Our aim for this term project was to create a versatile website that could serve as a useful tool for RPI students, drawing inspiration from various existing web services such as QuACS, virtual calendars, and the Google landing page to develop the RPI Dashboard. We then sought to combine these various utilities into one culminating web page that could work as a dashboard on an RPI students computer. The major features we structured the dashboard around were a dynamic schedule that would alert users of approaching lectures, live weather reports, while maintaining the functionality of google's search bar. The dynamic schedule and alert system were the primary focus in terms of practicality as there aren't any preexisting softwares that offer a dedicated calendar for tracking lecture times; this feature would work well with the other utilities on RPI dashboard, and furthermore distinguish our software from other customizable search engine landing pages such as Operagx.Gabriel made a great point of how students open their choice of search engine dozens of times a day, and incorporating other services such as schedule reminders, and even weather reports could ensure students receive information that could help them make better informed decisions that they otherwise wouldn't think to check. By having their schedule constantly appear while utilizing the google search engine, students stay informed and diligent while completing other tasks, and can now more efficiently manage their time, and have a greater chance of remembering to attend lectures.

Once our team had a good idea of what we wanted to make, and lay out a feasible project plan, we began development. When starting with the first component of the dashboard, the date and time, we ran into one problem with the JavaScript. The Date API (new Date()) offered by JS had only two modes of dates: Local time and UTC time, and no way of converting between time zones. We used a different approach, the Intl API, which gives the time at a set time zone. We decided to use the Intl API at the “America/New\_York” time zone. We needed some work to convert the string array it returned into an object, but once that was done, this served as our Date() replacement.

We had initially planned for the course addition process to be much easier, where all one would have to do was export a QUACS .ics file, and upload it to the dashboard, which would read it. However, the QUACS exported calendar files were not suitable as they did not contain the CRN and did not contain the section number. This meant our program would be unable to easily figure out which specific section a given course represented. It may have been possible to figure out the section by iterating through all weekly schedules for a given course and finding the one section with an exact time match. However, this was very complex as it introduced a very large layer of state, as we would need to compare an entire week’s worth of schedule data per course. Therefore, we made the decision to only accept CRNs as input, which are guaranteed to be unique.

Our data source for courses comes from the QUACS data repository at <https://github.com/quacs/quacs-data/tree/master/semester_data>, which in turn is scraped from SIS. As this is a large file, we decided not to download it every time the dashboard was loaded, and only download it once per session whenever the settings pane was opened. For persistent storage of the configuration, we decided to use localStorage as it had a very simple interface. We had considered cookies but they were significantly more complex and harder to access via JavaScript. localStorage behaves like a key-value datastore and is therefore much easier to access. We decided to source weather data from the [OpenWeatherMap Weather API](https://openweathermap.org/api), due to its very generous free plan of 1 million free API calls per month. As per the rest of the dashboard, I decided to hardcode the weather API to only lookup the weather in RPI’s zip code.

# Summary

By the end of the project, we had implemented every feature that we designed in our mockup. We’re happy with the result, and with more time can see a bright future for RPI dashboard. For next steps, we would like to add better manageability of the bookmarks, integration of the course calendar with external services like Google Calendar, better gradients instead of the solid colors used currently, and we would finally want to make the RPI dashboard configurable enough that it can be used for colleges other than RPI. We’d also like to add more bookmarks that come naturally with the landing page, so that it’s easier for students to access the tools they need without having to search for the tools themselves. This would be especially helpful for some freshman, many who may not know about RPI’s quirks, such as Submitty. Despite encountering several challenges along the way, our team persevered and successfully developed a feature-packed dashboard that integrates various utilities, such as a dynamic schedule with lecture alerts, live weather reports, and a Google search bar, all in one easy-to-use web page that could greatly benefit RPI students by helping them manage their time more efficiently and stay informed about important information.