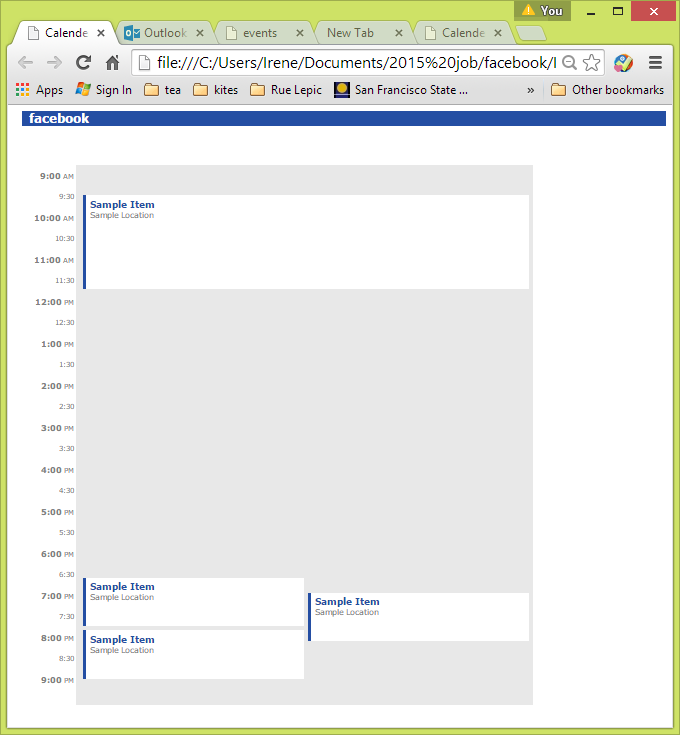
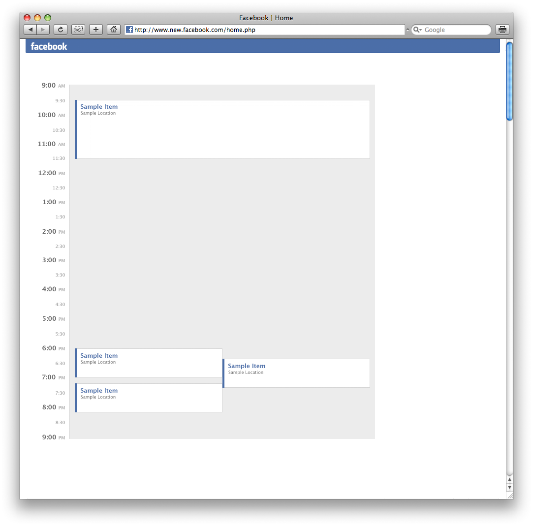
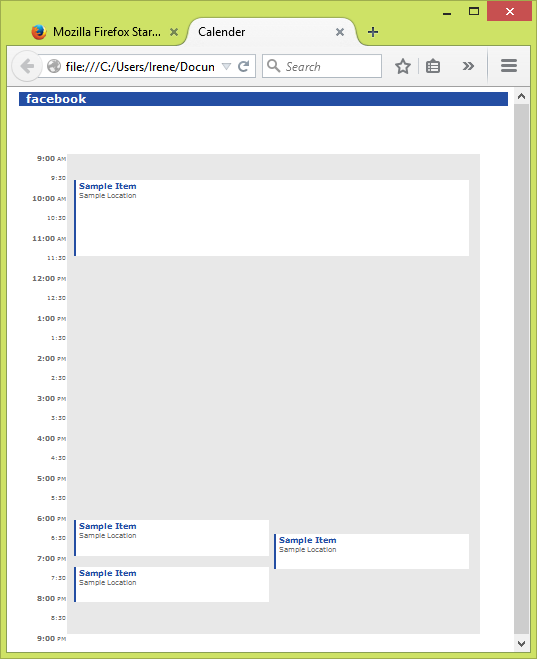
**­­**

