**Hackathon**

**Report**

Table of Contents

[Table of Figures 2](#_Toc130836270)

[Introduction: 4](#_Toc130836271)

[W3C Standards: 4](#_Toc130836272)

[Proof 5](#_Toc130836273)

[Website Accessible: 5](#_Toc130836274)

[Search Accessibility 6](#_Toc130836275)

[Button Accessibility 7](#_Toc130836276)

[Text Accessibility 11](#_Toc130836277)

[Accessibility in Navbar 11](#_Toc130836278)

[Legal Consideration: 12](#_Toc130836279)

[Security Consideration: 12](#_Toc130836280)

[Version Control: 13](#_Toc130836281)

[API Routes: 15](#_Toc130836282)

[References: 16](#_Toc130836283)

# Table of Figures

[Figure 1 : W3C Test 5](#_Toc130644223)

[Figure 2 : W3C Test 2 5](#_Toc130644224)

[Figure 3 : Search Accessibility 6](#_Toc130644225)

[Figure 4 : View Button 8](#_Toc130644226)

[Figure 5 : Delete Button 8](#_Toc130644227)

[Figure 6 : Update Button 9](#_Toc130644228)

[Figure 7 : Update Form 10](#_Toc130644229)

[Figure 8 : Input Validation 13](#_Toc130644230)

[Figure 9 : New Rep 14](#_Toc130644231)

[Figure 10 : CMD 14](#_Toc130644232)

[Figure 11 : Commit 15](#_Toc130644233)

# **Introduction:**

A hackathon site is a stage that is intended to have and work with a hackathon occasion. It fills in as a focal center point for members to enlist for the occasion, structure groups, submit projects, and get updates and data about the hackathon. The site can likewise incorporate a timetable of occasions, data about backers and judges, assets for members, and a way for participants to furnish criticism and speak with coordinators.

Notwithstanding these elements, security is a basic thought for a hackathon site. Given the idea of a hackathon, where members are much of the time testing their coding and improvement abilities by making new tasks, safeguarding the protected innovation and individual data of all participants is significant. This can incorporate executing secure client verification and information encryption conventions, consistently checking the site for potential security weaknesses and giving clear rules and approaches to members regarding the utilization of the site and the assurance of their data.

# **W3C Standards:**

W3C (Internet Consortium) principles are rules and proposals for building and planning sites and web applications. (Paris, M., 2006) These guidelines are laid out by the W3C association, which is answerable for advancing and creating web advancements.

