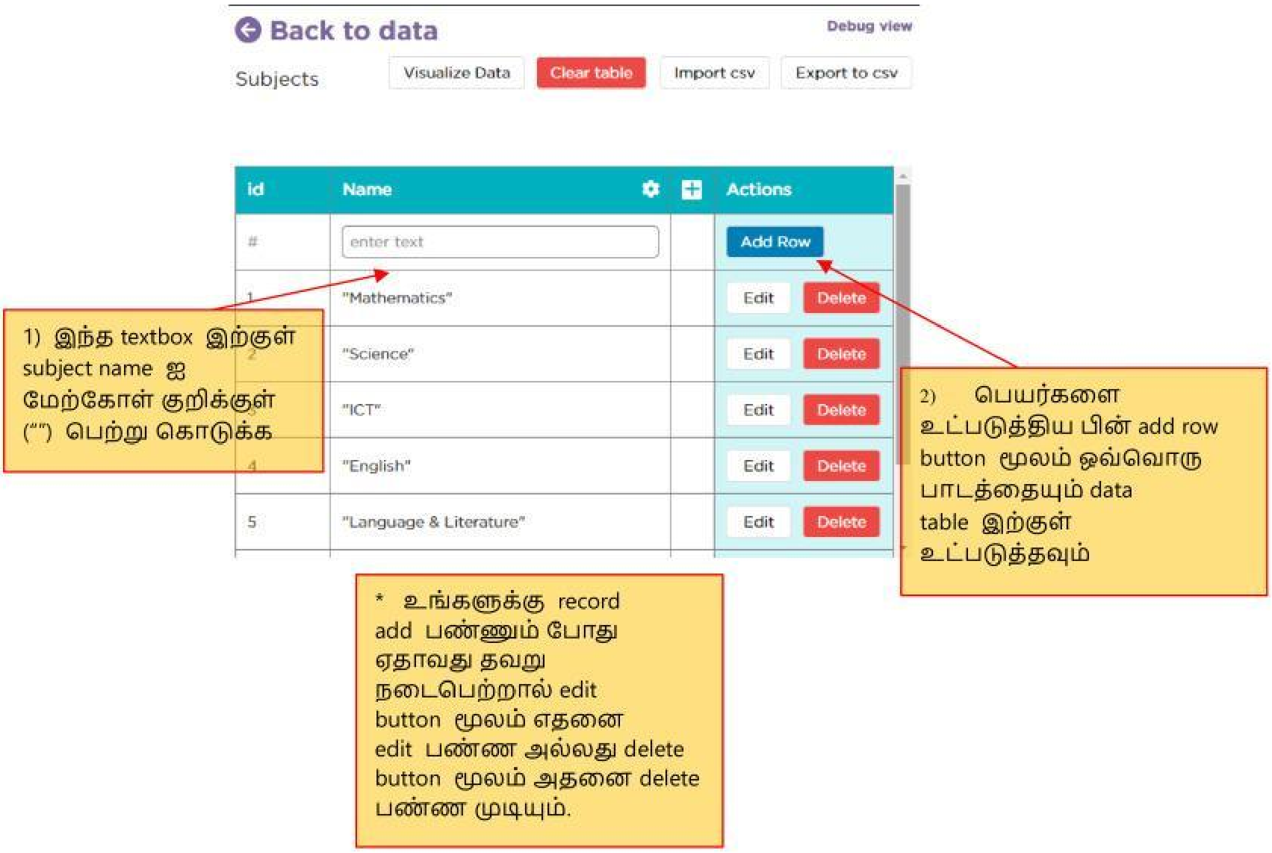
columns உருவாக்குவோம்.



1) உட்படுத்திய பின் இவ்வாறு தோன்றும் table இன் இந்த icon இன் மேல் click செய்து column ஐ rename செய்க. அதற்காக "Name" என்ற பெயரைப் பெற்றுக் கொடுக்க.

