**Getting Started with the UMBA project**

UMBA: INTERNAL RESERVATION SYSTEM FOR ERICSSON EMPLOYEES

August 2018

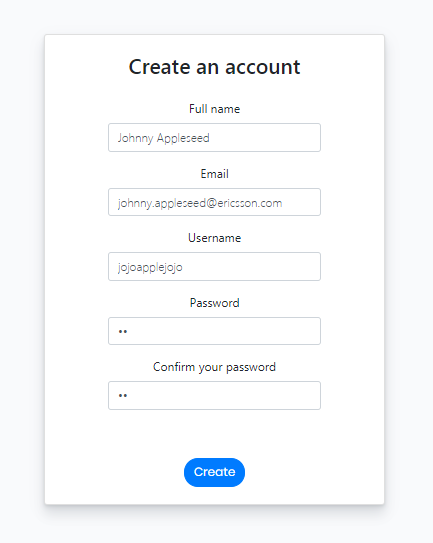
Jay ABI-SAAD, Mathieu LAPOINTE

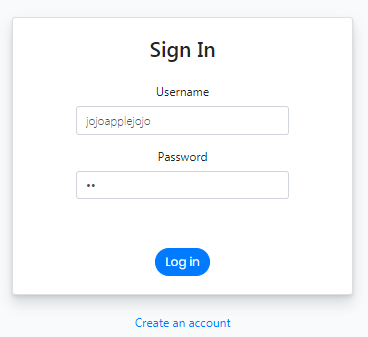
© Ericsson Canada, 2018

# Introduction

UMBA is a reservation system that gives Ericsson employees the ability to create and register for internal events. In the following lines, we’ll give you a brief overlook of the project and how it works.

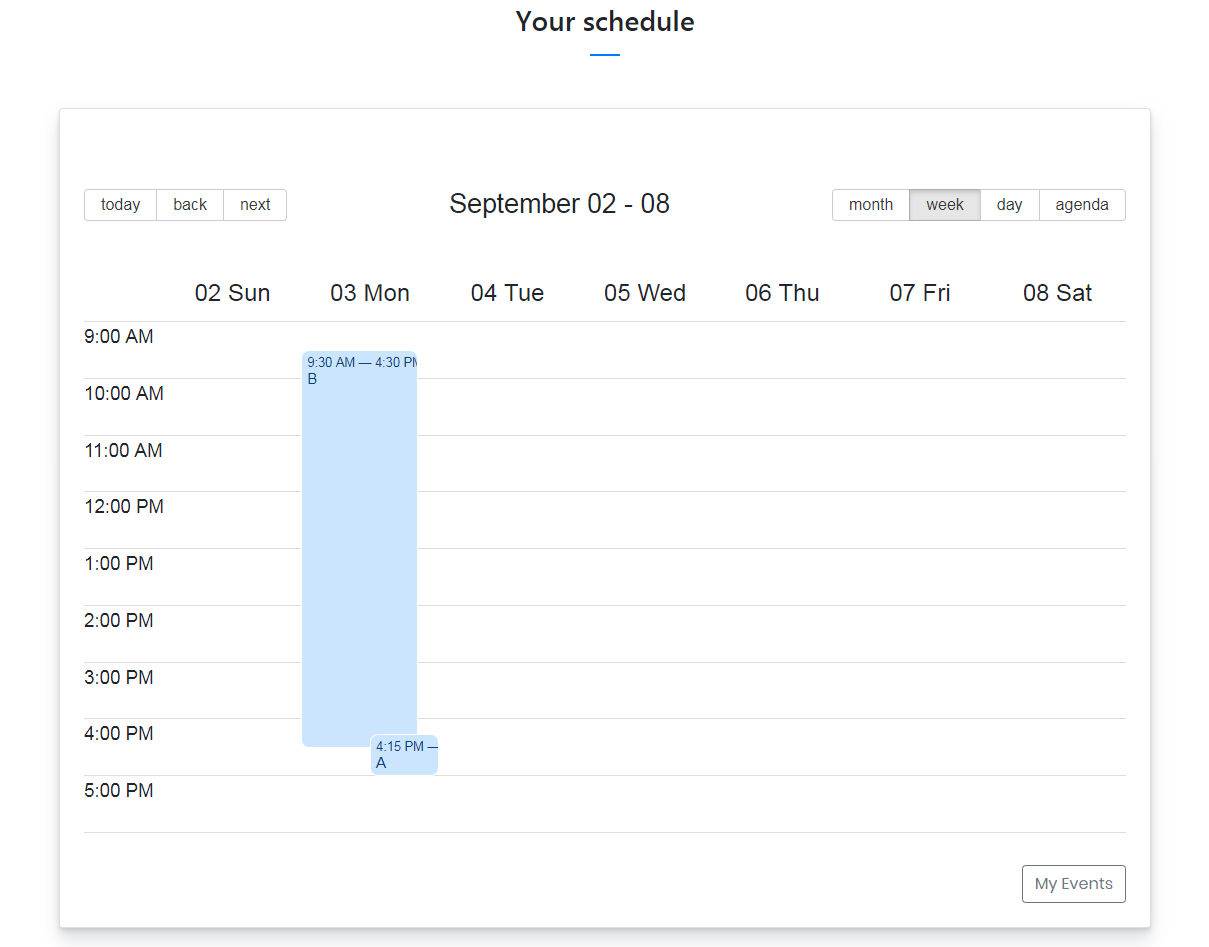
1. **Creating an account:** Users first need to create an account, after which they can then sign in and access the UMBA website.

