

Zewail University Dashboard Report

A) Dashboard Informations

1) Dashboard Users:

- 1) University Manager
- 2) Financial Manager
- 3) Marketing Manager
- 4) Academic Manager

2) Information Provided:

- 1) Comparison Between the Income and the Expenses.
- 2) Percentage of students in each faculty.
- 3) Knowing the student's grades in each faculty and identifying the faculty with the highest number of top performers.
- 4) Rank of the University in the last years.

3) Questions answered by the dashboard:

- 1) What is the faculty with the largest number of top performers?
- 2) what are the students grades?
- 3) what is the faculty that have largest number of students?
- 4) what is the Rank of University in the last years?

-Why selecting this position for the charts and title?

Because the Title of the Dashboard should be at the top left and the charts should be presented in way that makes them understandable and user friendly.

-Suggested future work: personalized Learning analytics – Real time attendance tracking

B) Charts Informations

1) Chart 1 Informations:

- 1) Chart type: PieChart
- 2) Title: Percentage of Students in each Faculty
- 3) Colors: Purple - LightBlue - LightCyan
- 4) Axis titles: No Axis Because Its PieChart

- 5) Legend: No legends
- 6) Question answered by the chart: What is the percentage of students in each faculty.
- 7) How will show up: Always shows up
- 8) Why I have selected this chart type: to represent the qualitative data and the percent.

2) Chart 2 Informations:

- 1) Chart type: Bar and Line mix.
- 2) Title: University Income vs Expenses.
- 3) Colors: LightBlue.
- 4) Axis titles: (Xaxis: Years) - (Yaxis: Money) .
- 5) Legend: (Income - Expenses).
- 6) Question answered by the chart: What is the financial performance of the University.
- 7) How will show up: Always Show up.
- 8) Why i selected this chart type: to represent the Comparison between The Expenses and the Income to find out whether the University is making a profit or incurring losses.

3) Chart 3 Informations:

- 1) Chart type: Clustered BarChart.
- 2) Title: Faculties Grades Comparison.
- 3) Colors: Purple - LightBlue – LightCyan.
- 4) Axis titles: (Xaxis: Grade Letter) - (Yaxis: Number of students).
- 5) Legend: Title of each faculty.
- 6) Question answered by the chart: What is the Performance of students in each faculty.
- 7) How will show up: Always shows up.
- 8) Why i selected this chart type: to represent the Comparison between each faculty at the same time.

4) Chart 4 Informations:

- 1) Chart type: Step Line chart.
- 2) Title: University's Rank.
- 3) Colors: LightBlue.
- 4) Axis titles: (Xaxis: Years) - (Yaxis: Rank).
- 5) Legend: No Legends.
- 6) Question answered by the chart: the University rank is High in the last few years or not.
- 7) How will show up: Always shows up.

8) Why i selected this chart type: Want to show changes happening at specific each year.

=====

C) Snapshots

Chart 1 PieChart:

