

Project 99

99



DP
EDUCATION

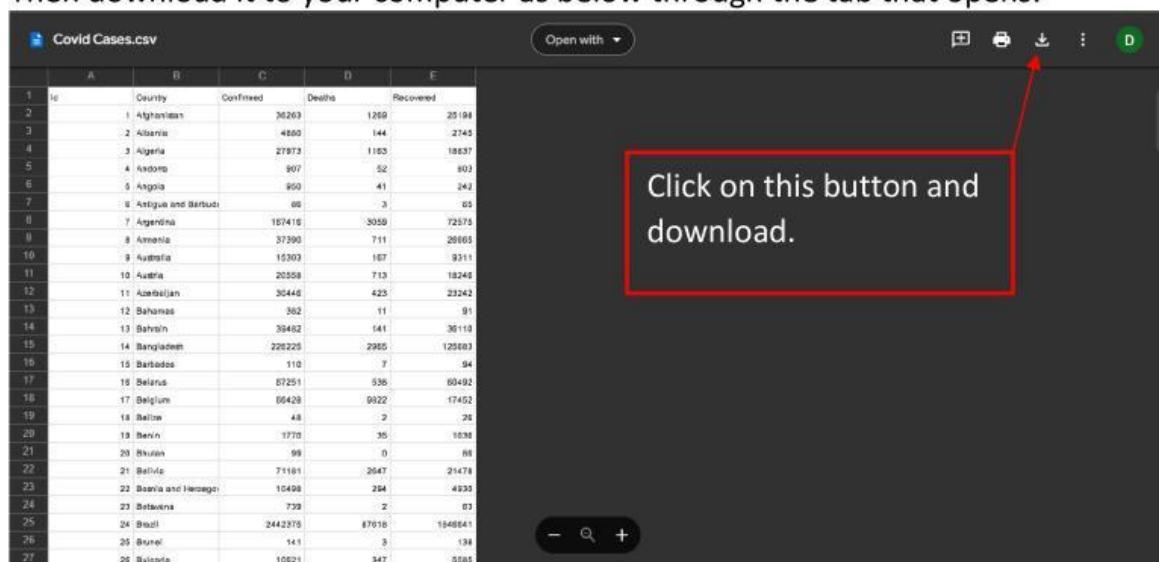
Coding School

How It Works

Start Here

Built on Code Studio ▾

- ❖ Let's create an App that shows the spread of the Corona virus and its impact in each country.
- ❖ The design related to the design of this app and the required data table have been given to you. Click on How it works and start the project.
- ❖ Now let's see how to create the data table.
- ❖ First, download the csv file related to the data table using the google drive link below.
https://drive.google.com/file/d/1krv9se1b0ltLsU-930SUe0lUm1p70D_1/view?usp=sharing
- ❖ Then download it to your computer as below through the tab that opens.



Click on this button and download.