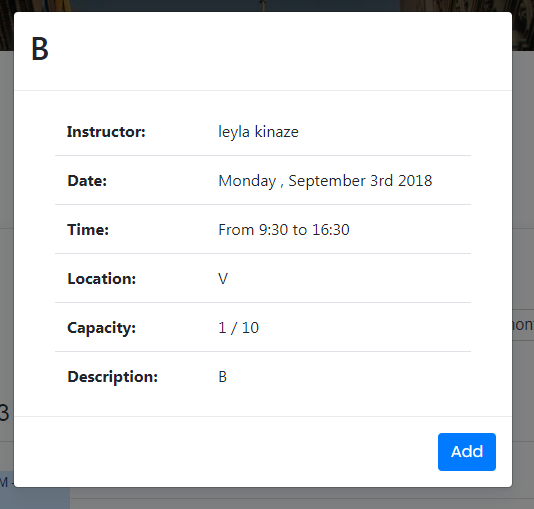


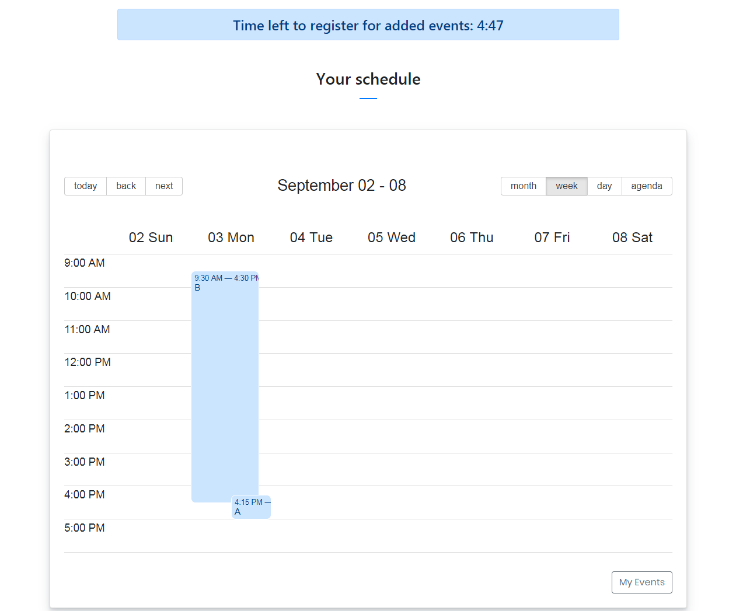
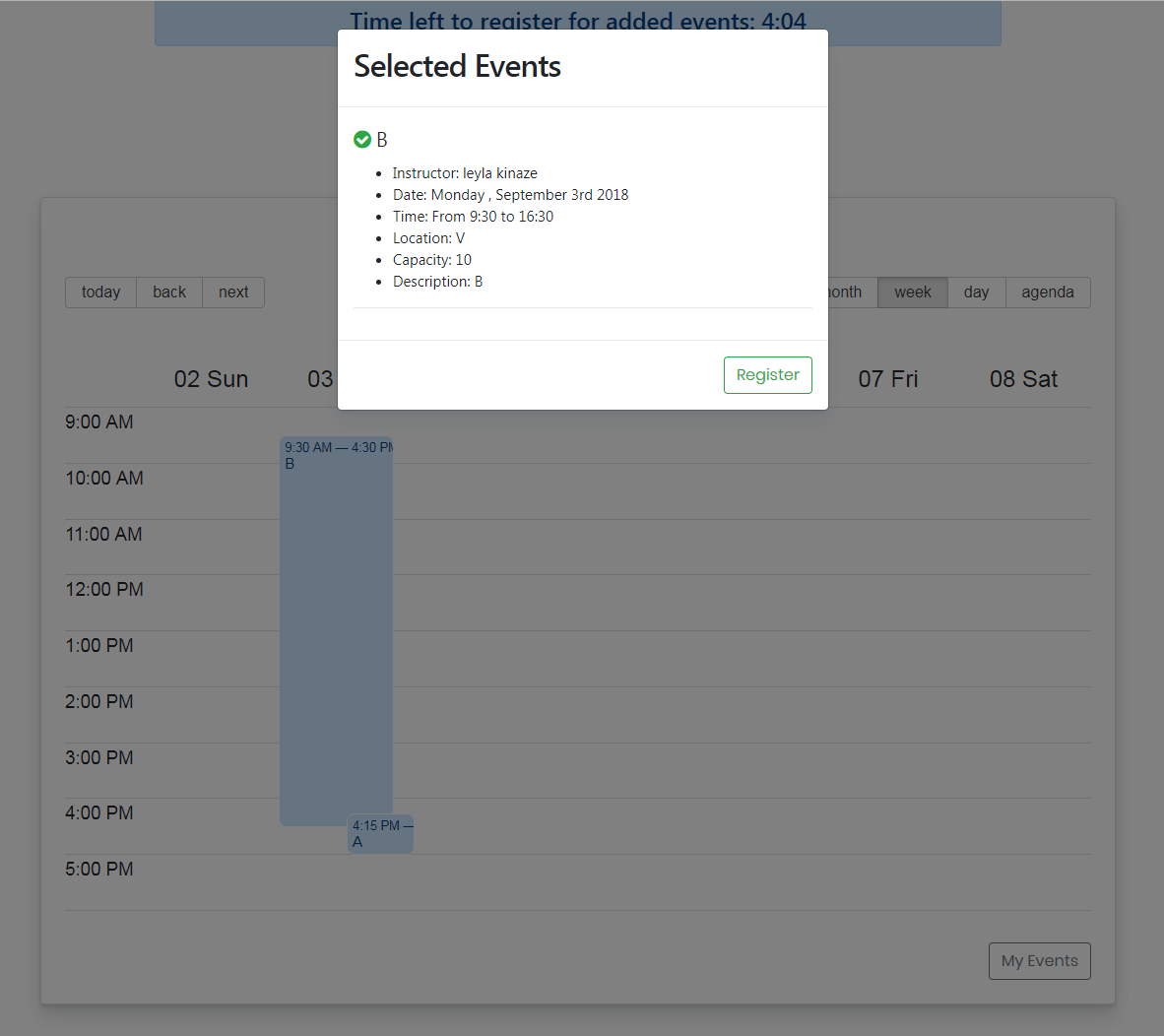