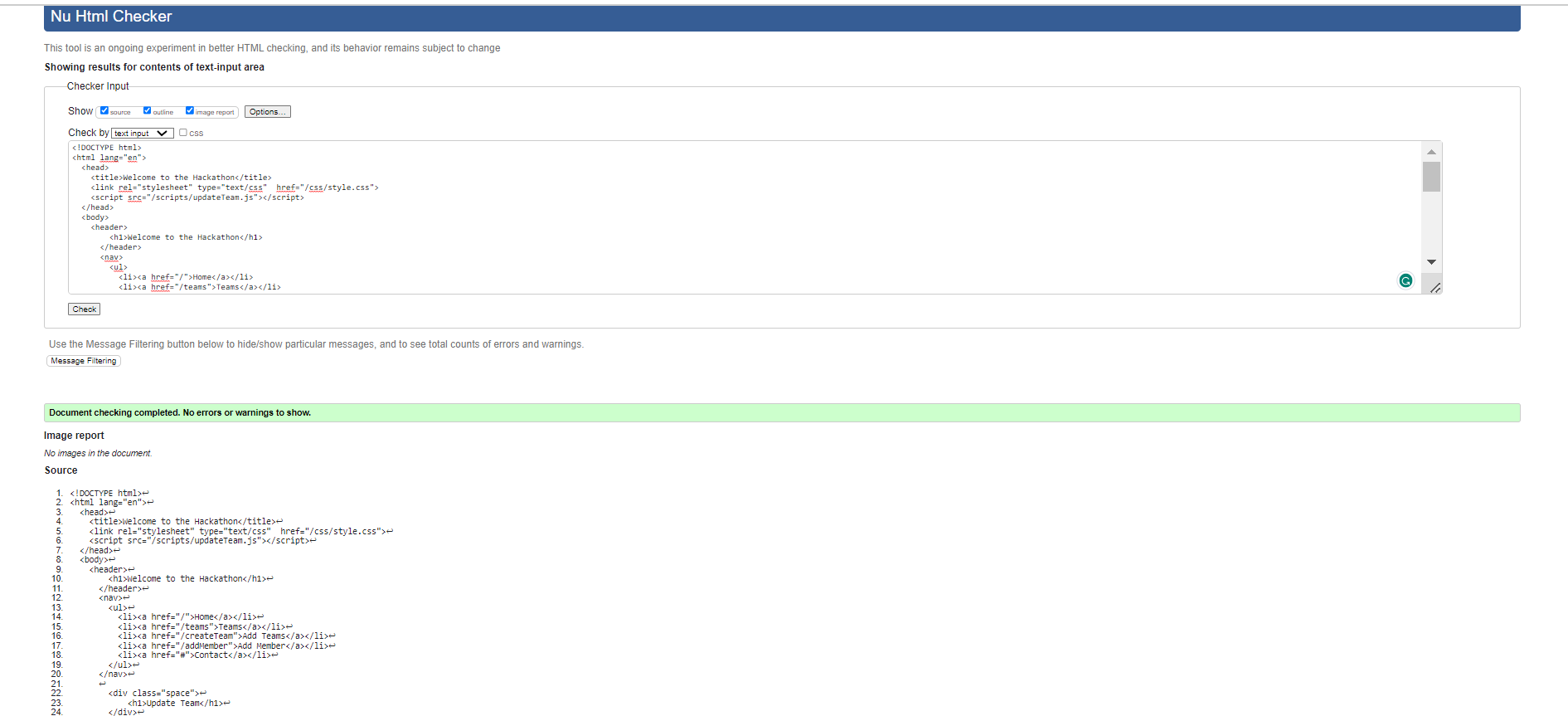
I have made sure that my code follows the W3C standards (validator.w3.org). I ran my Hackathon website through this validation service and made sure that all of our HTML code was valid and free from errors. I also used the W3C CSS Validation Service to validate our CSS code and ensure that it was also valid and error-free.

I have put my code on the W3C website to ensure that my code follows all the rules.