**Event Calendar**



* **Abstract**

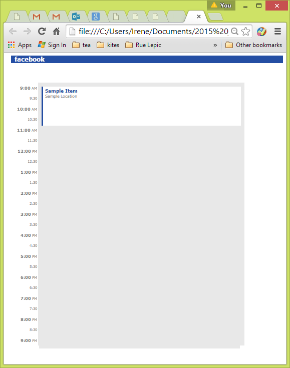
This **Event Calendar** project is based on the requirement of Facebook Front End Development job interview. The project not only tests candidates’ basic knowledge about HTML, CSS, and JavaScript but also is an interesting puzzle challenge on algorithms. Events must be layout with their start points and end points following by a time scale from 9:00 AM to 9:00 PM. Events which collide in time must be layout on the same layer but not collapse visually, and collapse events in time must have the same width. (Original Screen Shop from Facebook: Pic 1)

Pic 1: Facebook original screen shot Pic 2: Output screen shot

* **Plan and check points**

|  |  |  |
| --- | --- | --- |
| Step 1 | Design background style and time layout | http://www.elitetruckparking.com.au/wp-content/uploads/2014/03/check-mark-orange-md.png |
| Step 2 | Sort input events ordered by start points. | http://www.elitetruckparking.com.au/wp-content/uploads/2014/03/check-mark-orange-md.png |
| Step 3 | Create event Object with id, start, end, width, colNum. | http://www.elitetruckparking.com.au/wp-content/uploads/2014/03/check-mark-orange-md.png |
| Step 4 | Count Histogram by time scale vertically if event cross with the time. | http://www.elitetruckparking.com.au/wp-content/uploads/2014/03/check-mark-orange-md.png |
| Step 5 | Find the start and end edges of each group of overlap events in time. | http://www.elitetruckparking.com.au/wp-content/uploads/2014/03/check-mark-orange-md.png |
| Step 6 | Put overlap events into stack array by comparing the start and end points with the group start and end edges. | http://www.elitetruckparking.com.au/wp-content/uploads/2014/03/check-mark-orange-md.png |
| Step 7 | Count all the overlap number of events for the current event to the end in each stack. | http://www.elitetruckparking.com.au/wp-content/uploads/2014/03/check-mark-orange-md.png |
| Step 8 | Find out the max number of each group in histogram as the max column number of each event stack. The width of each overlap events in the same event stack will be equal to the container divided by the max column number. | http://www.elitetruckparking.com.au/wp-content/uploads/2014/03/check-mark-orange-md.png |
| Step 9 | Record the bottom line of each column when sign each overlap events into columns in the same group. Find the first fit position/column of each event by comparing the start points with the bottom line of each column. | http://www.elitetruckparking.com.au/wp-content/uploads/2014/03/check-mark-orange-md.png |
| Step 7 | Draw the pattern by html coding with event object information | http://www.elitetruckparking.com.au/wp-content/uploads/2014/03/check-mark-orange-md.png |

* **Design:**

1. **Frame:** File: styleSheet.css, calendar.js
2. Create **timeList** Array with **time** (number), **hour:minute** (string), and **sun** (“AM”/ “PM”).
3. Calculate time to **hour:minute** (string) format, and assign **“AM” / “PM”** only on the hour (minute must be “00”, etc. “10:00 AM”).
4. List time points on the browser with different style on the hour and the half.
5. Style designed as the sample screen shot (pic 1)
6. **Event layout:** file: styleSheet.css, script1.js
7. Use static data to get layout and style design.
8. Use **position: absolute;** to fix position of each event.
9. Sorted events by start points.

events.sort(function (a, b) {

if (a.start > b.start) {

return 1;

}

if (a.start < b.start) {

return -1;

}

return 0;

});

1. Create **event** object with event information.

function event(s, e) {

this.id = 0;

this.start = s;

this.end = e;

this.width = 100;

// this.left = 100;

this.colNum = 0;

};

1. Create **eventObjects** array by inherit event object and passing all events into event objects.

/\* eventObjects Array to store event Object \*/

var eventObjects = [];

/\* passing events into eventObjects Array\*/

for (var i = 0; i < events.length; i++) {

eventObjects[i] = new event(events[i].start, events[i].end);

eventObjects[i].id = i;

}

console.log(eventObjects);

