ProspectAve Product Guide

BILL DONG

BEN LIANG

MICHAEL MAN

JUNE HO PARK

YANG TU

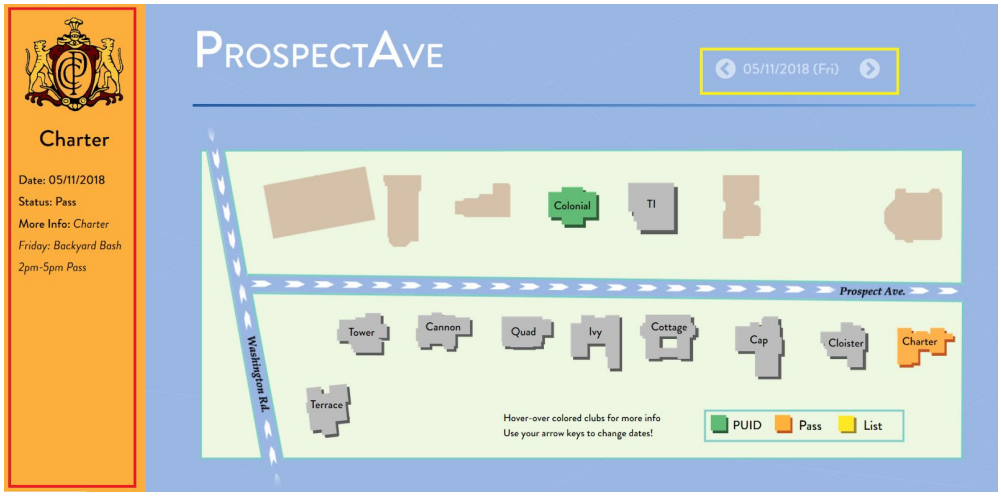
ProspectAve Student Guide

Getting Started

The ProspectAve web application can be accessed at prospectave.io. Those who access the site through a mobile device will be automatically redirected to the mobile page. To see instructions for mobile please consult the mobile section of this guide. Navigate between sections using the **sidebar**. On smaller screen sizes, this sidebar will appear as a **top-bar**, but has the same functionality. Click on a section name to be automatically scrolled to the desired section.

Map

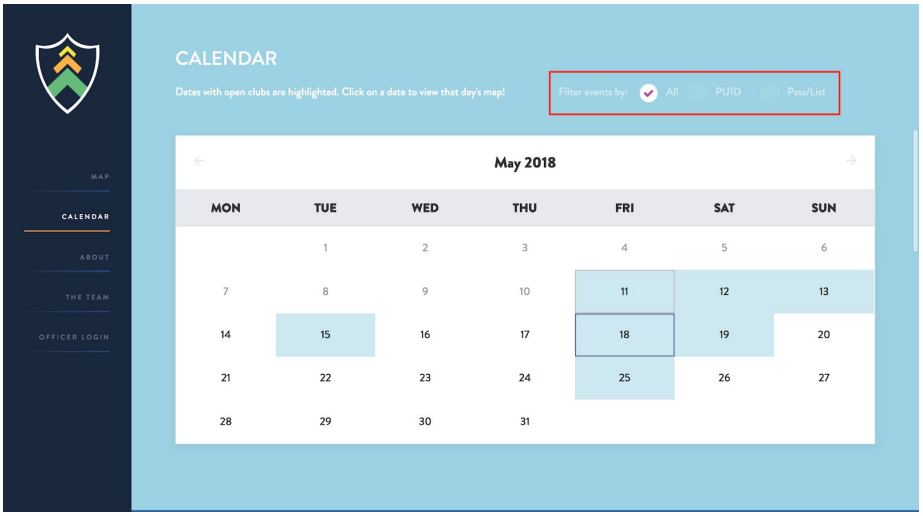
The first section of ProspectAve is a **street map** of Prospect Avenue. Each Eating Club is colored based on their status on the current date. As explained by the legend on the bottom-right corner of the map, green indicates the club is PUID, while orange and yellow represent pass and list respectively. Greyed out clubs are closed. Mouse-over a colored club to view more information on that club's opening. To change the date, click on the left and right arrow buttons in the top-right corner of the map. Alternatively, press the left and right arrow keys on your keyboard. The map will update accordingly as the date changes. The following image shows the **information overlay** when hovering over a club, in this case Charter. The yellow box indicates the **date buttons**, while the red box indicates the information overlay.