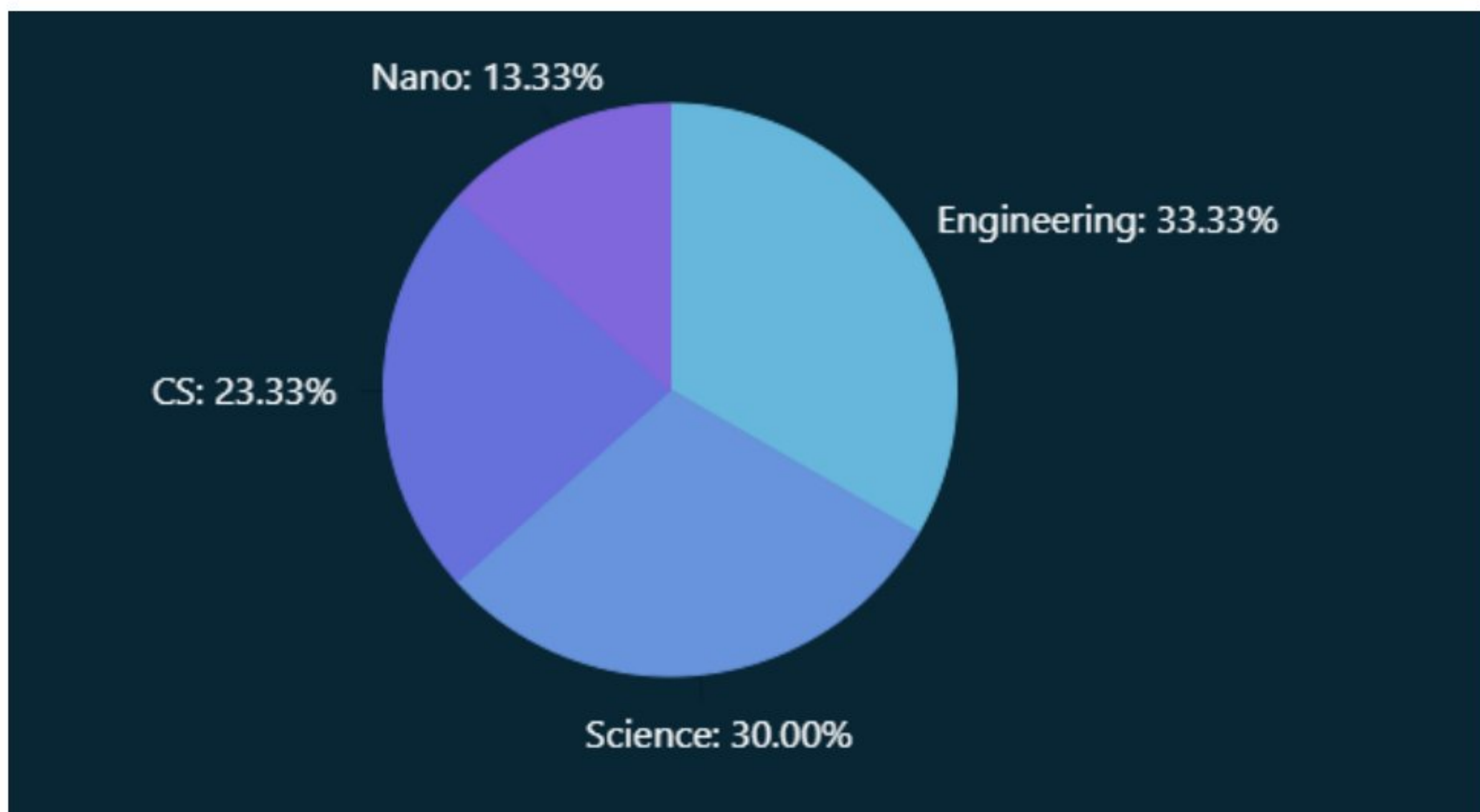


Chart 2 Bar and Line mix:



Chart 3 Clustered BarChart:

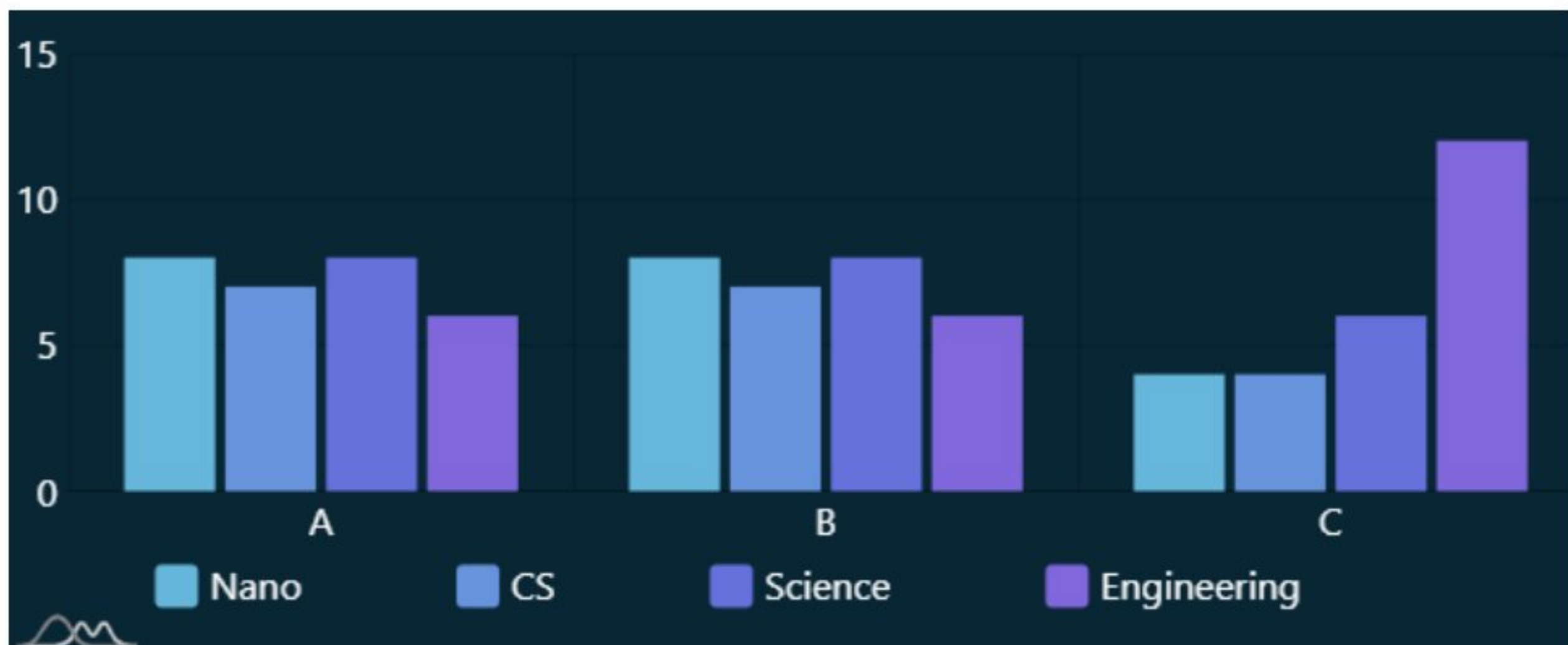
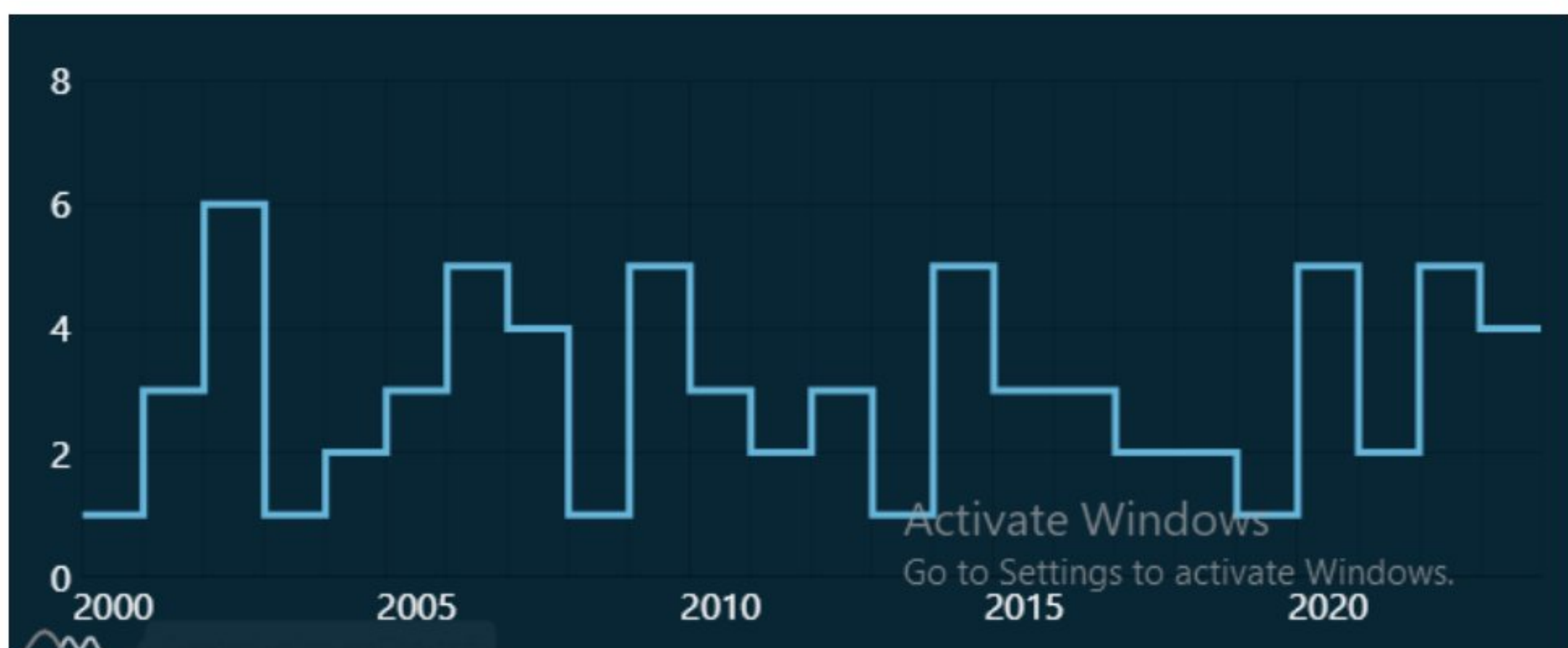