LIVEWORKSHEETS

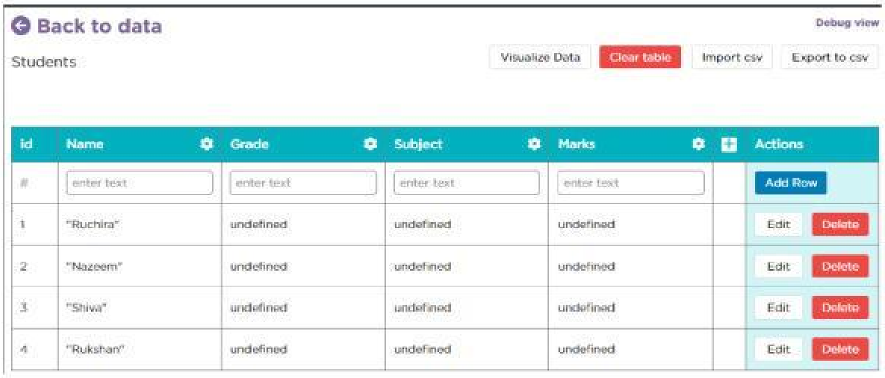
படிமுறை 3: column add பண்ணிய பின் table இற்கு rows add பண்ணி data add பண்ணும் முறையை பார்ப்போம்.



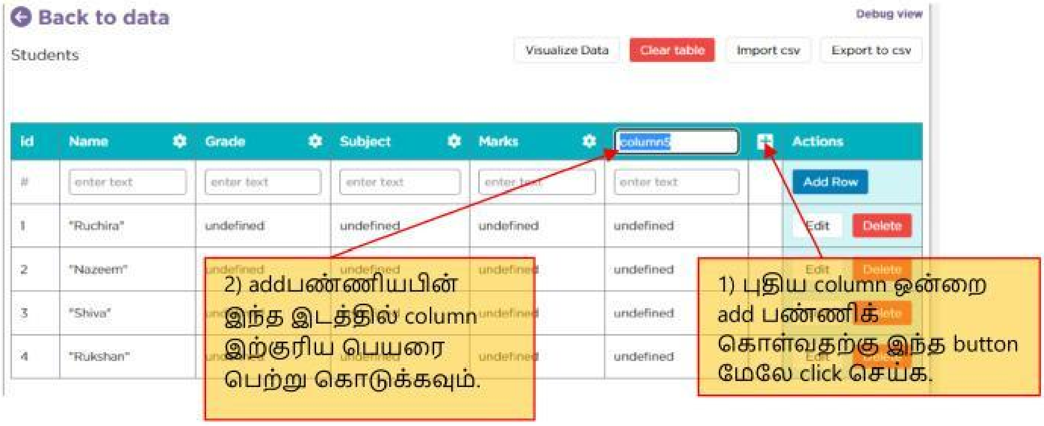
எல்லா records add பண்ணிய பின் back to data மூலம் back ஆக முடியும். மீண்டும் மேலும் records add பண்ண வேண்டுமாயின் மேலுள்ளவாறு table name இன் மேல் click செய்து மீண்டும் table ஐ view பண்ண முடியும்.

❖ இவ்வாறு மேலும் புதிய table ஒன்று "Students" என பின்வருமாறு உருவாக்கிக் கொள்க.

LIVEWORKSHEETS



இதனை உருவாக்கும் போது 4 column பாவிக்கப்பட்டுள்ளன. Name, Grade, Subject, Marks என அவ்வாறு புதிய column add பண்ணிக்கொள்ள பின்வரும் படிமுறைகளை பின்பற்றவும்.



❖ App உருவாக்குவதற்கு தேவையான data table இரண்டையும் உருவாக்கி முடிந்தபின் app உருவாக்கத்தை ஆரம்பிப்போம். உங்களுக்குத் தேவையான screens பெற்றுத் தரப்பட்டுள்ளது. Code செய்வதற்கு ஆரம்பிப்போம்.

