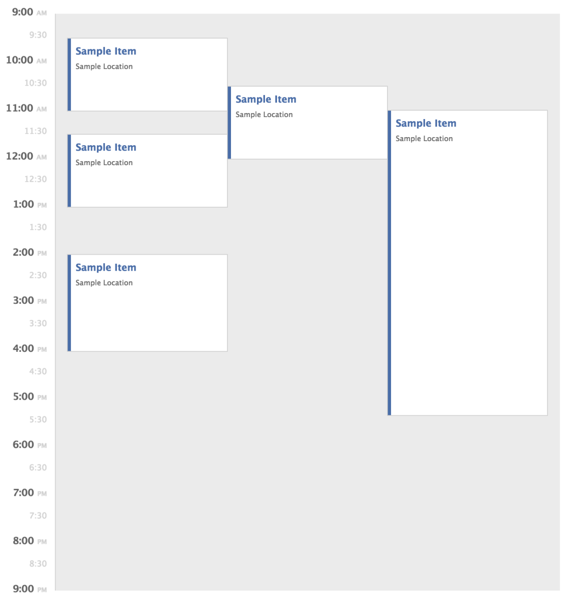
Interview Task - Calendar

REVOLT TV & Media



The Task

The annual REVOLT Music Conference packs days full of opportunities to learn, network, see artist performances, and have fun with events throughout the day. A real-world challenge that we've faced in the past is how to display all of these events in the browser with the best user experience, so our attendees know where to go throughout the day. Your task, should you choose to accept it, is to solve that challenge…

Build a web page that displays a series of events in a timeline, such as the example on the right (shown larger on the last page). The timeline should display times of a day evenly spaced, and lay out each event such that its visual representation begins at its start time and ends at its end time. You can assume that all events take place in a single day, but there may not be an event at every time, and there may be multiple events that take place concurrently. Each event's width should be equal to that of all events that it overlaps, and should attempt to fill 100% of the width of the container. Therefore, different events on the same timeline could have different widths based on overlaps. How you handle the logic for overlapping events is one of the main tests here.

Everything else is up to you. You're welcome to just build static HTML, CSS, and JS files, or render your page using a webserver. You may use any frameworks, code snippets, or build processes that you'd like (although avoiding JQuery is a plus). You can include the event data however you'd like, although it must be easy to replace. Creativity is encouraged when it comes to the interface, but make sure you're hitting the requirements.

The Requirements

- Each event's visual representation should align with its start time and end time
- Each event must display its title
- No 2 events may overlap visually
- Each event must be the same width as all events it overlaps in time
- Attempt to fill as much of the width of the container as possible at each time, up to a maximum of 100%, while obeying the above rules; e.g., an event with no overlaps should fill 100% of the width, 2 overlapping events should each fill up to 50% (unless one of them overlaps another event that has more overlaps), etc.

The Input

Attached with this document is a sample JSON file with the structure we'll use to test. It has a single object with a single key "events", whose value is an array of arbitrary length of event objects. Each event object will have the following fields:

- startTime
    - String representing the start time of the event in the format "HH:MM:SS" (e.g. "13:00:00" for 1pm)
- endTime
    - String representing the end time of the event in the format "HH:MM:SS" (eg "14:00:00" for 2pm)
    - You may assume that each event's endTime is later than its startTime
- title
    - String representing the title of the event; required to be displayed
- description
    - String representing the description of the event; no requirement on display
- address
    - String representing the street address of the event; no requirement on display
- type
    - One of: "performance," "panel," "speaker," "party," "workshop," or "demo"

Although we have a sample here, we will test your solution with additional data that may include additional edge cases, so we encourage you to build your own JSON to test.