Chart 4 Step LineChart:



Complete Layout:



Provided Data:

```
Row_num,Grade_letter,Student_ID,Semester,Course_number,Faculty
1,A,202201501,Fall,203,CS
2,B,202201602,Spring,201,Science
3,C,202201203,Fall,205,Nano
4,C,202201701,Spring,203,Engineering
5,A,202201802,Fall,202,Nano
6,C,202201903,Spring,205,Science
7,B,202202004,Fall,203,CS
8,B,202202105,Spring,201,Engineering
9,C,202202206,Fall,202,Nano
10,A,202201306,Spring,203,CS
11,C,202201307,Fall,201,Science
12,C,202201308,Spring,205,Engineering
13,A,202101705,Fall,203,Science
14,B,202201107,Spring,201,Engineering
15,A,202202307,Fall,202,Nano
16,B,202001304,Fall,203,CS
17,B,202001522,Spring,201,Science
18,B,202001404,Fall,205,Engineering
19,A,202001120,Spring,203,Nano
20,A,202001305,Fall,202,Nano
21,B,202102408,Spring,205,Science
22,A,202102509,Fall,203,CS
23,B,202101821,Spring,201,Engineering
24,C,202101715,Fall,202,CS
25,A,202101406,Spring,203,Nano
26,B,202102418,Fall,201,Engineering
27,C,202201308,Spring,205,Nano
28,A,202102519,Fall,203,CS
29,B,202101021,Spring,201,CS
30,C,202100715,Fall,202,CS
```


