# Event24 Documentation

Project Goal: To create a website that can facilitate the organization of events by various clubs in an institute.

Technologies used:

1. Frontend - HTML, CSS, Bootstrap, JavaScript (Rendered using EJS)

2. Backend - Nodejs

3. Database – MongoDB

Home page: http://localhost:8080/event24

## To use the website, follow these steps:

### For admin:

1. Create your account as mentioned in README.md.

2. Login (URL: http://localhost:8080/login)

3. You will be able to view the following pages :

 i. Events (URL: http://localhost:8080/events)

On this page, you will be able to view and modify all the events in the institute. Click on a particular event to get more details, edit details, or cancel the event.

To add a new event, click on the last box with the '+' sign. Add the event details and click on"Arrange Event". Interested users will receive an email notifying them about the event.

If an authorised user adds, edits, or cancels any event, you will receive an email informing you the same.

 ii. My Events (URL: http://localhost:8080/myevents)

This page consists of your personal events. Only you can view and modify these events. No emails will be sent for these events.

 iii. Requests (URL: http://localhost:8080/requests)

Here, you can view all the requests made by the users of the institute to add, edit, or cancel any event. You can take any one of the following actions:

Approve for this event –

The user will get the mentioned permission for this particular event. If request type is "Organise", then the user will be able to edit and cancel the event as well.

Approve for all events of "club name" –

The user will get all permissions for events which are organised by the mentioned club.

Decline - The user's request will be declined.

No action - Do nothing.

iv. Manage Users (URL: http://localhost:8080/mangeusers)

This page lists down all the users along with their permissions. You can modify the permissions of any user.

v. Profile (URL: http://localhost:8080/profile)

View and modify your account details.

## For other users:

1. Register (URL: http://localhost:8080/register)

Enter your username, password, email, and institute.

Select the clubs for which you would like to receive an email. You will be notified when a new event is arranged or if an existing event is edited or cancelled. You may also choose to receive a reminder one hour before an event is scheduled.

2. Login (URL: http://localhost:8080/login)

3. You will be able to see the following pages:

i. Events (URL: http://localhost:8080/events)

On this page, you will be able to view and modify all the events in the institute. Click on a particular event to get more details, edit details, or cancel the event.

To add a new event, click on the last box with '+' sign. Add the event details and click on "Arrange Event". Interested users will receive an email notifying them about the event. They will also get updates if the event is edited or cancelled.

Note that you must have permission to arrange the event. If you do not have it, then a request will be sent to your institute admin. Additionally, you may also request to edit or cancel an existing event. You will get an email about the admin’s action to your request.

Once you have the required permission, you can take the corresponding action. Please note that if you misuse your permissions, then the admin can deny you the given permissions.

ii. My Events (URL: http://localhost:8080/myevents)

This page consists of your personal events. Only you can view and modify these events. No emails will be sent for these events.

iii. Profile (URL: http://localhost:8080/profile)

View and modify your account details. You can also edit your interests here.

## Backend Procedure:

1. Register

The user is asked to enter the following details:

i. Username – Each user has a **unique** username. This is ensured while submitting the details. If a username already exists, the user is asked to change it. The username **cannot** be changed once the user is registered.

ii. Password – For login. It can be changed after registration, from the profile page.

iii. Email Address – Main mode of communication. Emails are sent regarding events, and for response to the requests made.

iv. Institute – The institute must first be entered into the database. Instructions are given in the README file.

v. Interests – The clubs in which the user is interested. User only receives mails for the clubs selected, though all the events of the institute are visible on the events page. The interests can be changed from the profile page.

vi. Reminder – User will receive reminder one hour before event is scheduled. This preference can also be changed from the profile page.

Upon registration, the following document is stored in the database:

```
{
    username: “username”,
    password: “password”,
    email: “email”,
    institute: “institute name”,
    interests: [interested clubs],
    permissions: [],
    myevents_id: 1,
    myevents: [],
    remind: true/false
}
```

2. Login

User is logged in by finding the document in the database using the entered credentials. For rest of the session, the user is identified using a cookie, named “user”. It has a maximum lifetime of 30 minutes, to enable auto logout.