1. Sign flag number to each event for collapse events from current event to the last one.

[

event { start=30, end=150, width=100, more...},

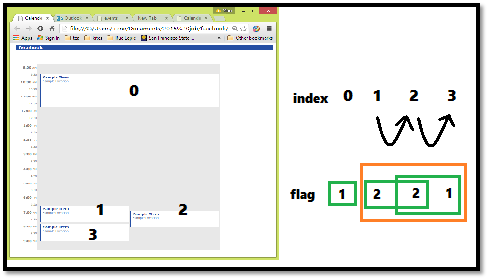
event { start=540, end=600, width=100, more...},

event { start=560, end=620, width=100, more...},

event { start=610, end=670, width=100, more...}

]

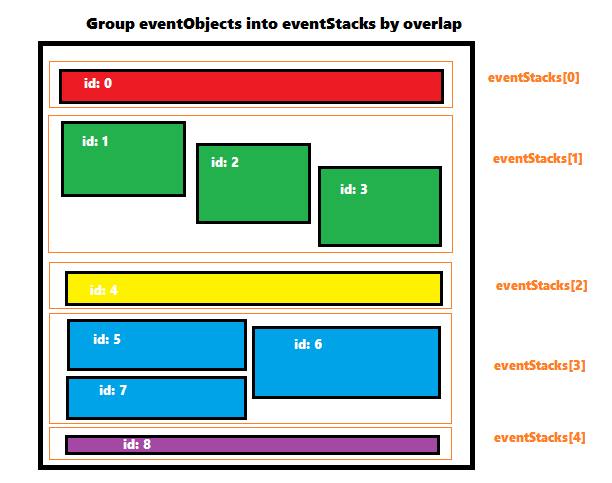
[1, 2, 2, 1]



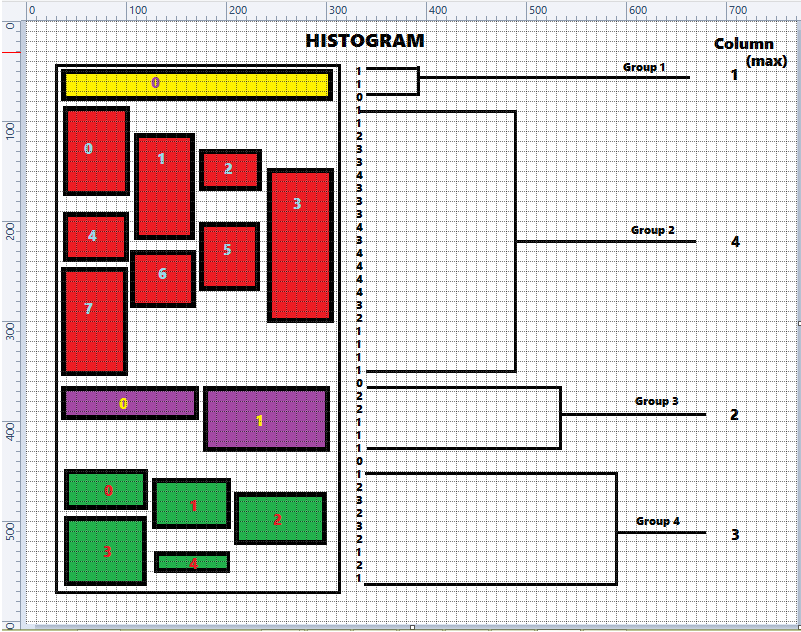
1. Put overlap events into event stack array.

|  |  |  |
| --- | --- | --- |
| **events** | **eventObjects** | **eventStacks** |
| [  Object { start=10, end=150},  Object { start=360, end=480},  Object { start=420, end=510},  Object { start=450, end=600},  Object { start=680, end=720},  Object { start=750, end=780},  Object { start=760, end=850},  Object { start=800, end=870},  Object { start=900, end=930}  ] | [  event { id=0, start=10, end=150, more...},  event { id=1, start=360, end=480, more...},  event { id=2, start=420, end=510, more...},  event { id=3, start=450, end=600, more...},  event { id=4, start=680, end=720, more...},  event { id=5, start=750, end=780, more...},  event { id=6, start=760, end=850, more...},  event { id=7, start=800, end=870, more...},  event { id=8, start=900, end=930, more...}  ] | [  [event { id=0, start=10, end=150, more...}],  [event { id=1, start=360, end=480, more...},  event { id=2, start=420, end=510, more...},  event { id=3, start=450, end=600, more...}],  [event { id=4, start=680, end=720, more...}],  [event { id=5, start=750, end=780, more...},  event { id=6, start=760, end=850, more...}],  [event { id=7, start=800, end=870, more...}],  [event { id=8, start=900, end=930, more...}]  ] |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **eventStacks[0]** | **eventStacks[1]** | **eventStacks[2]** | **eventStacks[3]** | **eventStacks[4]** |
| Id: 0 | Id:1,2,3 | Id: 4 | Id: 5,6,7 | Id: 8 |

