CS50 Final Project – Plugin Calendar API

Video at https://youtu.be/1dkeSseafiU

For my Final Project, I have built a plugin calendar api designed for use by small businesses for scheduling appointments and keeping track of staff availability.

My aim for this project was to clearly implement the MVC model, creating a clear distinction between the application (model) - a python file which handles database requests, the view - HTML and CSS along with a Javascript file which handles dynamic content, and the controller - a separate Javascript file which makes AJAX requests of the model and then crunches the requested data before feeding it to the view.

I also wanted to allow for straightforward third party implementation of the api, and tried (as much as possible) to reduce the endpoint to that of a typical cdn, i.e. one or two importable javascript and css files along with a summoning div id and single class instantiation.

For the purposes of the final project, I've created a basic website to surround the calendar and feed it the necessary data to correctly initialise it. In this instance, I've tailored the calendar for use for a music school, but it could just as easily be used for a company with a similar 'appointment based' setup.

I've included the typical 'Register' and 'Login' functionality. Currently, when someone registers, this creates a new user who will appear in the calendar service-provider list.

On the dashboard after logging in, I've included buttons to add/change or remove data. Currently I've only activated the ability to edit your own profile, username, password etc.

The calendar itself utilises all the expected features, e.g. days of the week, time index, day picker to select the visible day (click on day) etc. But there are a few other features that make the calendar more suitable for a business.

Within the back-end database, there is a table dedicated solely to remembering unique session data (beyond cookie functionality) for the user logged in.

There's a 'today' button which snaps to the current day, this defaults to 'day view' mode.

There are forward and backward buttons used to leaf through the days / weeks depending on viewing mode.

There's a 'display' button which opens a settings sub-page. Currently the only changeable option is ability to alter the visible timeframe. E.g. view all availability between 9:00am and 5:00pm.

There are two buttons to switch between day and week view. If the calendar's in 'week view' mode, it will present all availabilities in which the selected day is situated. As default, the user can only view one staff member's schedule at a time. In day view, the user can view multiple staff member's schedules side-by-side.

Day view has a 'notes' column in which you can add, edit and remove notes for that specific day.

The calendar features two filters to affect what the viewer sees. The calendar can be filtered by Centre, or by staff member (Teacher). The calendar relies upon Patrick Bauerochse's 'Searchable Option List' plugin. Documentation at: <https://pbauerochse.github.io/searchable-option-list/>

This allows the user not only so group-select options, but to search via text input which would be really useful if there were a long list of personnel.

The calendar features the ability to add, edit and remove a staff members availability.

In the future, I may decide to continue the project and implement appointment functionality, but for now, I feel like the calendar is already an effective tool which could be used in a real-world situation.

I've learned a great deal whilst building this application and am confident that the knowledge will benefit future work.

Other creditable libraries and technologies used:

JQuery

JQueru UI

Moment.js

Bootstrap