

## **Criteria A**

### **Scenario**

Tania Jain is currently an IBDP 2 student from my school. In her first year of DP, she found that it was difficult to manage her time between her various tasks create a schedule to finish her tasks, and feel motivated to finish them. Once, she mastered the art of time management by DP 2, she wanted to give back to the new set of students entering DP 1. With the help of a mentor, she identified other students who found similar issues and created a support group where each student could encourage one another to finish their tasks.

For the group, her mentor and her want a custom-built app that will help all the students in the group keep track of all their work, motivate themselves, and allow her to view the workload of each student so that she can motivate them. Other platforms they have explored such as Google Docs and Notion can provide them with some of their desired functions, but not all of them (Refer to Appendix A ).

### **Solution**

After speaking to the client, an app was proposed as she finds websites unappealing to work with and because most students spend drives to and from school, where Wi-Fi is limited to planning their schedule (Appendix A). The app will be divided into two parts. The first would allow students to organize all their events and view them in order of closest due date and urgency of completion, easing the burden of self-scheduling. They will also be provided with a Calendar. Each task on the Calendar would come with a reminder button to send reminder emails to the students, as we agreed that this was the best way for students to motivate themselves ever after the app was closed, something which they couldn't do with other platforms (Appendix A).

The second half would allow the client to be an administrator and view each student's Calendar, and their tasks and send reminders from her side as well, motivating them further.

### **Rationale**

The group's trial with large online websites designed to deal with various purposes failed to give them the customized experience they needed. This created the necessity for this app.

As many students with different devices will be using the app, electron was seen as the best framework to construct the app with as it can work on multiple Operating Systems. HTML, CSS, and JavaScript were used as they are widely known and easily accessible to developers. As the client requested that certain specific colors be used for tasks on the Calendar, CSS and HTML were used to allow for such customization. This will make the development of custom aspects of the app easier, and make it more extensible, something that was important to the client.

JavaScript was used for the backend as it can easily interact with the Node.js environment of the framework and because of its extensive libraries, especially for sending emails, rendering Calendars, and connecting to a database.

## **Success Criteria**

### **General Criteria:**

- Different sign-up pages for students and admins.
- Different log-in pages for students and admins.

### **Student Criteria:**

- The students will have a task management system on the website where events (anything ranging from meetings, exams, homework, classes, etc.) can be added, deleted, or edited.
- A display of all the tasks entered by the user, ordered in the form of a schedule that places tasks in order that would most benefit the student.
- A page that makes use of the deadlines and the event names inputted by the user to display these events on a calendar based on their deadlines, or submission dates.
- The students have space on the Calendar page to type out notes.
- A feature that allows students to receive reminder emails when they click a button to remind themselves to finish a task later.

### **Admin Criteria:**

- An admin window that displays all the students and admins and allows the admins to remove students if they leave the group or remove illegitimate admins.
- The admin window will also allow admins to access each student's Calendars and give them the ability to set reminders for the students.

**Word count:** 492 words