Row_num	Major	Student_ID	Name	Phone	Street_Num	Street_Name	B_date	Gender	Email	Faculty_Name
1	AI	202201501	Hamza Abdelmoreed	1142859850	10	Nile Street	01/10/2004	Male	hamza.fekry@zewailcity	Science
2	Software	202201602	Aya Ali	1142859851	20	Pyramids Street	15/02/2004	Female	aya.ali@zewailcity	CS
3	IT	202201203	Reem Samy	1142859852	30	Cairo Street	20/08/2004	Female	reem.samy@zewailcity	Nano
4	AI	202201701	Ahmed Hassan	1142859853	40	Sphinx Street	25/04/2004	Male	ahmed.hassan@zewailcity	Science
5	Software	202201802	Fatima Mohamed	1142859854	50	Pharaohs Street	30/10/2004	Female	fatima.mohamed@zewailcity	CS
6	IT	202201903	Omar Ali	1142859855	60	Nile Banks	05/01/2004	Male	omar.ali@zewailcity	CS
7	AI	202202004	Layla Ibrahim	1142859856	70	Nile View Street	12/03/2004	Female	layla.ibrahim@zewailcity	Nano
8	Software	202202105	Khaled Salah	1142859857	80	Pyramids View Street	18/09/2004	Male	khaled.salah@zewailcity	CS
9	IT	202202206	Noura Hassan	1142859858	90	Cairo View Street	25/12/2004	Female	noura.hassan@zewailcity	Science
10	AI	202201306	Fatima Ali	1142859864	90	Maadi Corniche	25/12/2004	Female	fatima.ali@zewailcity	CS
11	Software	202201307	Youssef Samir	1142859865	100	October City Street	05/08/2004	Male	youssef.samir@zewailcity	Nano
12	IT	202201308	Nourhan Ahmed	1142859866	130	New Cairo Street	28/02/2004	Female	nourhan.ahmed@zewailcity	CS
13	AI	202101705	Sarah Youssef	1142859854	50	Tahrir Street	30/10/2004	Female	sarah.youssef@zewailcity	Nano
14	Software	202201107	Youssef Ahmed	1142859865	100	October City Street	05/08/2004	Male	youssef.ahmed@zewailcity	CS
15	AI	202202307	Mohamed Emam	1142859859	100	Maadi View Street	05/07/2004	Male	mohamed.emam@zewailcity	Science
16	Nanotechnology	202102408	Mona Mahmoud	1142859860	110	Tahrir View Street	30/11/2003	Female	mona.mahmoud@zewailcity	Engineering
17	Renewable	202102509	Amr Samir	1142859861	120	Giza View Street	15/06/2003	Male	amr.samir@zewailcity	Engineering
18	CIE	202101821	Eman Farouk	1142859870	260	Sharkia Street	15/07/2003	Female	eman.farouk@zewailcity	Engineering
19	Nanotechnology	202101715	Sarah Youssef	1142859854	50	Tahrir Street	30/10/2003	Female	sarah.youssef@zewailcity	Engineering
20	Renewable	202101406	Mona Khaled	1142859855	60	Giza Street	05/01/2003	Male	marwan.khaled@zewailcity	Engineering
21	Nanotechnology	202102418	Mariam Samir	1141859860	110	Tahrir View Street	#####	Female	mariam@zewailcity	Engineering
22	Renewable	202102519	Amr Farouk	1141859861	120	Giza View Street	11/06/2003	Male	amr.samir@zewailcity	Engineering
23	CIE	202101021	Haneen Khaled	1141859870	260	Sharkia Street	05/07/2003	Female	Haneen.Khaled@zewailcity	Engineering
24	Nanotechnology	202100715	Sarah Youssef	1141859854	50	Tahrir Street	30/09/2003	Female	sarah.youssef@zewailcity	Engineering
25	Renewable	202101416	Mahmoud Khaled	1141859855	60	Giza Street	05/01/2003	Male	Mahmoud.khaled@zewailcity	Engineering
26	Biomedical	202001304	Ahmed Mustafa	1142859853	40	Maadi Street	25/04/2002	Male	ahmed.mustafa@zewailcity	Science
27	Nanoscience	202001522	Hussein Mahmoud	1142859871	270	Assiut Street	20/03/2002	Male	hussein.mahmoud@zewailcity	Science
28	Physics	202001120	Norhan Adel	1142859869	250	Fayoum Street	10/11/2002	Female	norhan.adel@zewailcity	Science
29	Biomedical	202001404	Layla Ahmed	1142859862	70	Zamalek Street	20/09/2003	Female	layla.ahmed	Science
30	Physics	202001305	Omar Hassan	1142859863	80	Mohandiseen Street	10/04/2003	Male	omar.hassan	Science