If the user voluntarily logs out, then this cookie is destroyed. If this cookie does not exist, then all the other web pages will redirect the user to the login page. If a user enters the url for an admin page, for example requests, then too he will be redirected to the login page.

3. Profile

This page displays all the details of the user. The password, email address, and preferences regarding events can be changed.

4. Events

All the events of an institute are stored in the events array inside the institute document in a MongoDB database. The array consists of objects, with each object to represent an event. The document is inside the collection “institutes”, which is inside the database “institute_data”. The event object consists of the following attributes:

```
{
        id: unique number to identify the event,
        name: “event name”,
        club: “organizing club”,
        venue: “event venue”,
        time: “date and time of event”,
        description: “event description”,
        accepted: true/false (depends on whether organizer has permission of admin or not),
        permissions: [array to know which users have permission for the event]
}
```

The id field is stored in the institute document. It is incremented every time an id is assigned. This ensures that no two events ever have the same id.

- New event :

A cron-job is scheduled which is responsible for deleting the event from the database. The event is deleted at its scheduled time (time field inside the object). If the organizer does not have permission to arrange the event, then the event will not be displayed until the request is accepted by the admin. When it is accepted, an email is sent to all users who are interested in the events of the organizing club. If the user already has permission, then the email is sent immediately. Also, users will now be able to view the event on the website. If any user has opted for reminder, then the same mail is once again scheduled using node-cron.

- Edit already existing event :

Users with permission can edit already existing events. When the event is edited, the following steps are taken:

i. The old event object is deleted from the array.
ii. The edited event is now treated in a similar way that a new event is treated.
iii. The object is inserted into the events array, with a new id. A new id is necessary so

that the cron-job to remove the event does not remove this event, since its time may have changed. Instead, a new cron-job is scheduled, using the new id.

iv. An email is sent to users about the update, and once again the removal and reminder is scheduled.

Otherwise, users can request to edit the event. Once approved by the admin, they can take appropriate action.

- Cancel already existing event :

Users with permission can cancel already existing events. When the event is cancelled, it is simply removed from the events array. An email is sent to interested users. When the cron-job of removing the event is executed, it will do nothing as there would be no event with the required id. Same would be the case with the reminder email.

5. My Events

This section works in the same way as the events section. The difference is that there are no emails to be sent, and no permissions required.

**Note:**
Since most of the users would not have done OAuth2 authentication, they would be unable to send emails from their gmail account through nodemailer. Emails can actually be sent from the admin's account to the users regarding these events. But, since it would compromise the privacy of the users, this feature is not added.

The events are stored in the form of an object in the myevents array inside the user's document in the collection "users", inside the database "userdata".

These events are also assigned a unique id, and are removed using node-cron.

6. Requests (Only for admin)

This page displays all the requests made by the users of the institute. There are 3 types of requests:

i. Request to organize an event
ii. Request to edit an event
iii. Request to cancel an event

These requests are stored in the permissions array inside the event object. A request is an object with the following attributes:

```
{
        username: "username of requesting user",
        email: "email address of requesting user",
        type: "type of request" (Namely "organise_request", "edit_request", and
                                "cancel_request")
}
```

These requests are displayed in a tabular manner to the admin. If approved, the type field inside the object is changed to "organise", "edit", and "cancel", respectively.

Events are removed from the events array if either an organize request is declined, or a

cancellation request is approved. Otherwise, if the event is accepted, then a mail is sent to interested users.
Using the “email” field in the request object, a mail is sent to the requesting user about the action taken by admin.

## 7. Manage Users (Only for admin)

This page displays all the users along with their permissions. The permissions are stored inside the user’s document in the form of an array, named “permissions”. The admin can edit these permissions. This array is then updated in the database.