Delivery

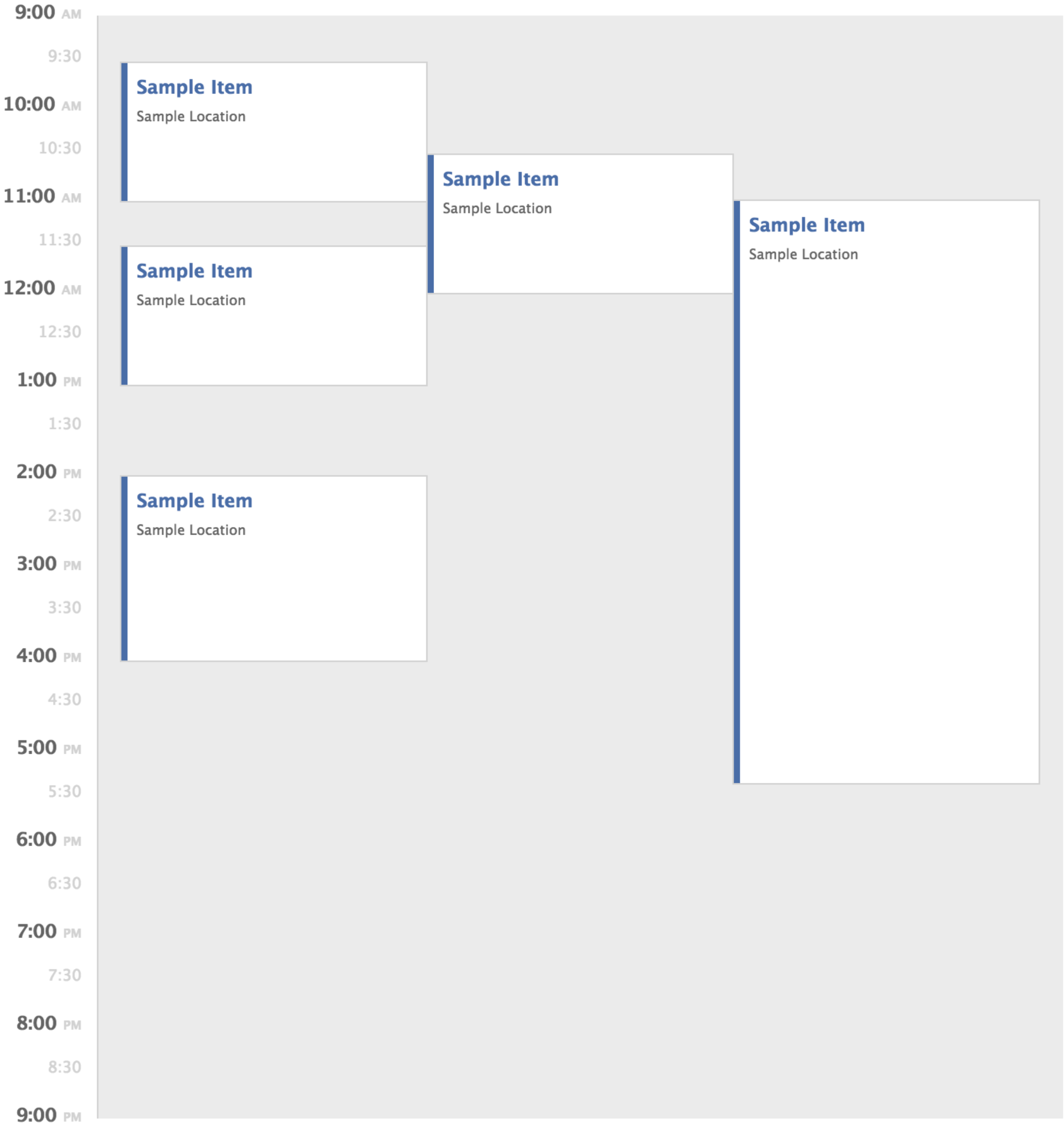
You may deliver the source code by whatever method you feel most comfortable (Github link, emailing files, etc), as long as I'm able to run it. You must include directions for how to view the webpage, and instructions for how to replace the data. You may assume that I will have Internet access when viewing.

Bonus

This is a fairly open-ended task with lots of opportunities to go beyond the specifications. **We absolutely will not penalize you** if you do not do anything extra. The task is designed to take an hour or so already, and the base task captures the main skills we're testing. With that said, there's opportunity to showcase **additional** skills, if you feel strongly that you want those skills considered in your candidacy. Here are some ideas for extensions:

- Showcase server-side abilities by connecting to a full back end with persistent storage of events
- Showcase user experience instincts by coming up with creative ways to integrate the additional information given to you (such as description, address, or type)
- Showcase design skill with the CSS and look & feel

Example:


9:00 AM
9:30
10:00 AM
10:30
11:00 AM
11:30
12:00 AM
12:30
1:00 PM
1:30
2:00 PM
2:30
3:00 PM
3:30
4:00 PM
4:30
5:00 PM
5:30
6:00 PM
6:30
7:00 PM
7:30
8:00 PM
8:30
9:00 PM
Sample Item
Sample Location
Sample Item
Sample Location
Sample Item
Sample Location
Sample Item
Sample Location
Sample Item
Sample Location