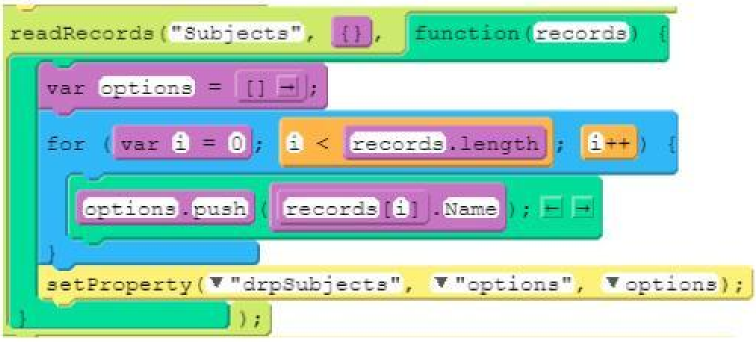
❖ முதலில் search box இற்குள் input செய்யும் பெயரை store பண்ணுவதற்கும் உட்படுத்திய பெயர் data table இற்கு உள்ளடங்கியுள்ளதா என்று பரீட்சித்துப் பார்ப்பதற்கும் பின்வருமாறு variable இரண்டு உருவாக்கவும்.

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

LIVEWORKSHEETS

❖ அதன் பின் Add Marks button இன் மேல் click செய்யும் போது புள்ளிகள் add பண்ணும் screen இற்கு மாறுவதற்கு உருவாக்கவும்.

❖ இங்கு subject என்ற drop down இற்கு options பெற்று கொடுப்பதற்கு மேலே உருவாக்கிய subjects என்ற data table இலுள்ள subjects பாவிப்போம்.



- readRecord block மூலம் Subjects என நாங்கள் உருவாக்கிய data table ஐ access பண்ணும்.
- இங்கு option என பெயரிடப்பட்ட array ஒன்றை உருவாக்கி கொள்ளவும். அதற்கு data table இன் data சேகரிப்புக்கு பயன்படுத்தப்படும்.
- For loop மூலம் எல்லா records மூலம் போய் ஒவ்வொரு record இன் Name column இலுள்ள பெறுமானம் மேலே உருவாக்கிக் கொண்ட "options" array இற்குள் சேமிக்கப்படும்.

- இதற்காக arrays களில் உள்ள push என்ற function பாவிக்கப்படும். அதன் மூலம் array இற்கு புதிய பெறுமானத்தை உட்படுத்திக் கொள்ள முடியும்.

LIVEWORKSHEETS

- அதன் பின் அவ்வாறு உட்படுத்திய options array ஐ

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

dropdown இற்குப் பெற்றுக் கொடுப்பதற்கு இந்த block பாவிக்கப்பட்டுள்ளது.