Row_num	Doctor_ID	Name	phone_number	Office_location	Office_hours	Course	Address	Email
1	10001	'Dr. Maher'	'890'	'Academic'	'9 AM - 5 PM'	'Circuits'	'123 Main St'	'dr.maher@zewailcity'
2	10002	'Dr. Khalid'	'210'	'Academic'	'10 AM - 6 PM'	'Visualization'	'456 Elm St'	'dr.khalid@zewailcity'
3	10003	'Dr. Yousry'	'333'	'Academic'	'8 AM - 4 PM'	'Database'	'789 Oak St'	'dr.yousry@zewailcity'
4	10004	'Dr. Mayda'	'555'	'Academic'	'11 AM - 7 PM'	'Datastructure'	'101 Pine St'	'dr.mayda@zewailcity'
5	10005	'Dr. Doaa'	'577'	'Academic'	'7 AM - 3 PM'	'OOP'	'202 Cedar St'	'dr.doaa@zewailcity'
6	10006	'Dr. Ashraf'	'891'	'Academic'	'9 AM - 5 PM'	'Network'	'123 Main St'	'dr.Ashraf@zewailcity'
7	10007	'Dr. Hatem'	'210'	'Academic'	'10 AM - 6 PM'	'Visualization'	'456 Elm St'	'dr.Hatem@zewailcity'
8	10008	'Dr. Azza'	'303'	'Academic'	'8 AM - 4 PM'	'Database'	'789 Oak St'	'dr.Azza@zewailcity'
9	10009	'Dr. Hadidy'	'550'	'Academic'	'11 AM - 7 PM'	'Datastructure'	'101 Pine St'	'dr.mayda@zewailcity'
10	10010	'Dr. Waleed'	'840'	'Academic'	'9 AM - 5 PM'	'Probability'	'123 Main St'	'dr.Waleed@zewailcity'
11	10011	'Dr. Mahmoud'	'210'	'Academic'	'10 AM - 6 PM'	'DATA'	'456 Elm St'	'dr.Mahmoud@zewailcity'
12	10012	'Dr. Ehab'	'333'	'Academic'	'8 AM - 4 PM'	'Cloud'	'789 Oak St'	'dr.Ehab@zewailcity'
13	10013	'Dr. Sabah'	'555'	'Academic'	'11 AM - 7 PM'	'Reasoning'	'101 Pine St'	'dr.Sabah@zewailcity'
14	10014	'Dr. Shref'	'577'	'Academic'	'7 AM - 3 PM'	'Ethics'	'202 Cedar St'	'dr.Shref@zewailcity'
15	10015	'Dr. Mahdy'	'891'	'Academic'	'9 AM - 5 PM'	'Logic'	'123 Main St'	'dr.Mahdy@zewailcity'
16	10016	'Dr. Anas'	'210'	'Academic'	'10 AM - 6 PM'	'Programming'	'456 Elm St'	'dr.Anas@zewailcity'
17	10017	'Dr. Ahmed'	'303'	'Academic'	'8 AM - 4 PM'	'Acquisition'	'789 Oak St'	'dr.Ahmed@zewailcity'
18	10018	'Dr. Amr'	'550'	'Academic'	'11 AM - 7 PM'	'Physics'	'101 Pine St'	'dr.Amr@zewailcity'

Source Code of the Dashboard:

HTML:

```
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="UTF-8" />
5      <title>Chart with Flask and amCharts 5</title>
6      <script src="https://cdn.amcharts.com/lib/5/index.js"></script>
7      <script src="https://cdn.amcharts.com/lib/5/xy.js"></script>
8      <script src="https://cdn.amcharts.com/lib/5/percent.js"></script>
9      <script src="https://cdn.amcharts.com/lib/5/themes/Animated.js"></script>
10     <script src="https://cdn.amcharts.com/lib/5/themes/Micro.js"></script>
11     <script src="/static/chart.js"></script>
12     <script src="/static/data.js"></script>
13     <script src="/static/bar.js"></script>
14     <script src="/static/line.js"></script>
15     <link rel="stylesheet" href="/static/style.css" />
16     <h1 class="display-1 border-bottom">Zewail University Dashboard</h1>
17     <style>
18       body {
19         background-color: #092635;
20         margin: 0;
21         color: white;
22       }
23     </style>
24   </head>
25   <body>
26     <div>
27       <div id="chartdiv"></div>
28       <div id="datadiv"></div>
29       <div id="bardiv"></div>
30       <div id="linediv"></div>
31     </div>
32   </body>
```


CSS:

```
1  #chartdiv{
2      position: fixed;
3      top: 0;
4      right: 0;
5      width: 50%;
6      box-sizing: border-box;
7      height: 300px;
8      margin-top: 25px;
9  }
10
11  #datadiv{
12      position: fixed;
13      bottom: 0;
14      left: 0;
15      width: 50%;
16      box-sizing: border-box;
17      height: 270px;
18      /*border-right: 2px solid #000;*/
19  }
20
21  #bardiv{
22      box-sizing: border-box;
23      top: 0;
24      left: 0;
25      width: 50%;
26      height: 280px;
27      margin-top: 10px;
28      /*border-bottom: 2px solid #000;
29      border-right: 2px solid #000*/
30  }
31
32  #linediv{
```



```
20
21 #bardiv{
22     box-sizing: border-box;
23     top: 0;
24     left: 0;
25     width: 50%;
26     height: 280px;
27     margin-top: 10px;
28     /*border-bottom: 2px solid #000;
29     border-right: 2px solid #000*/
30 }
31
32 #linediv{
33     position: fixed;
34     width: 50%;
35     height: 270px;
36     bottom: 0;
37     right: 0;
38     box-sizing: border-box;
39     /*border-top: 2px solid #000*/
40
41 }
42
```

Python:


```

1  from flask import Flask, jsonify, render_template
2  import pandas as pd
3
4  df = pd.read_csv("Student_data.csv")
5  df2 = pd.read_csv("G_with_F.csv")
6  df3 = pd.read_csv("Finance.csv")
7  df4 = pd.read_csv("Rank.csv")
8
9
10 app = Flask(__name__)
11
12
13 @app.route("/")
14 def index():
15     return render_template("index.html")
16
17
18 @app.route("/get-datachart")
19 def get_datachart():
20     classes = df["Faculty_Name"].value_counts().index
21     values = df["Faculty_Name"].value_counts().values
22
23     data = []
24
25     for i in range(len(classes)):
26         data.append({"class": classes[i], "value": int(values[i])})
27
28     return jsonify(data)
29
30
31 @app.route("/get-datatable")
32 def get_datatable():