1. **User view:** By default, new users are not admins, which means they can view and register for events, but they can’t create events. Users can view the calendar by months, weeks, days and a list view (top right).

* They can also skim through the calendar with the “back” and “next” buttons. When they click on an event, a pop up opens which displays all relevant information of the selected event. While viewing an event, users can register to it by simply clicking the “Add” button. Adding an event is equivalent to reserving a spot.

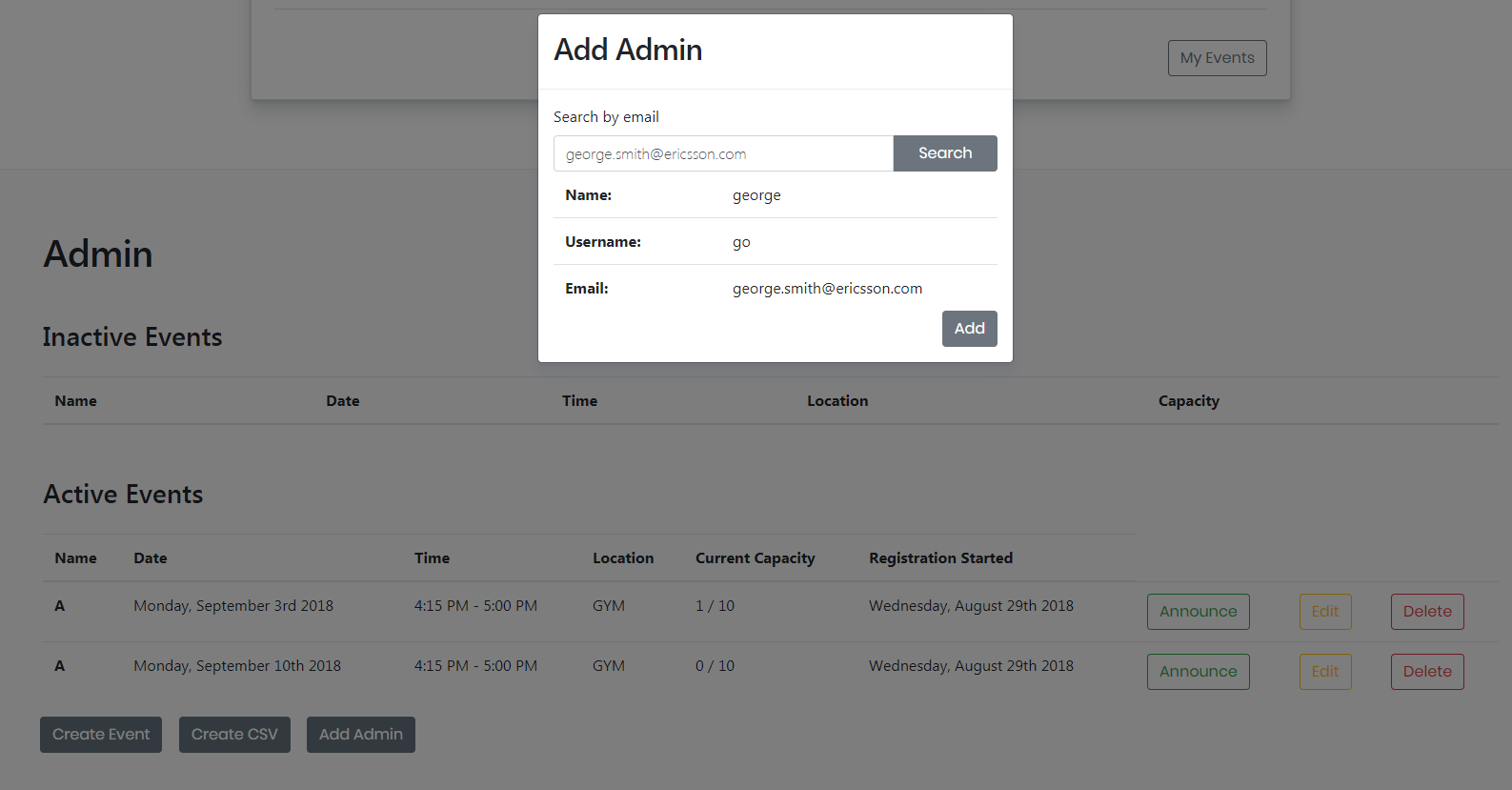