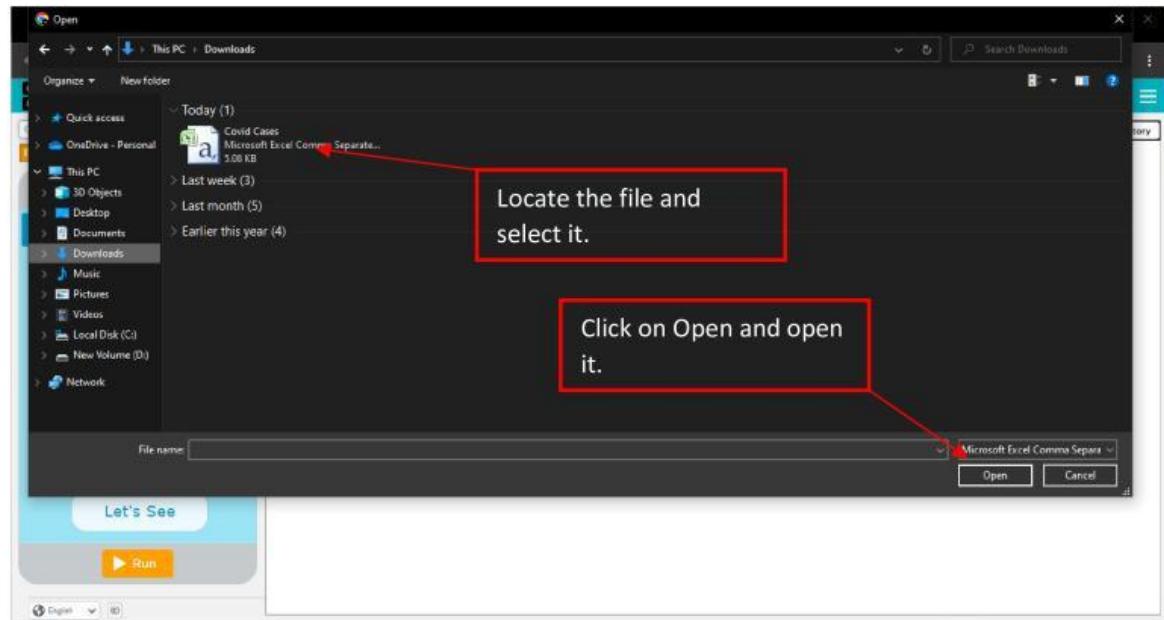
	A	B	C	D	E
1	Id	Country	Confirmed	Deaths	Recovered
2	1	Afghanistan	36263	1269	25198
3	2	Albania	4800	144	2742
4	3	Algeria	27973	1103	18637
5	4	Angola	907	52	803
6	5	Angola	950	41	242
7	6	Anguilla and Barbuda	65	3	65
8	7	Argentina	167416	3059	72575
9	8	Armenia	37390	711	26605
10	9	Australia	15303	167	9311
11	10	Austria	20558	713	18246
12	11	Azerbaijan	30448	425	23242
13	12	Bahamas	362	11	91
14	13	Bahrain	39482	141	36110
15	14	Bangladesh	228225	2985	129693
16	15	Barbados	110	7	94
17	16	Belarus	67251	538	60492
18	17	Belgium	66428	9322	17452
19	18	Belize	48	2	26
20	19	Bern	1770	35	1538
21	20	Brunei	98	0	88
22	21	Bolivia	71101	2647	21478
23	22	Bosnia and Herzegovina	10498	294	4330
24	23	Botswana	739	2	83
25	24	Bouvet	2442376	37516	1949641
26	25	Burkina	141	3	138
27	26	Bulgaria	10621	347	5585

❖ Go to the data tab and add a table as "Covid Cases" as below.

The screenshot shows the Data Library interface. The 'Data' tab is selected. In the 'Data Tables' section, there is a text input field for 'Table name' containing 'Covid Cases'. A red arrow points to the 'Add' button. A callout box on the right contains the text: 'Give the name as "Covid Cases" and click on the Add button.'

❖ Open the created table and import the data table downloaded as below.

The screenshot shows the Data Browser interface. The 'Data' tab is selected. A table is displayed with a single row and one column. The 'Actions' column contains a blue 'Add Row' button. A red arrow points to this button. A callout box on the right contains the text: 'Open the file containing the data table downloaded by Import CSV.'



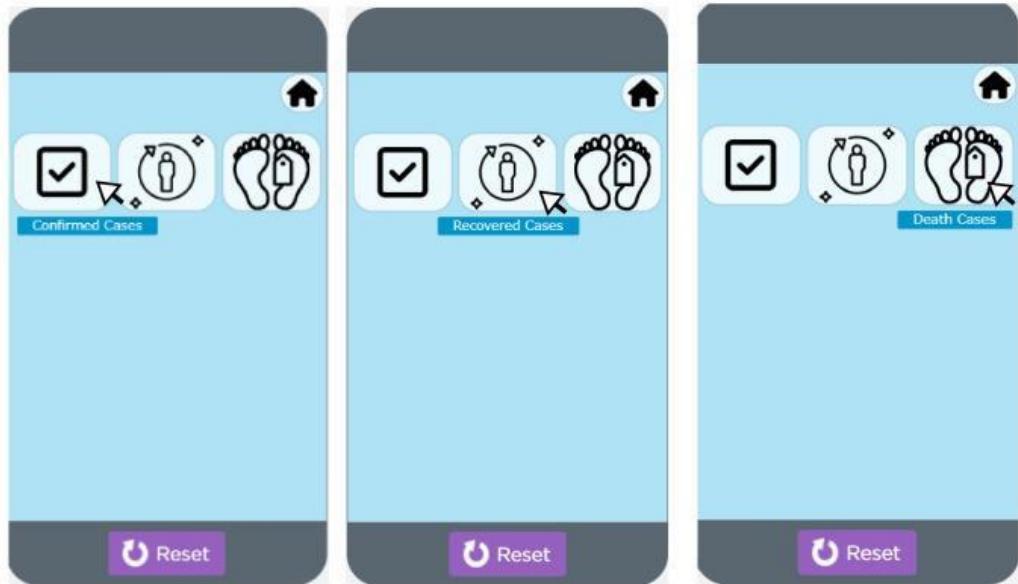
- ❖ Then the data table will open as follows.