```



```

31 @app.route("/get-datatable")
32 def get_datatable():
33     Grade = df2["Grade"]
34     Engineering = df2["Engineering"]
35     CS = df2["CS"]
36     Nano = df2["Nano"]
37     Science = df2["Science"]
38
39     datab = []
40
41     for i in range(len(Grade)):
42         datab.append(
43             {
44                 "Grade": str(Grade[i]),
45                 "Engineering": int(Engineering[i]),
46                 "CS": int(CS[i]),
47                 "Nano": int(Nano[i]),
48                 "Science": int(Science[i]),
49             }
50         )
51
52     return jsonify(datab)
53
54
55 @app.route("/get-databar")
56 def get_databar():
57     year = df3["year"]
58     income = df3["income"]
59     expenses = df3["expenses"]
60
61     datac = []
62

```



```

54
55 @app.route("/get-databar")
56 def get_databar():
57     year = df3["year"]
58     income = df3["income"]
59     expenses = df3["expenses"]
60
61     dataac = []
62
63     for i in range(len(year)):
64         dataac.append(
65             {"year": str(year[i]), "income": income[i], "expenses": expenses[i]}
66         )
67
68     return jsonify(dataac)
69
70
71 @app.route("/get-dataline")
72 def get_dataline():
73     year = df4["year"]
74     value = df4["value"]
75
76     data = []
77
78     for i in range(len(year)):
79         data.append({"year": str(year[i]), "value": int(value[i])})
80
81     return jsonify(data)
82
83
84 if __name__ == "__main__":
85     app.run(debug=True)

```

Chart 1 code: Pie Chart


```
1 function fetchDataAndUpdateChart() {
2     fetch('/get-datachart')
3         .then(response => response.json())
4         .then(data => {
5             console.log(data);
6             updateChart(data);
7         })
8         .catch(error => console.error('Error:', error));
9 }
10
11
12 function updateChart(data_df) {
13
14     console.log(data_df)
15     am5.ready(function() {
16
17         var root = am5.Root.new("chartdiv");
18
19         root.setThemes([
20             am5themes_Animated.new(root),
21         ]);
22
23
24         var chart = root.container.children.push(
25             am5percent.PieChart.new(root, {
26                 layout: root.verticalHorizontal,
27                 centerX: am5.percent(50),
28                 x: am5.percent(50)
29             })
30         );
31
32         // Create series
```



```

32     // Create series
33     var series = chart.series.push(
34         am5percent.PieSeries.new(root, {
35             name: "Series",
36             valueField: "value",
37             categoryField: "class",
38             alignLabels: false
39         })
40     );
41     series.data.setAll(data_df);
42
43
44     series.labels.template.setAll({
45         fill: am5.color("#ffffff")
46     })
47
48
49
50
51 });
52 }
53
54 document.addEventListener('DOMContentLoaded', function() {
55     fetchDataAndUpdateChart()
56
57 });
58

```

Chart 2 code: Bar and Line mix:


```

1  ✓ function fetchDataAndUpdateBar() {
2      fetch('/get-databar')
3          .then(response => response.json())
4      ✓  .then(data => {
5          |      updateDataBar(data);
6          |  })
7          .catch(error => console.error('Error:', error));
8  }
9
10
11 ✓ function updateDataBar(data) {
12 ✓  am5.ready(function() {
13
14
15      // Create root element
16      var root = am5.Root.new("bardiv");
17
18      // Set themes
19  ✓  root.setThemes([
20      |      am5themes_Animated.new(root)
21      |  ]);
22
23      // Create chart
24  ✓  var chart = root.container.children.push(
25  ✓  |      am5xy.XYChart.new(root, {
26      |          panX: false,
27      |          panY: false,
28      |          wheelX: "panX",
29      |          wheelY: "zoomX",
30      |          paddingLeft: 0,
31      |          layout: root.verticalLayout
32      |      })

```



```
32     })
33   );
34
35   // Add scrollbar
36   chart.set(
37     "scrollbarX",
38     am5.Scrollbar.new(root, {
39       orientation: "horizontal"
40     })
41   );
42
43   // Create axes
44   var xRenderer = am5xy.AxisRendererX.new(root, {
45     minorGridEnabled: true,
46     minGridDistance: 60
47   });
48   var xAxis = chart.xAxes.push(
49     am5xy.CategoryAxis.new(root, {
50       categoryField: "year",
51       renderer: xRenderer,
52       tooltip: am5.Tooltip.new(root, {})
53     })
54   );
55   xRenderer.grid.template.setAll({
56     location: 1
57   });
58
59   xAxis.data.setAll(data);
60
61   var yAxis = chart.yAxes.push(
62     am5xy.ValueAxis.new(root, {
63       min: 0,
```

```

63         min: 0,
64         extraMax: 0.1,
65         renderer: am5xy.AxisRendererY.new(root, {
66             strokeOpacity: 0.1
67         })
68     })
69 );
70
71
72 // Add series
73 var series1 = chart.series.push(
74     am5xy.ColumnSeries.new(root, {
75         name: "Income",
76         xAxis: xAxis,
77         yAxis: yAxis,
78         valueYField: "income",
79         categoryXField: "year",
80         tooltip: am5.Tooltip.new(root, {
81             pointerOrientation: "horizontal",
82             labelText: "{name} in {categoryX}: {valueY} {info}"
83         })
84     })
85 );
86
87 series1.columns.template.setAll({
88     tooltipY: am5.percent(10),
89     templateField: "columnSettings"
90 });
91
92 series1.data.setAll(data);
93
94 var series2 = chart.series.push(