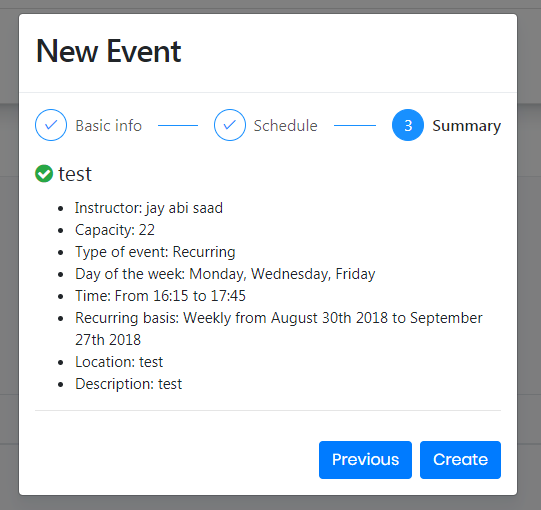
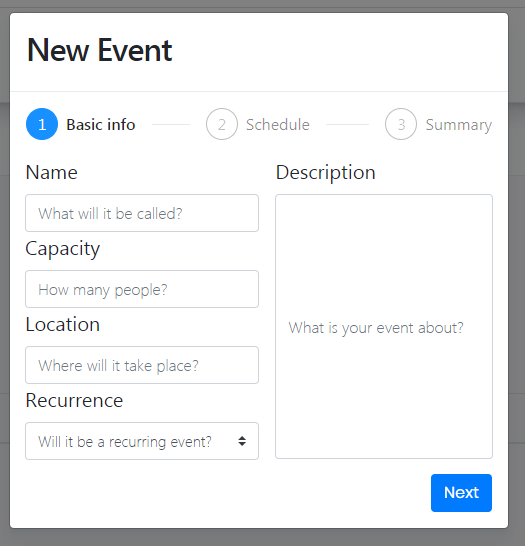


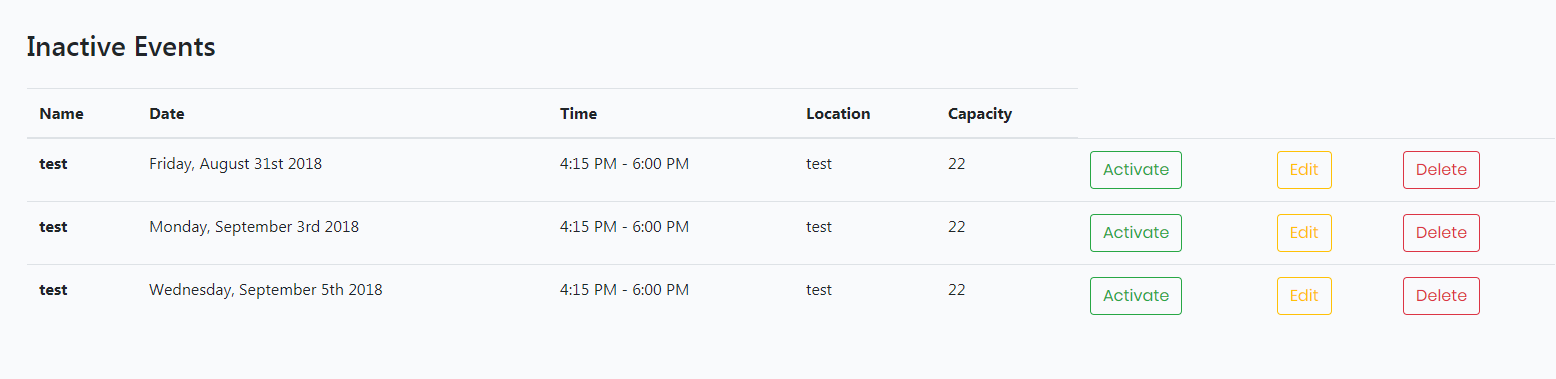
* The user can add as many events as he wants and view them by clicking the “My Events” button (bottom right). If a user has not clicked the “Register” button of the “My Events” pop up within 5 minutes of adding the event, his reserved spot will be lost (hence the counter). The 5 minutes counter resets every time an event is added, it pauses when the user is viewing his added events, and it stops when he clicks the final “Register” button.

