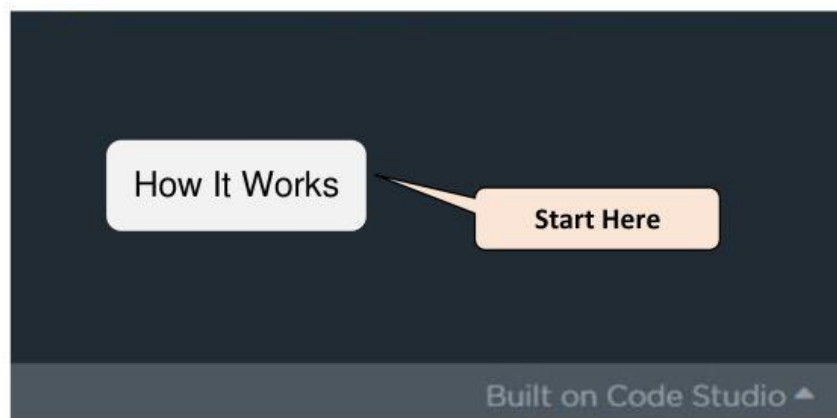


Project 94



Coding School



- ❖ Let's create how to view the results added by the user in the 93rd project.
- ❖ For this use the 93 project itself and develop it.
- ❖ First, after adding and submitting the report details, compare it with the standard values in the QuizTable and create how to display the health status in the report screen.
- ❖ For this, when the submit button is clicked, the value entered for HDL must be stored in the hdl variable.

```
onEvent(▼"btnReportSubmit", ▼"click", function() {
  hdl = getNumber(▼"txtValue");
  createRecord("HealthRecords", {UserId:loggedUserId,Suger:suger,Presure:presure,TotalCol:totalCol,LDL:
  });
});
```

- ❖ Then read the quiz table and compare the inputted values with the standard values and create how to display the health status for each value.

```
readRecords("QuizTable", {}, function(records) {
  if(suger > records[0].StandardHighLevel)
  {
    setText(▼"lblSuger", "High blood suger");
  }
  else if(suger < records[0].StandardLowLevel)
  {
    setText(▼"lblSuger", "Low blood suger");
  }
  else
  {
    setText(▼"lblSuger", "Normal blood suger");
  }
  if(presure > records[1].StandardHighLevel)
  {
    setText(▼"lblPresure", "High blood suger");
  }
  else if(presure < records[1].StandardLowLevel)
  {
    setText(▼"lblPresure", "Low blood presure");
  }
  else
  {
    setText(▼"lblPresure", "Normal blood presure");
  }
});
```

Display as "High blood suger" if the value entered by the user for blood suger is higher than the standard high level in the data table.

```

if( totalCol < records[2].StandardLowLevel )
{
    setText(▼"lblTotCol", "Normal total presure");
}
else if( totalCol > records[2].StandardLowLevel && totalCol < records[2].StandardHighLevel )
{
    setText(▼"lblTotCol", "Borderline total presure");
}
else if( totalCol > records[2].StandardHighLevel )
{
    setText(▼"lblTotCol", "High total presure");
}
if( ldl > records[3].StandardLowLevel && ldl < records[3].StandardHighLevel )
{
    setText(▼"lblLdl", "Optimal LDL");
}
else if( ldl > records[3].StandardHighLevel )
{
    setText(▼"lblLdl", "High LDL");
}
if( hdl > records[4].StandardHighLevel )
{
    setText(▼"lblHdl", "High HDL");
}
else if( hdl < records[4].StandardLowLevel )
{
    setText(▼"lblHdl", "Low HDL");
}
else
{
    setText(▼"lblHdl", "Optimal HDL");
}
setScreen(▼"Report");
}

```

- ❖ As above, compare each value with the standard values and display them as High if they are higher than the required value, and Low if they are lower than the required value, and then display them as Optimal or normal in the respective label.
- ❖ Then, when the homepage button is clicked, let's create to refer back to the dashboard.

```

onEvent(▼"btnHome", ▼"click", function() {
    setScreen(▼"Dashboard");
});

```

- ❖ Now, when you click on the view current health status button on the dashboard, let's create it to refer to the viewReport screen.

```
onEvent (▼ "btnViewStatus", ▼ "click", function () {
  setScreen (▼ "viewReport");
  viewReport ();
});
```

- ❖ When referring to that screen, viewReport(); to display the report details last entered by the respective user. The function is called.
- ❖ Now let's create that function.
- ❖ When viewing the report, use the users table to display the user's name, age, etc. related to the report and the HealthRecords data table to display the health status

```
function viewReport () {
  readRecords ("Users", {}, function(records) {
    for (var i=0; i < records.length; i++) {
      if (loggedUserId == records[i].id) {
        setText (▼ "lblRVName", records[i].Name);
        setText (▼ "lblRVAge", records[i].Age);
        setText (▼ "lblRVGender", records[i].Gender);
      }
    }
  });
}
```

- ❖ Here, the Id of the logged user is stored in the loggedUserId variable. Read the same id from the users table and set the values of the Name, Age, Gender columns to the labels as above.

```
setText (▼ "lblRVName", records[i].Name);
setText (▼ "lblRVAge", records[i].Age);
setText (▼ "lblRVGender", records[i].Gender);
```

- ❖ Now check if there is a record in the HealthRecords table related to the same user id and set the column values to the relevant labels.


```

readRecords("HealthRecords", {}, function(records) {
  for (var i = 0; i < records.length; i++) {
    if (loggedUserId == records[records.length - 1].UserId) {
      setText(▼ "lblRVSuger", records[records.length - 1].Suger);
      setText(▼ "lblRVPressure", records[records.length - 1].Pressure);
      setText(▼ "lblRVTotCol", records[records.length - 1].TotalCol);
      setText(▼ "lblRVLDL", records[records.length - 1].LDL);
      setText(▼ "lblRVHDL", records[records.length - 1].HDL);
    }
  }
});

```

- ❖ Here, to view the last health record entered by the logged in user, the index of the records array has been given as length -1 as above.

```
records[records.length - 1]
```

This is done because the index of an array always starts at 0.

If there are 10 records, the index of the last record is 9.

- ❖ If a record related to that user id is stored in the table, set the values of each column to labels like this.

```

setText (▼ "lblRVSuger", records[ records.length - 1 ] .Suger );
setText (▼ "lblRVPresure", records[ records.length - 1 ] .Presure );
setText (▼ "lblRVTotCol", records[ records.length - 1 ] .TotalCol );
setText (▼ "lblRVLDL", records[ records.length - 1 ] .LDL );
setText (▼ "lblRVHDL", records[ records.length - 1 ] .HDL );

```

- ❖ When you click the Home button to return to the dashboard, set the Home screen like this.

```

onEvent (▼ "btnRVHome", ▼ "click", function ( ) {
    setScreen (▼ "dashboard");
} );

```