- பிள்ளை ஒன்றின் விபரங்களை update செய்வதற்கு முன்பு பிள்ளையின் பெயரை search box இற்குள் type செய்து search button ஐ click செய்து அப்பெயரை students table இற்குள் உட்படுத்தப்பட்டுள்ளதா எனப் பார்க்க முடியும். அதற்காக பின்வருமாறு code செய்யவும்.

```
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 இற்குள் input செய்யப்பட்ட பெறுமானங்களை மேலே உருவாக்கிய variable இற்கு உட்படுத்திக் கொள்ளல்

Student table இல் உள்ள records, read பண்ணி அந்த ஒவ்வொரு record இன் Name column இன் பெறுமானத்தை for loop இல் ஒவ்வொரு loop இலும் recordLc என்ற variable இற்குள் போட்டுக் கொள்ளும்.

அதன் பின் இந்த variable இன் பெறுமானத்தை search box இற்குள் உட்படுத்தி பெறுமானங்களுக்கு சமனா என்று if block மூலம் check பண்ணப்படும். அவ்வாறு சமமான record ஒன்று இருந்தால் ஒவ்வொரு column இன் values txtName, txtGrade, drpSubjects, txtMarksAdd என்ற fields களுக்கு set செய்யும்.

மேலே check செய்த condition, false ஆனால், அதாவது input செய்த name, data table இல் உட்படுத்தபடாவிட்டால் not found message ஒன்று தோன்றி ஒரு வினாடிக்கு பின் அது மறைந்து செல்வதற்கு codes பயன்படுத்தவும். "lblSubmit" என label ஒன்று இதற்காக design பண்ணப்பட்டுள்ளது. அதனை பயன்படுத்தவும்.

LIVEWORKSHEETS

- இவ்வாறு code செய்த பின் data table இற்குள் search செய்யும் பெயர் இருந்தால் text field இற்குள் அந்த record களின் பெறுமானங்களை காட்டும்.
- அதன் பின் அவ்வாறு search செய்த record ஒன்றை update செய்வதற்கு புள்ளிகள் add பண்ணும் போது plus button மற்றும் minus button இன் செயற்பாட்டிற்கு பின்வருமாறு code செய்யவும்.

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

Marks என புதிய variable ஒன்றை உருவாக்கி "txtMarksAdd" என்ற புள்ளிகளை add பண்ணும் text box இன் பெறுமானம் getNumber மூலம் பெற்று store செய்யவும். getNumber மூலம் அந்த textbox இன் பெறுமானம் number எனப் பெற்று variable இல் store செய்யும்.

Plus button ஐ click செய்யும் போது textbox இன் பெறுமானம் ஒன்றால் கூடுவதற்கு இவ்வாறு code செய்யப்பட்டுள்ளது.

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

- இவ்வாறே minus button இன்மேல் click செய்யும் போது text box இன் பெறுமானத்தில் ஒன்று குறையுமாறு உருவாக்கவும்.
- அதன் பின் search செய்து update செய்து கொண்ட record ஒன்றை submit செய்து data table இல் save செய்து கொள்ளும் முறையை பார்ப்போம்.