Calendar

The second section of ProspectAve is a **calendar**. Using the calendar, navigate to any future date to see the events on that date. Dates with an open club are highlighted in blue; the current date is indicated by a gray border, and the selected date is indicated by a blue border. Additionally, **filter the calendar** to show only dates with PUID events, or only dates with pass/list events. The PUID filter will highlight days with PUID events in green, while the pass/list

filter highlights days with pass/list events in orange. Below, the red box shows the location of



the filters.

MOBILE PAGE

ProspectAve has a completely redesigned **interface for mobile** in order to ensure that students can access it on the go. **Swipe left or right** on the date bar to change the date. Instead of a map, the 11 clubs are represented on a **grid**, with a panel for each club that includes an icon with the club's logo on it. This icon changes color whenever a club is open. In order to learn more about an event, tap on an icon, and an information panel will pop up. To get rid of this information and return to the main page, swipe to the left. The rest of the sections are identical to before. On the right, the date bar is boxed in yellow, while Terrace's panel is boxed in red.

PROSPECTAVE OFFICER GUIDE

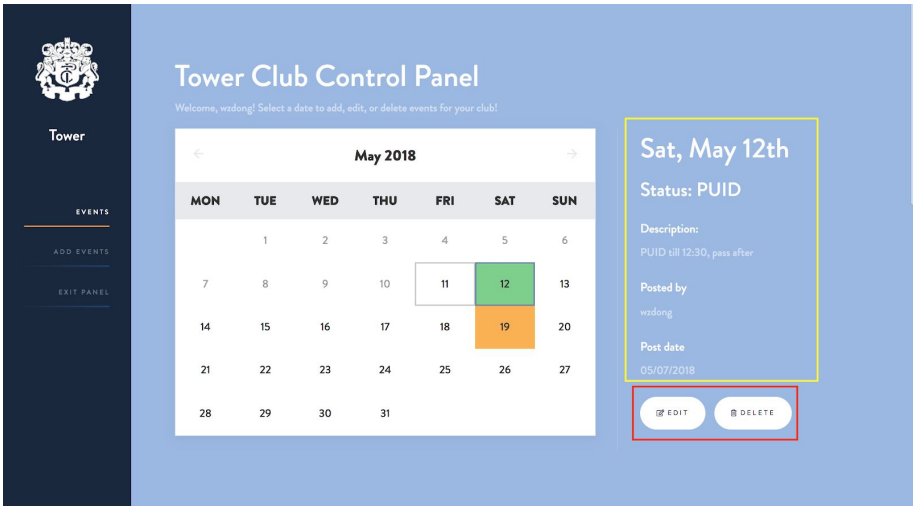
WHITELIST NETIDS

Have the President of the club send the NetIDs of all officers who have permission to edit events to the team at prospectaveio@gmail.com . ProspectAve is **CAS Authenticated** to protect eating club data, and without being put on the whitelist, you cannot edit events.