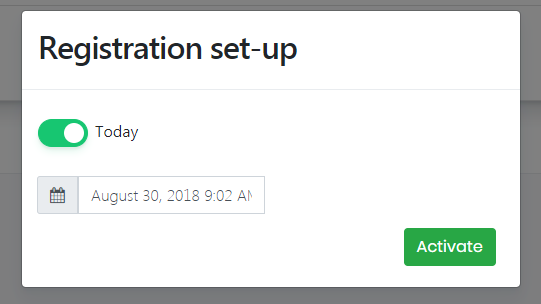
1. **Admin view:** The admin view also contains the same calendar as the user view, which means that an admin can also view, add and register for events. However, the admin view adds an admin dashboard.

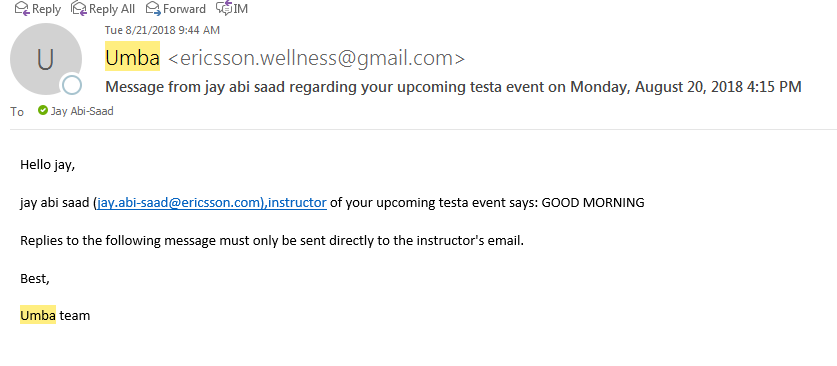
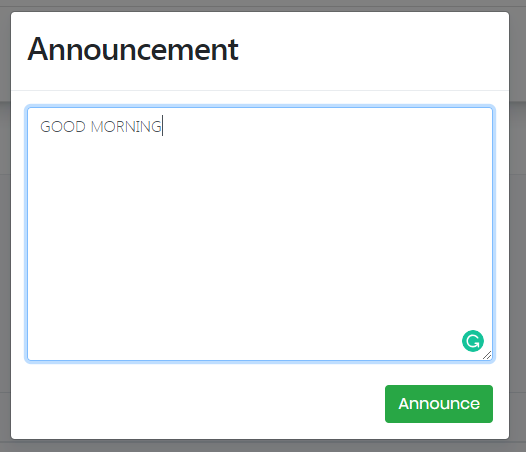
* The only way for a regular user to gain administrator privilege is to be made an admin by another admin. To do so, an admin would have to click the “Add Admin” button of the dashboard. Then, the admin would be able to search the email address of the user he wishes to grant admin privilege. If the user exists, the admin will be presented with the user’s info, and will have the option of adding him as a new admin.

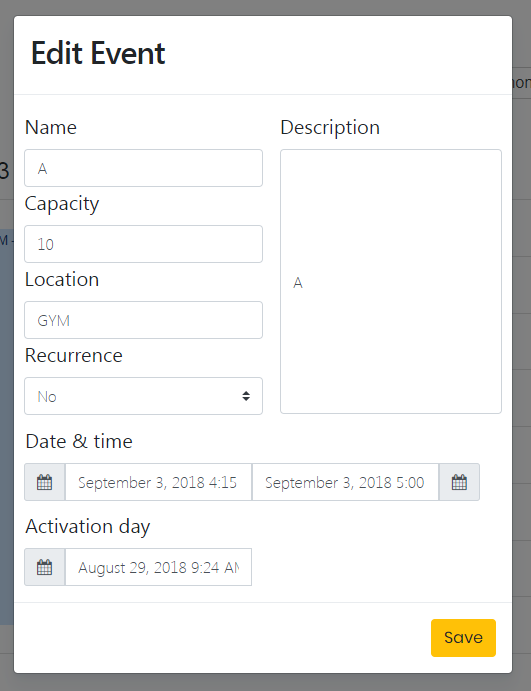


* As mentioned, only admins have the privilege of creating events. To do so, you need to click the “Create Event” button, and follow the steps to create either a recurring, or non-recurring event. Once created, the event will be shown in the calendar and in the “Inactive Events” table of the dashboard. Other users will be able to view the event, but they won’t be able to register for it until the event is activated by its admin.
* In the “Inactive Events” table of the dashboard, admins can delete, edit or activate each event. While deleting and editing are self-explanatory, the “Activate” button requires a bit more explication. Activating an event means setting the day when its registration will open. When an event is activated, it will move from the “inactive Events” table to the “Active Events” table.