	Id	Country	Confirmed	Deaths	Recovered	Actions
1	1	"Afghanistan"	36263	1269	25198	<button>Edit</button> <button>Delete</button>
2	2	"Albania"	4880	144	2745	<button>Edit</button> <button>Delete</button>
3	3	"Algeria"	27923	1863	18837	<button>Edit</button> <button>Delete</button>
4	4	"Andorra"	907	52	803	<button>Edit</button> <button>Delete</button>
5	5	"Angola"	950	41	242	<button>Edit</button> <button>Delete</button>
6	6	"Antigua and Barbuda"	86	5	65	<button>Edit</button> <button>Delete</button>
7	7	"Argentina"	167436	1059	72575	<button>Edit</button> <button>Delete</button>
8	8	"Armenia"	57590	211	26665	<button>Edit</button> <button>Delete</button>

- ❖ When you click on the "Let's see" button to go to the Cases Screen, code like this.

```
onEvent( ▶ "btnViewCases", ▶ "click", function() {
    setScreen( ▶ "Cases" );
});
```

- ❖ Create the label that shows its name when moving the mouse over the three buttons on the Cases Screen and hide it when the mouse is removed.



- ❖ For that, the above label 3 has been hidden in the design. Let's create show hide by code as above.
- ❖ Code as follows to create what happens when the mouse is placed on the Confirmed cases button and when the mouse is removed.

```
onEvent(¬"btnConfirmed", ¬"mouseover", function() {
  showElement(¬"btnConfirmLabel");
});
onEvent(¬"btnConfirmed", ¬"mouseout", function() {
  hideElement(¬"btnConfirmLabel");
});
```

- ❖ Code as follows to create what happens when the mouse is brought to the Recovered cases button and when the mouse is removed.

```
onEvent(¬"btnRecovered", ¬"mouseover", function() {
  showElement(¬"btnRecoveredLabel");
});
onEvent(¬"btnRecovered", ¬"mouseout", function() {
  hideElement(¬"btnRecoveredLabel");
});
```

- ❖ Code as follows to create what happens when the mouse is brought to the death cases button and when the mouse is removed.

```

onEvent(▼ "btnDeath", ▼ "mouseover", function() {
  showElement(▼ "btnDeathLable");
}
);
onEvent(▼ "btnDeath", ▼ "mouseout", function() {
  hideElement(▼ "btnDeathLable");
}
);

```

- ❖ Now let's see how to create the graph of records related to each button when clicked.
- ❖ For this, the columns of the table imported above should be used.
- ❖ First, when you click on the Confirmed cases button, read the data table and code as follows to display the reported cases in each country through a line chart.

```

onEvent(▼ "btnConfirmed", ▼ "click", function() {
  drawChartFromRecords(▼ "chart1", ▼ "line", "Covid Cases", ["Country", "Confirmed"]);
}
);

```

Give the id of the chart in the Design to this position

The name of the data table

The two columns to be obtained from the data table for the x and y axes of the chart

Select the type of chart to be created at this point.

- ❖ As above, when you click on the Recovered cases button, generate the chart from the related records. For that, prepare the codes as follows.

```

onEvent(▼ "btnRecovered", ▼ "click", function() {
  drawChartFromRecords(▼ "chart1", ▼ "line", "Covid Cases", ["Country", "Recovered"]);
}
);

```

- ❖ When you click on the Death cases button, code as follows to create the chart from the records related to it.

```

onEvent(▼ "btnDeath", ▼ "click", function() {
  drawChartFromRecords(▼ "chart1", ▼ "line", "Covid Cases", ["Country", "Deaths"]);
}
);

```

- ❖ Finally, when you click on the Home button, code as follows to go back to the home screen.

```

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

```