LOGIN TO OFFICER PAGE

Once your netID has been **whitelisted**, go to prospectave.io on your laptop. **Important: You cannot access the officer page through mobile.** At the prospectave.io home page, click on "Officer Login" located on the sidebar/topbar. Proceed through CAS Authentication. If your

netID is whitelisted, you should now see the control panel for your club. **Highlighted dates** on



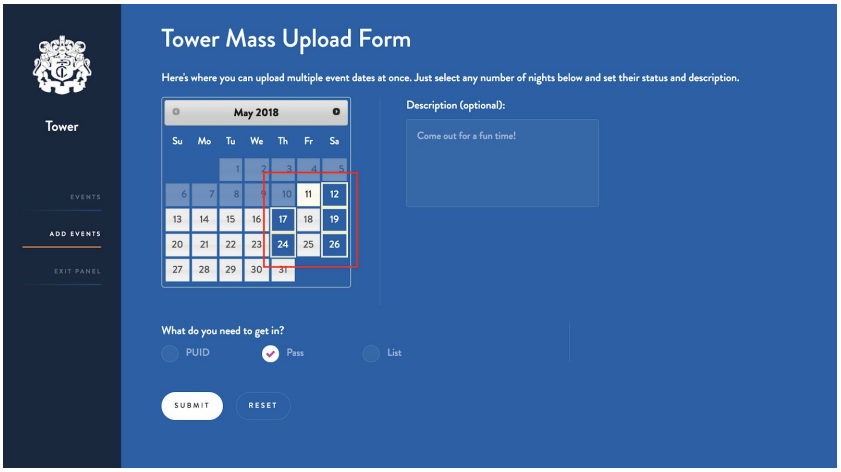
the calendar indicate there is an event for your club on that date.

Add/Edit/Delete Events

To add an event, click on the date for the event on the calendar. On the right side, you should see a panel indicating the date (Note that on small screens, this panel may appear below the calendar rather than on the right). Use the **"ADD EVENT"** button to add events. To edit or delete an event, navigate to a date with an event. Use either the **"EDIT"** or **"DELETE"** buttons depending on which you would like to do, and submit your changes when done. See above for a screenshot of the main officer page, with the event information panel boxed in yellow and the EDIT and DELETE buttons boxed in red.

Bulk Uploading Events

The **bulk upload section** allows you to add events with the same description and status to multiple dates at once. Click on the **create** button to open the bulk upload panel. On the **calendar**, select all dates you'd like to add the event to. Add a description, and select what students need to get in. You can use **"RESET"** to clear the form. Press the **"HIDE"** button to close the bulk upload section. See the right for an example of selected multiple dates with the mass upload form.



ProspectAve Developer Guide

Back-End Systems

Database

SQL Schema for table: club_status

Column	Purpose
club_name	The name of the club for this event
date	An ISO datestring representing the date of this event
poster	netID of the officer that added this event
post_date	An ISO datestring of when this event was added
status	One of PUID/Pass/List corresponding to this event
info	Any additional information that will be displayed

Officer Authentication

Officer netIDs are stored in **Auth.js** as a mapping. Each entry in Auth.js is a mapping between the officer netID and the club that they are authorized for. This is enforced with Princeton's CAS authentication. Upon opening the officer page, if there is no officer cookie, the client is redirected to CAS, where they must authenticate. Upon successful authentication, the client is given a cookie that will keep them logged in for 24 hours.

Server

As of right now, the back-end is run with a virtual private server (VPS) on GoDaddy. This server is located at **104.238.94.40**, and an administrator can FTP into the server at port 21 and upload/modify files. Furthermore, with the correct RSA keypair, one can also SSH into the server and restart the server. The server is run with pm2, which restarts automatically in case of any unexpected errors, and also keeps the Node.js runtime alive after an SSH connection is terminated. To restart the server, one can just run the command "pm2 restart all".

File Structure

The primary logic for handling HTTPS requests are in **server.js**, which connects the external HTTPS endpoints to the database IO logic. Furthermore, Auth.js holds the officer identity information and also certain functions for linking an netID to an officer identity.