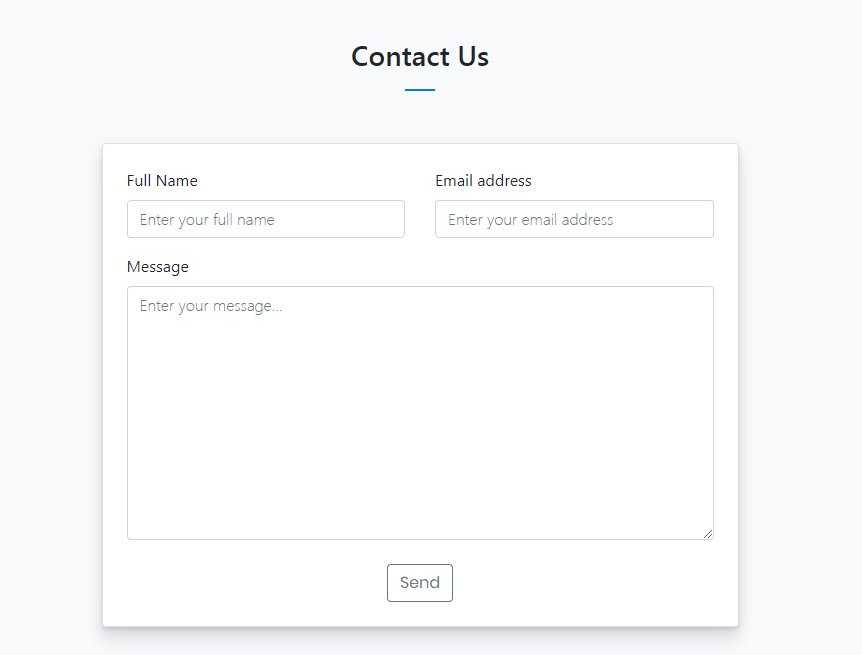


* When an event’s activation day comes, any other users will now be able to register for this event. In the “Active Events” table, admins can also announce messages to all users who are registered to an event. This message will be sent to the user’s email address provided when they created their account. Again, admins can edit and delete an event even when its activated.
* If an admin clicks one of his events on the calendar, he will be presented with the “Edit Event” pop up, but if he clicks on another admin’s event, he will only be able to view and register for it. Therefore, an admin cannot register for one of his own events. Another technicality of the admin view is that the dashboard only displays the logged-in admin’s event, where as the general calendar shows all events in the database.



* Lastly, admins can click the “Create CSV” button of the dashboard to generate an excel sheet containing all their events information.

1. **Contact Us page:** Both admin and user views feature a “Contact Us” page, which lets any user or admin send feedback to the project administrators (you and other developers working on UMBA).



# Project Set Up & Technologies Used

Before getting started, note that this project was first hosted on GitHub on the portfolio of one of the contributors. The purpose of that was to allow the contributors to keep working on this project outside of Ericsson. As such, once you get the code on Gerrit, you also need to update it with the code on GitHub because it is possible that some progress has been made of the GitHub repo. The repo is located at <https://github.com/matlapo/Umba>.

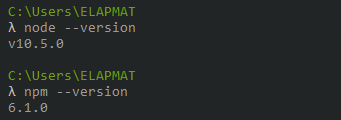
Finally, the code on Gerrit can be downloaded from <https://gerrit.ericsson.se/#/admin/projects/univ/umba>. You will need to set up your SSH key (<https://gerrit.ericsson.se/#/settings/ssh-keys>) first and require authorization to work on this project.

Our project is built with a very common technology stack: **MongoDB**, **Express** and **React**. There are hundreds of tutorials on each of these technologies available online and they are relatively easy to pick up for a beginner. We use React because of the MVVM pattern that very convenient to work with. Thus, we don’t have any dependencies on **jQuery** and we want to keep it that way. We also use **Reactstrap**, which is a subset of Bootstrap adapted to React. Before working on this project, it is highly recommended to work through the documentation of React and understand how components works and how we structure them (<https://reactjs.org/docs/thinking-in-react.html>).

This project contains both the client and the server; thus, both must be started in order to work on UMBA. If you are not familiar with **Node** and **npm** (or the whole JavaScript ecosystem in general), it would also be a good idea to learn the basics before going further. Assuming you understand those, you can first set up the backend.

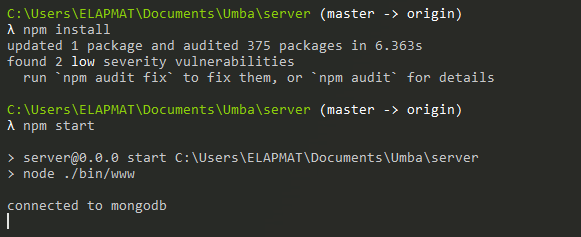
Backend set up

First, make sure you have *Node* and *npm* installed on your machine. You can check that by running the following commands:

