

# TaskMaster

---

TaskMaster is a web application tool for volunteer organizations to automate everything which can be automated.

The web application should be comfortably viewable on both desktop and mobile screens and has a dark theme.

## Member registry

A member registry should be securely stored and easily managed.

- People can click a button in the UI to apply for membership. An account is automatically created in a database with the membership status "pending".
- An admin can manually approve a membership, at which point an email is automatically sent to the new member with an autogenerated password.
- When logging in with the password, the new member is directed to a page where membership payment can be made. After paying, the new member can change their password. After changing their password, their membership status is changed to "valid" and a date of registry is attached to the membership in the database.
- The member registry is automatically checked every day to automatically send a reminder to members whose membership expires in one week.
- All valid and expired members can log in using a button in the UI.
- Members with expired memberships can only see their own profile with the option to renew the membership.

There exists a member blacklist to prevent overload

- The blacklist consists of a list of email addresses.
- Users on the blacklist cannot apply for membership.
- Users on the blacklist cannot renew their membership.
- User memberships on the blacklist are considered "pending".

## Admin access

The list of all registered members can be accessed on a protected page only accessible to admins.

The membership page contains an overview of the members including a graph of membership count with one curve each for pending, valid, and expired members.

The full member list is accessible in a sub-section of the page where:

- Admins can create new users.
- Admins can update existing users.
- Admins can delete existing users.

The member list is shown as a table where all properties of the user are visible. The list is sortable and filterable by all properties of the users.

## User profile

All users have access to their own user profile where they can see their membership properties, including membership status, registration date, and renewal date. They can add their name, nickname, and contact information, including phone number and email address. If the user is an admin, their membership role "admin" is visible in their user profile.

- Name and contact info can be edited by the user.
- Membership can be renewed by clicking a button, at which point the renewal date is extended by the length of the membership period (e.g., one year).
- Members can see their owned event tickets.
- Members can see an overview of their booked volunteer shifts. The volunteer shifts should be exportable to Google Calendar, Outlook Calendar etc.
- Members can see a list of events they are hosting.

## General Admin Configurations

Admins have access to a configuration page where they can manage general settings for the application. This includes:

- Setting the membership period (e.g., one year, six months).
- Configuring email templates for membership approval, reminders, and other notifications.
- Managing default event tasks.
- Setting up synchronization with external calendars (e.g., Google Calendar, Outlook Calendar)?.
- Configuring UI texts and translations in the application.

## Calendar

The calendar visualizes the events happening in the organization. Each event can be clicked to take the user to the event page. The event calendar should be synchronized with that of the organization's Facebook page and other relevant social media.

## Event manager

An event page can be accessed by clicking on it in the calendar. An event page contains an event image, title, start time, end time, location, host, list of co-hosts, description, and a button to access the event's task manager.

An event can be created by clicking a button on the calendar page or a time slot in the calendar. The logged-in user is automatically set as the host. Additional co-hosts can be added through an autocomplete list of registered users, where only the nickname of the user is visible, or a comma-separated list of names. Only the host or co-hosts of the event or admins can edit and delete existing events.

The list of participants and reserves is handled automatically.

- Members can buy a ticket in the application.
- Members can see a list of their owned tickets and events in their profile page
- Members can request a refund of their ticket and the ticket will automatically be offered to a reserve. If the reserve pays for the ticket, the ticket price is automatically refunded to the original owner.

An event host, co-host, or admin can access the event's task manager. An event is split into three phases: preparation, execution, and aftermath.

## Task manager

Each event phase has its own tab in the event manager. Each tab contains a list of tasks to be done in that phase. Each task can be created by clicking a button. A task contains an event id, title, status, due date, assignee, reporter, reminder date, and description. An event host, reporter, or admin can click the assignee or reporter to show a redacted user profile showing only the user's nickname and contact info.

Tasks are automatically checked every day. On the reminder date, an email is automatically sent out to the assignee with a reminder text configurable by the event host, co-hosts, or admins.

## Default tasks

There is a list of default event tasks in the database, configurable by admins. These default tasks are automatically suggested at the creation of an event. The creator of the event can then de-select tasks as needed.

## Kanban task manager

There is a kanban-board, with columns "to do", "in progress" and "completed", containing tasks not tied to an event. All users can see these tasks but only admins can create, update, and delete tasks. All users can assign tasks to themselves and drag-and-drop a task from the "to do" column to the "in progress" column.

## Administration automation

- The database is periodically purged of stale membership applications
- Users are notified by email when their membership is about to expire
- Membership payment is handled automatically through Swish
- Monthly newsletters are automatically sent out to all members with a summary of upcoming events etc. Templates and unsubscriptions are handled by email host Sendgrid.

## Information bank

Possibly an external link to a well-organized cloud drive.

TBD

## Languages

The application should be viewable in several languages by selecting a flag at the top of the page.

The texts for each language should be stored in the database, indexed by an id of each field. Upon rendering, the texts for the chosen language are loaded into a LanguageContext. Wrappers are created around label components, typographies, links etc. which shows the text for the selected language from the LanguageContext and defaults to English if no translation exists for the field. The text should be editable by an admin. Upon edit, a new text is saved in the database for the field id.

## Contributions

To contribute to this project, go through the following steps to setup the environment on your computer:

- Create a [GitHub](#) account.
- Ask Kellen to add you to the TaskMaster project to give you access to the code and detailed plans for the project.

Download your tools:

- [Git](#)
- [Node.js](#)
- [Visual Studio Code](#)

Create an empty folder on your machine and open the folder in Visual Studio Code (hereafter: VS). Open a terminal and run the commands one after another:

```
git init
git add origin https://github.com/KellenSmith/TaskMaster.git
git fetch origin
```

Follow the steps to log in to GitHub when prompted.

You should now have a bunch of files in your previously empty folder. Install all the tools the project used by running the commands:

```
pnpm install
```

Start the application by running the command

```
pnpm dev
```

You should then be able to access the application at <https://localhost:3000>

Create a new file in the root folder called ".env" and ask Kellen for the contents. To begin working, go to GitHub and grab one of the "issues" tagged with the topic you prefer.s