HTTPS ENDPOINTS

Base: https://prospectave.io:21

Endpoint	Request Type	Input Data	Description
/status	**GET**	None	Gets all future events
/login	**GET**	None	Redirects to CAS, creates cookie
/officer_download*	**GET**	None	Gets all future events for this club
/delete*	**POST**	Date (mm/dd/yyyy)	Deletes a given event (if it exists)
/edit*	**POST**	Date and a new event (JSON)	Deletes the event in the DB for this night and creates a new one
/officer_post*	**POST**	List of events (JSON array)	Adds a new event to the DB for each element in the array

***Authenticated endpoint:** must have a valid authentication cookie

FRONT-END SYSTEMS

CSS

Our UI relies heavily on CSS. We used a theme called **Hyperspace** by HTML5 UP, the CSS for which is contained in **main.css**. The CSS for the main webapp is contained in **style.css**; it includes the CSS formatting for all of the webpage, from eating club positioning to sidebar formatting to styling for the team section. For the officer page, **officer.css** handles the formatting for the main control panel and the event bulk upload form. On the mobile page, **mobile.css** formats the date panel and all of the club icons. To create the map on the main page, we took made the shapes of all 11 clubs and the map background in Photoshop; we used CSS to position the clubs on the map, overlay them with text, and applied CSS filters to make them change colors based on the status of the club. Additionally, we formatted them so that the map and the text automatically resize depending on the size of the window. Style.css also contains the formatting for the info bar which appears upon hovering on an open club.

JAVASCRIPT

Nearly all of the information displayed on our app is due to **Javascript**, and some of its libraries. We use AJAX to download all the data from the backend. Using Javascript functions (contained in map.js and mobile.js), we take this information and put it in the infobar on both the web and

mobile apps. We also use JS to display this data on the calendar on both web and mobile (showDatesWithEvents() in map.js and mobile.js). Furthermore, JS functions handle changing the date using either the arrow keys, the arrow buttons or the calendar (shiftDate() and changeDate()), as well as deciding which dates to highlight if either the PUID or pass/list filter is selected (showPUIDEvents() and showPassListEvents()). JS also helps to display the **infobar** when you hover over a club on the map/tap on an icon on mobile (showInfo() and hideInfo()). On the officer page, JS functions display the data on both the calendar and on the event panel (downloadSuccess() in officer.js). Additionally, we used **jQuery** in **officer-ui.js** to handle many of the animations that occur on click or hover.

Several Javascript libraries were extremely important to our project. **VanillaCalendar**, available publicly here, is the template we used for our calendar on the web, mobile, and officer pages; we synced the code up with our own code in mobile.js and map.js so that changes to months on the calendar updated our data as well. Since dates were a big part of our project (showing the current date above the map, changing dates using the arrows or the calendar), we needed some sort of package to making working with dates easier. **Moment** makes it significantly easier to parse, manipulate, and display dates in JS, and it was very helpful to us. **Swipe.js** is modified from code available here, which helped implement the ability to swipe right or left on the mobile page. Finally, we connected ProspectAve to Google Analytics in order to track our pageviews and other data after we launched.

BOOTSTRAP

Our UI is highly responsive due to **Bootstrap** (and partly CSS). For window widths greater than 1100, we use a sidebar format on both the main and officer pages. If the window is less than or equal to 1100 pixels wide, the sidebar moves to the top (becomes a top-bar). If the window is less than 736 pixels, the top-bar disappears entirely. Additionally, the sidebar, map, text, and logo all scale based on the window width. We detect when a user is on mobile and automatically redirect them to the mobile landing page.

FRONT-END BACK-END INTEGRATION

We use **AJAX** to download the data from the backend. Both the map page (main desktop view) and the mobile page rely on the public API endpoint /status in order to read the event data. Upon a GET request to /status, the server will pull all relevant data from the SQL database and then return it to the client as JSON. Meanwhile, the officer page relies on the authenticated endpoints /officer_download, /delete, /edit, /officer_post to read, modify, and delete events in the database. AJAX enables us to create responsive pages that do not need to refresh in order to asynchronously read/write from the server.