- Submit என்ற button இன் மேல் click செய்யும் போது அவ்வாறு update ஆவதற்கு onEventClick block ஐ பின்வருமாறு பயன்படுத்தவும்.

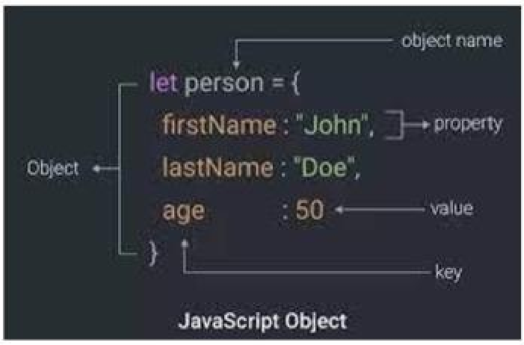
```
onEvent("btnSubmit", "click", function( ) {
  var name = getText("txtName");
  var grade = getText("txtGrade");
  var subject = getText("drpSubjects");
  var marks = getText("txtMarksAdd");
```

LIVEWORKSHEETS

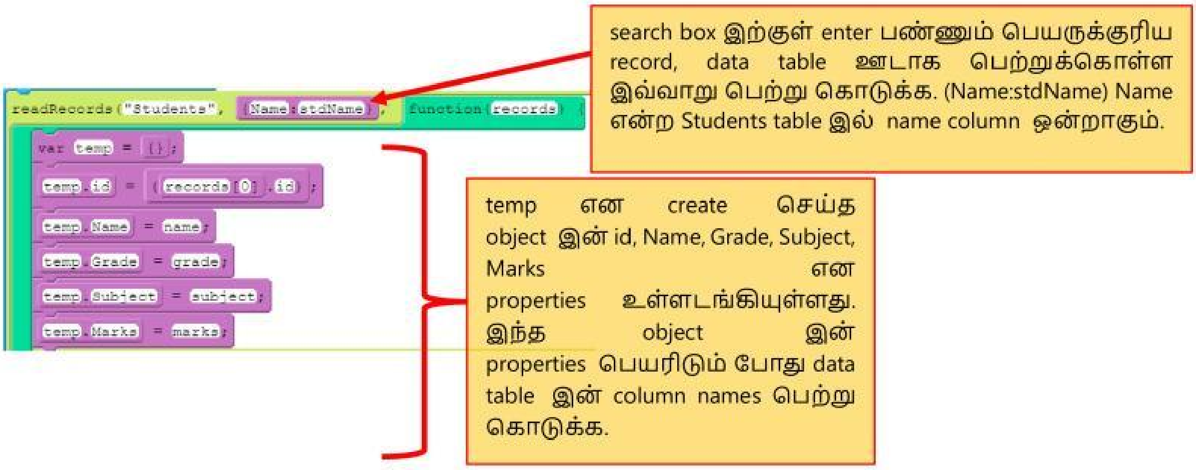
❖ name, grade, subject மற்றும் marks என variable 4 create செய்து getText மூலம் ஒவ்வொரு textbox களின் பெறுமானங்களை variable களில் store செய்து கொள்ளும்.

❖ அதன் பின் search பண்ணும் பெயர், data table இற்குள் உள்ள பெயர்களை மேலே register variable true பண்ணிக் கொள்ளும். இங்கு true என்றால் அப்பெயர்களுக்குரிய record ஐ readRecord block மூலம் பின்வருமாறு பெற்று அதன் பெறுமானங்களை temp என்ற object சேமித்துக் கொள்ளும்.

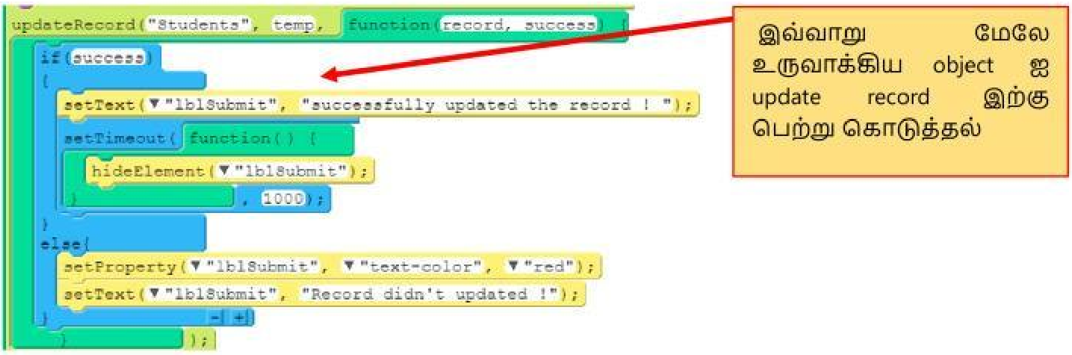
❖ Object என்பது வெவ்வேறு வகையான data type உடன் properties அடங்கியுள்ள entity ஆகும். கீழே காட்டப்பட்டுள்ளது அவ்வாறு person என create செய்த object ஆகும். Object இற்குள், ஒரு property இற்கு key ஒன்று மற்றும் value ஒன்று அடங்கியுள்ளது. இவ்வாறு properties பல object ஒன்றில் அடங்கியுள்ளது.



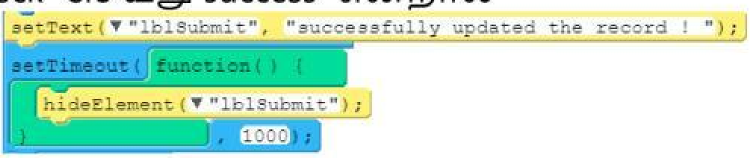
LIVEWORKSHEETS



❖ இங்கு text box களில் enter செய்து கொண்ட பெறுமானங்களை object இன் properties இற்குள் store செய்து கொள்ளும். இந்த object data table update ஆவதற்கு பின்வருமாறு pass செய்தல் நடைபெறும்.



❖ Update ஆதல் சரியாக நடைபெற்றால் function இற்குள் success parameter pass ஆவது நடைபெறும். அது if block இற்குள் check செய்து success என்றால்



❖ இந்த block display ஆகி ஒரு வினாடியில் மறையும்.

Success இல்லை என்றால் பின்வரும் blocks display ஆகும்.

LIVEWORKSHEETS

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

❖ Submit button ஐ click செய்யும் போது நடைபெற வேண்டியவற்றுக்காக முழுமையாக code பின்வருமாறு அமையும்.

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

LIVEWORKSHEETS

- இவ்வாறு app ஐ உருவாக்கி app மூலம் record update செய்து மீண்டும் data table இற்கு சென்று அது update ஆகியுள்ளதா என பார்க்கவும் . சரியாக app ஐ உருவாக்கியிருந்தால் அவ்வாறு update செய்யும்போது data table update ஆக வேண்டும்.

- கடைசியில் back button ஐ click செய்யும் போது மீண்டும் home screen இற்கு செல்வதற்கு உருவாக்கவும்.

LIVEWORKSHEETS