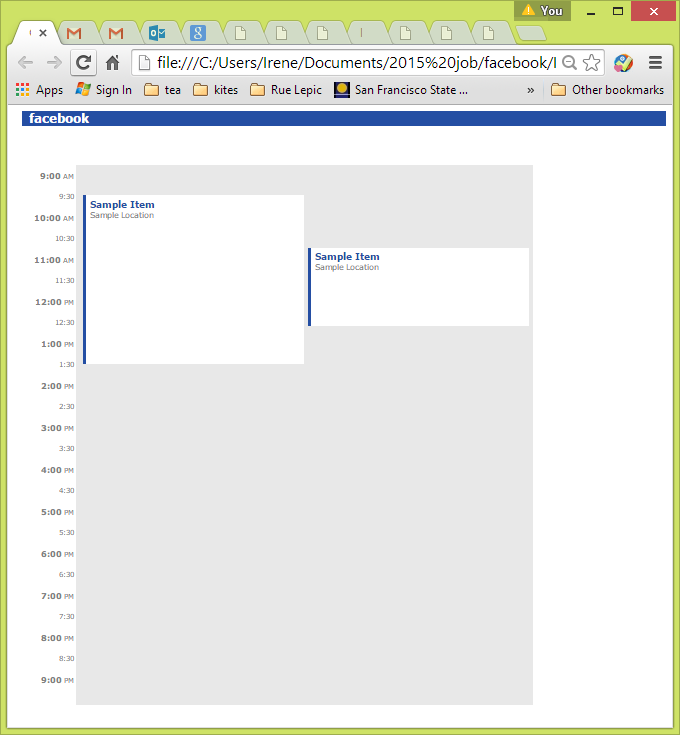
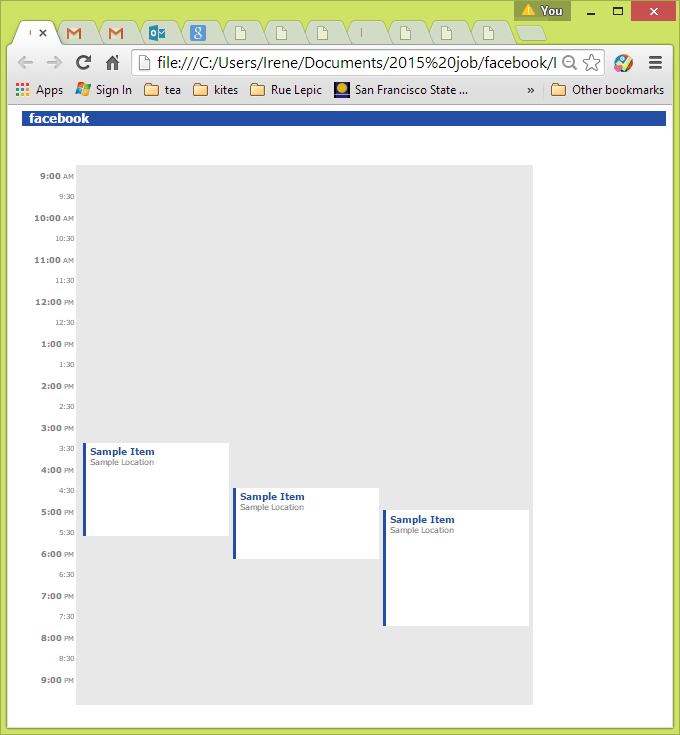
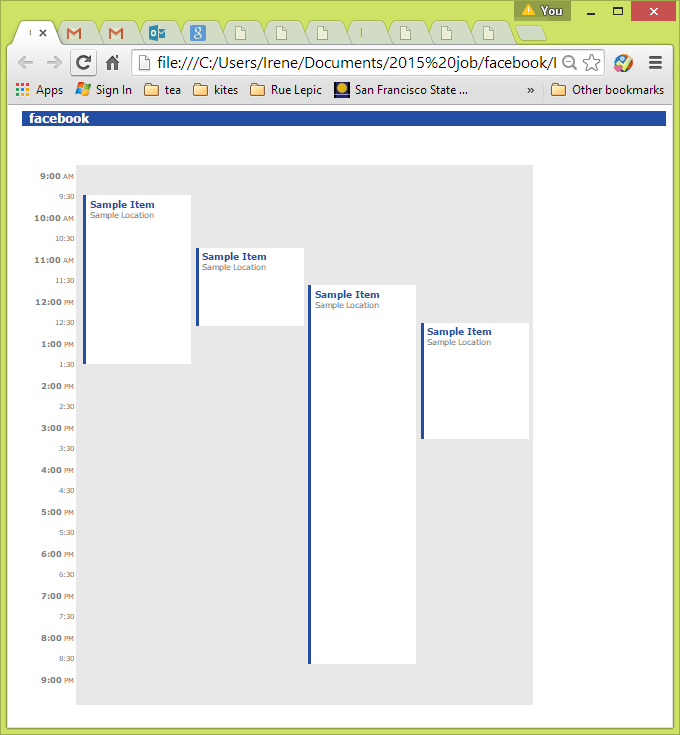
http://www.qacps.k12.md.us/ces/clipart/Carson%20Dellosa%20Clipart/Carson%20Dellosa%20Back%20to%20School/Images/Color%20Images/School%20Clip%20Art/SCISSORS.jpghttp://www.qacps.k12.md.us/ces/clipart/Carson%20Dellosa%20Clipart/Carson%20Dellosa%20Back%20to%20School/Images/Color%20Images/School%20Clip%20Art/SCISSORS.jpghttp://www.qacps.k12.md.us/ces/clipart/Carson%20Dellosa%20Clipart/Carson%20Dellosa%20Back%20to%20School/Images/Color%20Images/School%20Clip%20Art/SCISSORS.jpghttp://www.qacps.k12.md.us/ces/clipart/Carson%20Dellosa%20Clipart/Carson%20Dellosa%20Back%20to%20School/Images/Color%20Images/School%20Clip%20Art/SCISSORS.jpg