```



```

94     var series2 = chart.series.push(
95         am5xy.LineSeries.new(root, {
96             name: "Expenses",
97             xAxis: xAxis,
98             yAxis: yAxis,
99             valueYField: "expenses",
100             categoryXField: "year",
101             tooltip: am5.Tooltip.new(root, {
102                 pointerOrientation: "horizontal",
103                 labelText: "{name} in {categoryX}: {valueY} {info}"
104             })
105         })
106     );
107
108     series2.strokes.template.setAll({
109         strokeWidth: 3,
110         templateField: "strokeSettings"
111     });
112
113
114     series2.data.setAll(data);
115
116     series2.bullets.push(function () {
117         return am5.Bullet.new(root, {
118             sprite: am5.Circle.new(root, {
119                 strokeWidth: 3,
120                 stroke: series2.get("stroke"),
121                 radius: 5,
122                 fill: root.interfaceColors.get("background")
123             })
124         });
125     });

```

```

127     chart.set("cursor", am5xy.XYCursor.new(root, {}));
128
129     // Add legend
130     var legend = chart.children.push(
131         am5.Legend.new(root, {
132             centerX: am5.p50,
133             x: am5.p50,
134             alignLabels: false
135         })
136     );
137     legend.data.setAll(chart.series.values);
138
139     legend.labels.template.setAll({
140         fill: am5.color("#ffffff")
141     });
142
143     // Make stuff animate on load
144     chart.appear(1000, 100);
145     series1.appear();
146
147
148     xAxis.get("renderer").labels.template.setAll({
149         fill: root.interfaceColors.get("alternativeText")
150     });
151
152     xAxis.setAll({
153         background: am5.Rectangle.new(root, {
154             fill: root.interfaceColors.get("white"),
155             fillOpacity: 0.7
156         })
157     });
158

```



```

150     });
151
152     xAxis.setAll({
153         background: am5.Rectangle.new(root, {
154             fill: root.interfaceColors.get("white"),
155             fillOpacity: 0.7
156         })
157     });
158
159
160     yAxis.get("renderer").labels.template.setAll({
161         fill: root.interfaceColors.get("alternativeText")
162     });
163
164     yAxis.setAll({
165         background: am5.Rectangle.new(root, {
166             fill: root.interfaceColors.get("white"),
167             fillOpacity: 0.7
168         })
169     });
170
171
172 });
173 }
174
175
176
177
178 document.addEventListener('DOMContentLoaded', function() {
179     fetchDataAndUpdateBar()
180 });

```

Chart 3 code: Clustered Bar Chart

```

1  function fetchDataAndUpdateTable() {
2      fetch('/get-datatable')
3          .then(response => response.json())
4          .then(data => {
5              updateDataTable(data);
6          })
7          .catch(error => console.error('Error:', error));
8  }
9
10 function updateDataTable(data) {
11     am5.ready(function() {
12
13         // Create root element
14         var root = am5.Root.new("datadiv");
15
16
17
18         // Set themes
19         root.setThemes([
20             am5themes_Animated.new(root)
21         ]);
22
23
24         // Create chart
25         var chart = root.container.children.push(am5xy.XYChart.new(root, {
26             panX: false,
27             panY: false,
28             paddingLeft: 0,
29             wheelX: "panX",
30             wheelY: "zoomX",
31             layout: root.verticalLayout
32         }));

```



```

35 // Add legend
36 var legend = chart.children.push(
37     am5.Legend.new(root, {
38         centerX: am5.p50,
39         x: am5.p50
40     })
41 );
42
43
44 // Create axes
45 var xRenderer = am5xy.AxisRendererX.new(root, {
46     cellStartLocation: 0.1,
47     cellEndLocation: 0.9,
48     minorGridEnabled: true
49 })
50
51 var xAxis = chart.xAxes.push(am5xy.CategoryAxis.new(root, {
52     categoryField: "Grade",
53     renderer: xRenderer,
54     tooltip: am5.Tooltip.new(root, {})
55 }));
56
57 xRenderer.grid.template.setAll({
58     location: 1
59 })
60
61 xAxis.data.setAll(data);
62
63 var yAxis = chart.yAxes.push(am5xy.ValueAxis.new(root, {
64     renderer: am5xy.AxisRendererY.new(root, {
65         strokeOpacity: 0.1
66     })