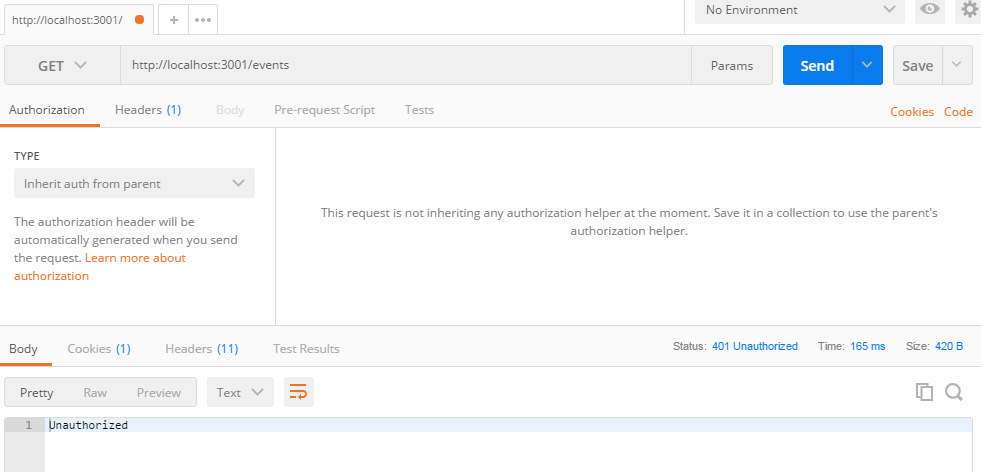


You will also need to install *MongoDB* (Community Edition) and a client for directly accessing the data, we recommend **MongoDB Community Compass**, but any client will work of course. Make sure you read a bit about *MongoDB* and how to start the **mongod** process on your machine before continuing.

Assuming MongoDB is properly set up, the server is now ready to be started. Simply run the commands below:



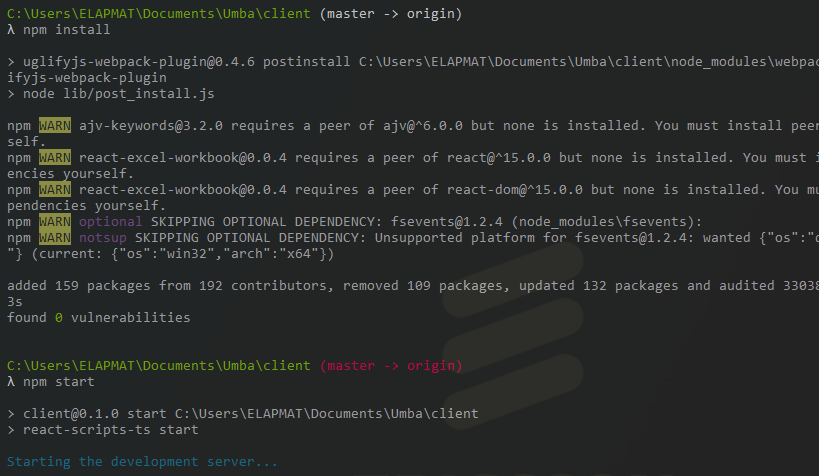
The server connects with the database on the default port 27017. You do not have any data in your database now, but you can do a sanity check by executing a GET request to the server by hand. This can be done with a HTTP client such as **Postman** (highly recommended), we also heavily use it for testing new features on the server.



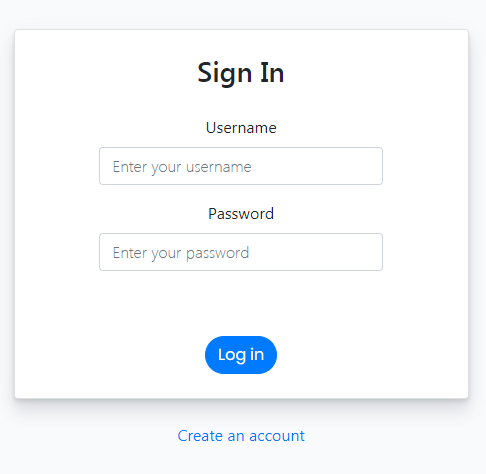
Note that you will get a HTTP 401 (Forbidden) instead of 200 OK. This is fine, it means that you need to be logged in to request data from the server. At this point, the backend for UMBA should be fully set up. We can now set up the frontend.

Frontend set up

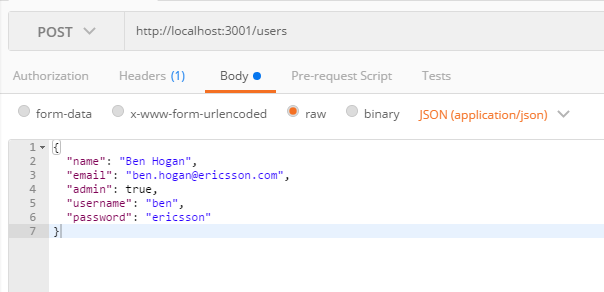
Getting the frontend up and running is much easier than the backend, simply do the steps below and you should be good to go.



You should be directed to this page:



At this point, you are done. All that is left to do is to create an account with admin access. Admins in UMBA can create events and so on. Only an admin can promote another user to admin status, but here we need an admin for testing, so we need to bypass the frontend. This can be done with Postman as follows:



If this worked, then you are good to go. You can login with this new user and start using UMBA.

