Criterion A - Planning

The Problem

My client, Mr [redacted] is the Activities Manager in my School. He spoke to me about an issue that he was having managing our Multi-Purpose Hall or MPH bookings. He said that it was tedious and inefficient for him to manually read emails from every teacher about booking the MPH and subsequently, informing other teachers and students of the booking. This had become a major issue after the renovation of the MPH as more teachers had started to use it for their lessons.

After having a detailed conversation with him (Appendix A), where he elaborated on the issues he was facing, I suggested using a web-based solution to manage the scheduling system. I explained that a web application could be accessed easily by teachers and students alike, as everyone is required to have a laptop and is connected to the school's Wi-Fi. The client was happy with this proposal. We also decided on the name I suggested for the product as **'mph_scheduler'.**

The Product and Rationale

mph_scheduler allows for easier and faster **management of the school's Multi-Purpose Hall bookings.** It is a **web-based application** that allows **teachers to book, edit and delete their lessons** that will take place in the MPH and for **students to view this schedule**. All while being **accessible only to users that are teachers or students of the school.**

As the product is an application that runs on a website, I decided to use a **Model-View-Controller (MVC)** design pattern using **JavaServer Pages (JSP)** and **Java Servlets** to implement it.

1. The JSP handles the user front-end viewing while adding dynamic features to the webpage. The **JSP Standard Tag Library (JSTL)** will be used for most Java code in the JSP pages because it is considered to be good practice as it makes code easier to read and update at later points in time.
2. The Java Servlet handles all the back-end business logic to connect with the **MySQL** database using the **Java Database Connectivity (JDBC)** API.
3. **HTML** and **CSS** will be used for UI elements, and **SQL** is used as the database querying language.

I chose to use Java as the main programming language as an OOP can handle the complexity of database handling and security features well and is also platform-independent. The product will be run on an **Apache Tomcat Server**. I will also be using the **Eclipse IDE** for writing and developing the product as it has integrated support for Dynamic Web Projects and is convenient for auto-generating methods.

There are a few existing scheduling products that are web-based such as Omnify, Boulevard Self-Booking, and Acuity (Omnify Features; "Self-Booking - Boulevard"; "Acuity Scheduling"). The problem with these products is that it does not meet my client's needs as they do not provide access for other users (like students) to see the entire schedule with all appointments. The client is not willing to spend money on these products that cost a lot as they have extra features that are not required by my client.

Success Criteria

(Please refer to Appendix A)

The product, **'mph_scheduler'** is successful if it meets the following criteria:

A	The schedule should be accessible only after login
B	Teachers can view, edit, delete and add lessons to the schedule
C	Students can **only** view the schedule (cannot edit, add, or delete lessons).
D	User is logged out after 14 days if they have not logged out manually.
E	Teachers can only add their own information. i.e. they cannot fill in information for a class other than theirs.
F	Teachers should have the convenience of not being required to manually type in lesson details.
G	The schedule must be secure to only members of the school (teachers and students).
H	There must be no limit to the number of lessons that can be added to the schedule.

Word Count: 507

Works Cited

"Acuity Scheduling - Online Appointment Scheduling Software." *Acuity Scheduling a Squarespace Company*, www.acuityscheduling.com/. Accessed 23 Sept. 2020.

Omnify Features: Calendars & Scheduling. *Omnify*, www.getomnify.com/all-features#Marketing. Accessed 23 Sept. 2020.

"Self-Booking - Boulevard." *Self-Booking - Boulevard*, www.joinblvd.com/products/self-booking. Accessed 23 Sept. 2020.