```

```

70 // Add series
71 function makeSeries(name, fieldName) {
72     var series = chart.series.push(am5xy.ColumnSeries.new(root, {
73         name: name,
74         xAxis: xAxis,
75         yAxis: yAxis,
76         valueYField: fieldName,
77         categoryXField: "Grade"
78     }));
79
80     series.columns.template.setAll({
81         tooltipText: "{name}, {categoryX}:{valueY}",
82         width: am5.percent(90),
83         tooltipY: 0,
84         strokeOpacity: 0
85     });
86
87     series.data.setAll(data);
88
89     // Make stuff animate on load
90     series.appear();
91
92     series.bullets.push(function () {
93         return am5.Bullet.new(root, {
94             locationY: 0,
95             sprite: am5.Label.new(root, {
96                 text: "{valueY}",
97                 fill: root.interfaceColors.get("alternativeText"),
98                 centerY: 0,
99                 centerX: am5.p50,
100                 populateText: true
101             })

```



```

102     });
103 });
104
105     legend.data.push(series);
106
107     legend.labels.template.setAll({
108         fill: am5.color("#ffffff")
109     })
110 }
111
112     makeSeries("Nano", "Nano");
113     makeSeries("CS", "CS");
114     makeSeries("Science", "Science");
115     makeSeries("Engineering", "Engineering");
116
117
118     // Make stuff animate on load
119     chart.appear(1000, 100);
120
121
122     xAxis.get("renderer").labels.template.setAll({
123         fill: root.interfaceColors.get("alternativeText")
124     });
125
126     xAxis.setAll({
127         background: am5.Rectangle.new(root, {
128             fill: root.interfaceColors.get("white"),
129             fillOpacity: 0.7
130         })
131     });
132
133

```

```

130     })
131   });
132
133
134   yAxis.get("renderer").labels.template.setAll({
135     fill: root.interfaceColors.get("alternativeText")
136   });
137
138   yAxis.setAll({
139     background: am5.Rectangle.new(root, {
140       fill: root.interfaceColors.get("white"),
141       fillOpacity: 0.7
142     })
143   });
144
145   });
146
147
148   }
149 }
150
151
152
153 document.addEventListener('DOMContentLoaded', function() {
154   fetchDataAndUpdateTable()
155 });
156

```

Chart 4 code: Step Line Chart


```
1 function fetchDataAndUpdateLine() {
2     fetch('/get-dataline')
3         .then(response => response.json())
4         .then(data => {
5             console.log(data);
6             updateLine(data);
7         })
8         .catch(error => console.error('Error:', error));
9 }
10
11
12 function updateLine(data_df) {
13
14     //console.log(data_df)
15     am5.ready(function() {
16
17
18         // Create root element
19         var root = am5.Root.new("linediv");
20
21         root.dateFormatter.setAll({
22             dateFormat: "yyyy",
23             dateFields: ["valueX"]
24         });
25
26
27         // Set themes
28         root.setThemes([
29             am5themes_Animated.new(root)
30         ]);
31
32 }
```

```

33 // Create chart
34 var chart = root.container.children.push(am5xy.XYChart.new(root, {
35     panX: true,
36     panY: true,
37     wheelX: "panX",
38     wheelY: "zoomX",
39     pinchZoomX:true
40 }));
41
42
43 // Add cursor
44 var cursor = chart.set("cursor", am5xy.XYCursor.new(root, {
45     behavior: "none"
46 }));
47 cursor.lineY.set("visible", false);
48
49
50 // Create axes
51 var xAxis = chart.xAxes.push(am5xy.DateAxis.new(root, {
52     maxDeviation:0.5,
53     baseInterval: { timeUnit: "year", count: 1 },
54     renderer: am5xy.AxisRendererX.new(root, {pan:"zoom", minorGridEnabled: true}),
55     tooltip: am5.Tooltip.new(root, {})
56 }));
57
58 var yAxis = chart.yAxes.push(am5xy.ValueAxis.new(root, {
59     maxDeviation:1,
60     renderer: am5xy.AxisRenderery.new(root, {pan:"zoom"})
61 }));
62
63 // Add series

```



```

63 // Add series
64 var series = chart.series.push(am5xy.StepLineSeries.new(root, {
65     xAxis: xAxis,
66     yAxis: yAxis,
67     valueYField: "value",
68     valueXField: "year",
69     tooltip: am5.Tooltip.new(root, {
70         labelText: "{valueX}: {valueY}"
71     })
72 }));
73
74 series.strokes.template.setAll({
75     strokeWidth: 3
76 });
77
78
79 // Set up data processor to parse string dates
80 series.data.processor = am5.DataProcessor.new(root, {
81     dateFormat: "yyyy",
82     dateFields: ["year"]
83 });
84
85 series.data.setAll(data_df);
86
87
88 // Make stuff animate on load
89 series.appear(1000);
90 chart.appear(1000, 100);
91
92 xAxis.get("renderer").labels.template.setAll({
93     fill: root.interfaceColors.get("alternativeText")
94 });

```

```
94     });
95
96     xAxis.setAll({
97         background: am5.Rectangle.new(root, {
98             fill: root.interfaceColors.get("white"),
99             fillOpacity: 0.7
100         })
101     });
102
103
104     yAxis.get("renderer").labels.template.setAll({
105         fill: root.interfaceColors.get("alternativeText")
106     });
107
108     yAxis.setAll({
109         background: am5.Rectangle.new(root, {
110             fill: root.interfaceColors.get("white"),
111             fillOpacity: 0.7
112         })
113     });
114
115
116     });
117
118     }
119
120     document.addEventListener('DOMContentLoaded', function() {
121         fetchDataAndUpdateLine()
122
123     });
124
```