# What’s left to do

1. **Integration with Outlook:** Since the beginning of this project, we wanted to connect the user’s account to their Ericsson Outlook account. Therefore, any user could register for an event and have it appeared in their Ericsson Outlook calendar. We were not able to follow through this goal, as using the Outlook Rest API required Microsoft’s approval of the project, which we didn’t have time to do. Here’s a link to get you started with the Rest API and project authentication: <https://docs.microsoft.com/en-us/previous-versions/office/office-365-api/api/version-2.0/calendar-rest-operations>
2. **Field validation:** Throughout the whole UMBA website, there’s still a lot of inputs that have no field validation from the front end, and the back end. In fact, only the input field of the “Create Event” modal contain some. In the future, you should work on adding the following field validation to the front end, AND the back end (to ensure a consistent UI, use the same alerts as the ones found in the “Create Event” modal (red text under input field)):

* **Activate event:** User should not be able to click the “Activate” button in the event activation modal if no activation date is set by the user. The user should get an error message telling him that an activation date is required.
* **Edit event:** The validation rules that are enforced in the “Create Event” modal should also be enforced in the “Edit Event” modal. Currently, breaking those rules (ex: putting a number in the event name, or putting letters in the event capacity) alerts the user with the error, but it doesn’t prevent him from clicking the final “Edit” button that saves the edited event information. Please make sure that the user is not able to click the final “Edit” button in the edit event modal if the errors are showing.
* **Announce:** When you click on the “Announce” button of the admin page, the admin user can send an email to all users that are registered to his event. Therefore, the only validation required is that the admin must not be able to announce if the message is empty.
* **Create Account:** User should only be able to create an account with a valid Ericsson email. In addition, all usernames and emails should be unique, which means that all users in the database must have different emails and usernames. Finally, both the “password” and “re-enter password” fields should match for the user to be able t create an account. If one of those rules are not respected, the user should be alerted and not able to create an account.
* **Sign in:** User should be alerted that his email and/or password are wrong if he doesn’t provide the right information.

1. **Waitlist:** If a specific event is full, users should be able to register to a waitlist. If someone decides to un-register from the event, the waitlisted user will get registered to the event automatically, and the waitlist will be updated.
2. **Unregister from events:** Currently, users can register for events, but they can only unregister from them in the “My Events” modal (before the 5 minutes counter ends). Therefore, you could add an “unregister” button on every event that the user is registered so that he can unregister from them.
3. **View and remove registered users from admin page:** Currently, admins can see the live capacity of each event. In the future, it would be great if the admin could view the registered emails for a specific event and be able to unregister them for any reason.
4. **Remove an admin:** Currently, an admin can make any other user an admin, but if for any reason an admin needs to be removed, it would be great if the “Add Admin” modal could become a “Manage Admin” modal that lets the admin user add or remove admin privilege to any user. It would also be great if the user could be notified via email when an admin granted or removed him admin privilege.
5. **Banners, alerts, and loaders:** Throughout the whole UMBA project there is absolutely no banners, alerts and spinners to indicate the user that an action is currently being processed, an action was successfully executed, or an action failed. In the future, you could add spinners when something is loading, and banners/alerts when an action was successful or not, just so that the user is better aware of what’s happening.

# Design Language

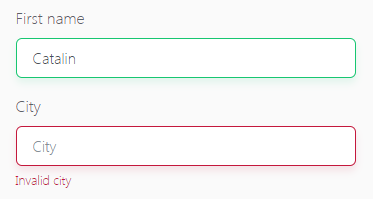
When working on the UMBA project, we would ask you to follow the same design rules to ensure UI consistency throughout the website.

**Components**: For most of the UI components, we use Reactstrap and a bootstrap library called Shards. Here’s a link to Shard’s demo website where you can grab any of their components and use it in the project: <https://designrevision.com/demo/shards/>

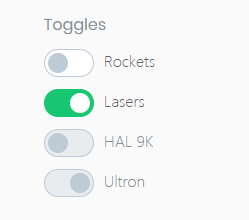
Here’s also a link to Reactstrap for other useful components that fit the design language of UMBA:

<https://reactstrap.github.io/components/buttons/>

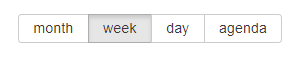
* Valid and invalid fields:

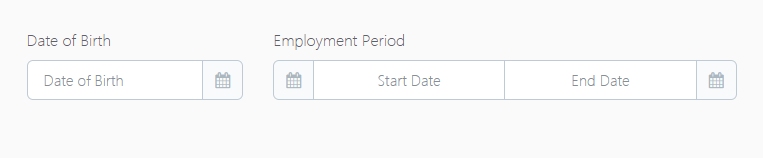


* Toggles:

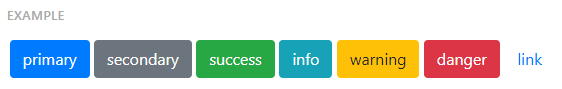


* Radio buttons (straight from bootstrap):



* Date pickers:
* Alerts:
* Buttons: We use Reactstrap “outline” buttons (top) for repeated buttons, and regular Reactstrap buttons (bottom) for non-repeated buttons.





**General rule:** Please make sure that any future development of the project follows the same design language used. It will contribute to the website’s elegance and ease of use. Lastly, the font we use is the following: -apple-system, BlinkMacSystemFont, sans-serif