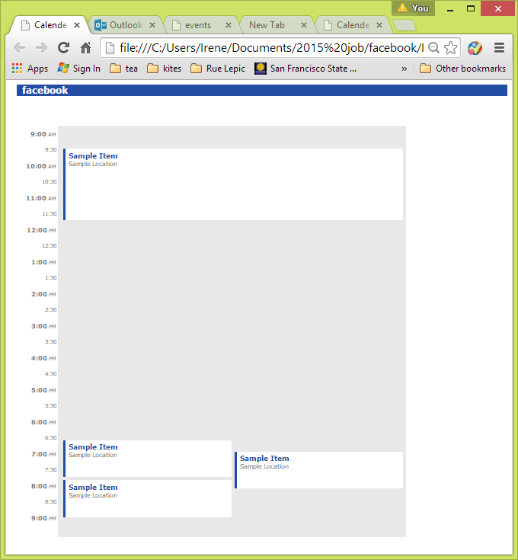
1. Histogram Layout.



1. Divided **event container width (W)** by the **column**. Sign events into each column by the order of start points from left to the right. Get the bottom height of each column, and order by the bottom line when insert the next event if it exists. Check the start point with the bottom line of each column, find the first fit column, and sign the colNum into event.

**widthOfEachEvent = widthOfEventContainer /Column**

1. Work on DOM and draw event blocks on browser.

var text = "";

console.log(eventStacks.length);

for (var i = 0; i < eventStacks.length; i++) {

for (var j = 0; j < eventStacks[i].length; j++) {

text += " <div class=\"event\" style=\" width:" + (eventStacks[i][j].width - 15) + "px; left: " + (eventStacks[i][j].colNum \* W / volNumber[i]) + "px; top: " + eventStacks[i][j].start + "px; height: " + (eventStacks[i][j].end - eventStacks[i][j].start - 8) + "px\">" +

" <div class=\"eventHead\"><strong>Sample Item</strong></div>" +

" <div class=\"eventContent\">Sample Location</div>" +

" </div>";

}

}

document.getElementById("event-container").innerHTML += text;

1. **Test on more events layout.** (Adjust position and height later for the best fit on time.)

|  |  |
| --- | --- |
| var events = [  { start: 30, end: 70 },  { start: 100, end: 200 },  { start: 150, end: 650 },  { start: 200, end: 350 },  { start: 560, end: 630 }  ]; |  |
| var events = [  { start: 30, end: 150 },  { start: 100, end: 200 },  { start: 150, end: 650 },  { start: 200, end: 350 },  ]; |  |
| var events = [  { start: 30, end: 250 },  { start: 100, end: 200 },  { start: 150, end: 650 },  { start: 200, end: 350 },  { start: 560, end: 630 }  ]; |  |
| var events = [  { start: 10, end: 150 },  { start: 360, end: 480 },  { start: 450, end: 600 },  { start: 420, end: 510 }  ]; |  |
| var events = [  { start: 30, end: 150 },  { start: 540, end: 600 },  { start: 590, end: 670 },  { start: 610, end: 650 }  ]; |  |
| var events = [  { start: 30, end: 150 },  { start: 540, end: 670 },  { start: 560, end: 600 },  { start: 610, end: 670 }  ]; |  |
| var events = [  { start: 30, end: 150 },  { start: 540, end: 600 },  { start: 560, end: 620 },  { start: 610, end: 670 }  ]; |  |
| var events = [  { start: 30, end: 250 },  { start: 100, end: 400 },  { start: 150, end: 650 },  { start: 200, end: 350 },  { start: 560, end: 630 }  ]; |  |
| var events = [  { start: 30, end: 100 },  { start: 150, end: 280 },  { start: 170, end: 320 },  { start: 200, end: 380 },  { start: 400, end: 440 },  { start: 450, end: 500 },  { start: 480, end: 580 },  { start: 520, end: 600 },  { start: 630, end: 680 }  ]; |  |
| var events = [  { start: 30, end: 600 },  { start: 150, end: 280 },  { start: 170, end: 400 },  { start: 200, end: 380 },  { start: 400, end: 450 },  { start: 450, end: 500 },  { start: 480, end: 580 },  { start: 520, end: 600 },  { start: 630, end: 680 }  ]; |  |
| var events = [  { start: 30, end: 600 },  { start: 150, end: 280 },  { start: 170, end: 580 },  { start: 200, end: 380 },  { start: 400, end: 600 },  { start: 450, end: 500 },  { start: 480, end: 580 },  { start: 520, end: 600 },  { start: 630, end: 680 }  ]; |  |
| var events = [  { start: 30, end: 400 },  { start: 150, end: 280 },  { start: 170, end: 680 },  { start: 200, end: 380 },  { start: 400, end: 500 },  { start: 450, end: 550 },  { start: 480, end: 600 },  { start: 520, end: 600 },  { start: 400, end: 600 },  { start: 450, end: 550 },  { start: 480, end: 600 },  { start: 520, end: 600 },  { start: 630, end: 700 }  ]; |  |
| var events = [  { start: 30, end: 300 },  { start: 150, end: 280 },  { start: 170, end: 480 },  { start: 200, end: 380 },  { start: 240, end: 480 },  { start: 450, end: 700 },  { start: 480, end: 600 },  { start: 520, end: 600 },  { start: 400, end: 700 },  { start: 520, end: 600 },  { start: 340, end: 460 },  { start: 450, end: 550 },  { start: 480, end: 600 },  { start: 520, end: 600 },  { start: 630, end: 700 }  ]; |  |

* **Continue**

1. Adjust the position and the scale size of time list. Adjust start point, end point, width, and height of event boxes.
2. Final test of output with different event array.
3. Wrap up file folder with report.

* **Browsers and testing tool**

1. Google Chrome
2. Mozilla Firefox
3. IE 11
4. Firebug