

# SponsorMatch Documentation

SponsorMatch is a Content Management System (CMS) website centered around internships. While the main purpose of the website is to serve as a catalog for companies that offer internships to international students, currently sponsormatch is serving as a blog website on topics centered around internships. The website allows users to view and interact with blogs written by fellow students and organizations. The website has four pages, which will be discussed in the Front-end section of the documentation.

## #Front-End

The front end of the SponsorMatch website includes the following pages:

[index.html](#): This page displays clickable snippets of the blogs. The snippet includes the title, subtitle, the author's first and last name and the date the blog was created.

### Features:

**Full Blog:** When the snippet of the blog is clicked, it will lead to the post page displaying the full blog.

**Author Info:** When the name of the author is clicked, it will to the user page that displays full information about the user.

**Pagination:** The index page includes features for the user to scroll through all of the posts made on the website by simply pressing the next button or the previous button. These will result in the index page sorting through the posts made in the order of the first post to latest post. As new posts are added, more spots and pages are added and the user will be able to continue to browse. If the user tries to go past the last page of posts the index will resort back to the first page looping through the posts again from the beginning.

[post.html](#): This page displays the full blog including the title, subtitle, author, date, and comments of the blog. It has a feature that allows users to comment, like, and delete their comments. It also includes a feature to edit or delete the blog post. When either the delete or edit buttons are pressed it will redirect the user to the pages accordingly.

[user.html](#): This page contains the user information such as first name, last name, username, email address, and whether or not the user is actively looking for an internship or if they are casually browsing.

[about.html](#): This page contains a description of the purpose of the sponsormatch website.

The front-end functionality of the website is implemented in [app.js](#). The file includes the blogs object with the following functions:

- **index**: This function calls the index function in database.js to retrieve all the blogs and display them on the index page.
- **detail**: This function calls the detail function in database.js to retrieve a specific blog and displays it on the detail page.
- **displayUser**: This function calls the detail function in database.js to retrieve a specific blog and displays only the information about the user on the user page.
- **create**: This function calls the create function in database.js to create a new blog based on the input information from a user.
- **update**: This function calls the update function in database.js to edit a specific blog and display the information and enables the user to edit a blog in the edit page.
- **addComment**: This function calls the addComment function in database.js to manage the functions of adding liking and deleting a comment on the blog post page.

## #Back-End

The back end of the SponsorMatch website includes the following files:

- **api.js**: This file contains the API routes and functions for interacting with the database.
- **database.js**: This file contains the functions for interacting with the remote database hosted on JSONblob.com through the api.js functions.

The back-end functionality of the website is implemented in api.js. Which includes the following functions:

- The first function is GET. It takes a document ID and a callback function as parameters and uses the axios library to make an HTTP GET request to the specified JSON blob endpoint with the document ID appended to the endpoint URL.

- The second function is PUT, which takes a document ID, data, and a callback function as parameters. It uses the axios library to make an HTTP PUT request to the specified JSON blob endpoint with the document ID appended to the endpoint URL and the provided data in the request body.

These functions provide a way to get and update JSON data stored on jsonblob.com using HTTP requests.

## **#Database**

The database for the SponsorMatch website is hosted remotely on JSONblob.com and accessed through the api.js file.

The database schema consists of an array of objects representing blogs and the structure includes 3 entities: blogs, comments and users. The blog has the properties author's first and last name, username, title, subtitle, email, date, blog text, and comments, while the comments array has the properties first and last name, comment likes, date published and the comment text. Finally, the user has the properties sharing with the blog including first and last name, username, email, and whether they are looking for internship or not.