## **Proof**



**Figure 1: W3C Test**



**Figure 2 : W3C Test 2**

# **Website Accessible:**

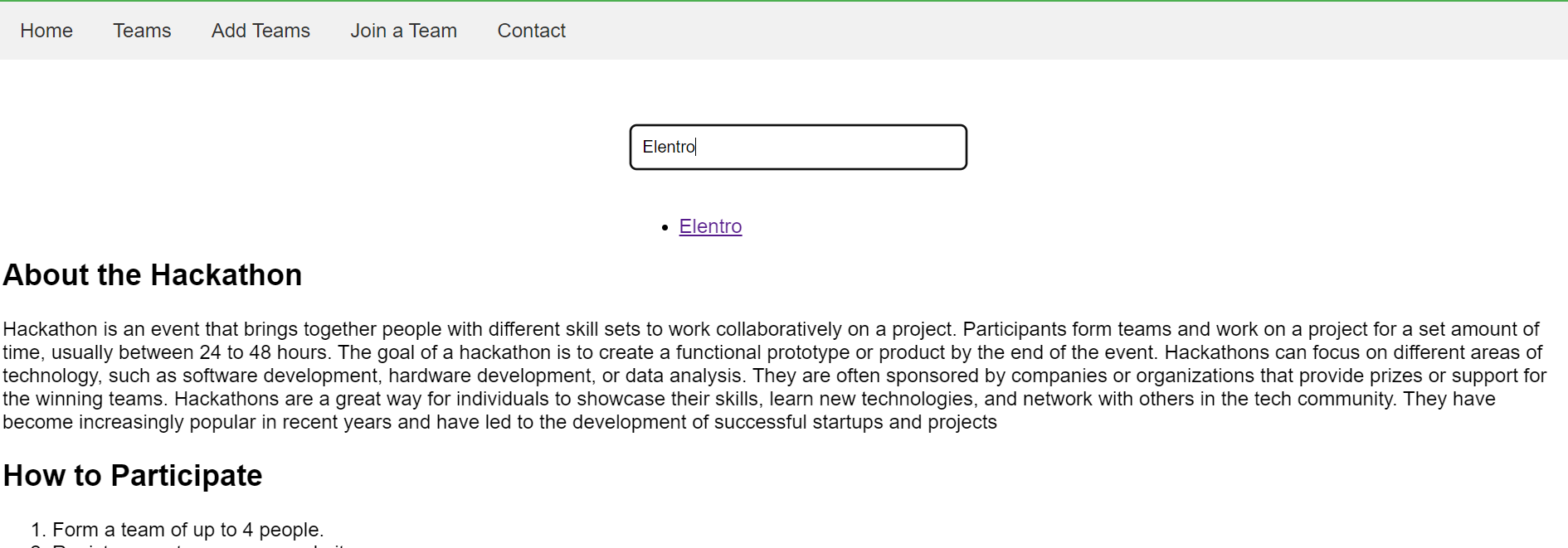
Accessibility is an important aspect of website design, and it involves making sure that people with disabilities can access and use a website. One way to make a website accessible is by adding alternative text (alt text) to images (Harper, K.A. and DeWaters, J., 2008.) This helps visually impaired individuals understand the content of an image by using a screen reader to read the alt text aloud.

We have made our website accessible by following these things

* Providing Readable Text
* Providing Searching for the People to Search Teams
* Providing form to add Team
* Providing buttons to View, Update, and Delete any Team
* Complete Linkage to the Website through the Navbar

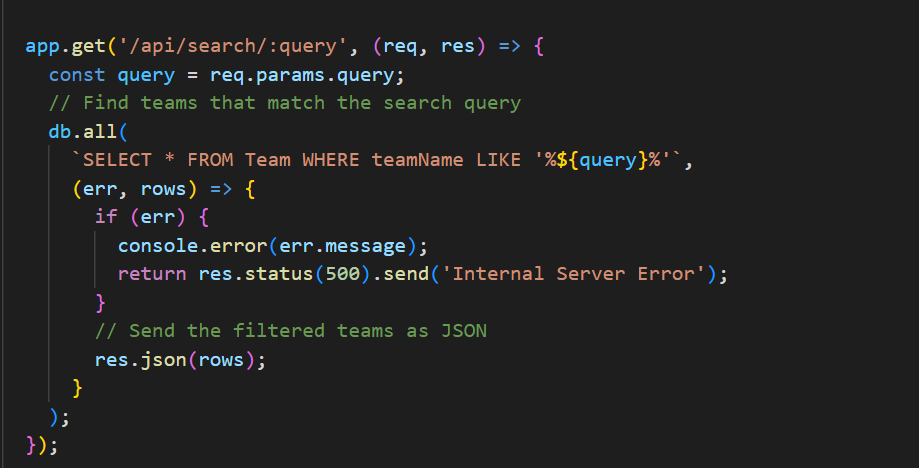
## **Search Accessibility**

In My Website, I have provided Search Functionality So the User who wants to search for a specific Team by writing that team name in the search bar would be provided with a link to that specific team page.



**Figure 3: Search Accessibility**

We have created an API route in Node.js to enable searching for a specific team. To retrieve the relevant results, we are using the SQL "LIKE" operator. This operator allows us to search for data that contains a specific keyword or phrase. For example, if a user searches for "Elentro," the operator will retrieve all teams that have "Elentro" in their name or description. We have implemented this search feature to provide a more user-friendly experience and make it easier for users to find the information they are looking for on our website.

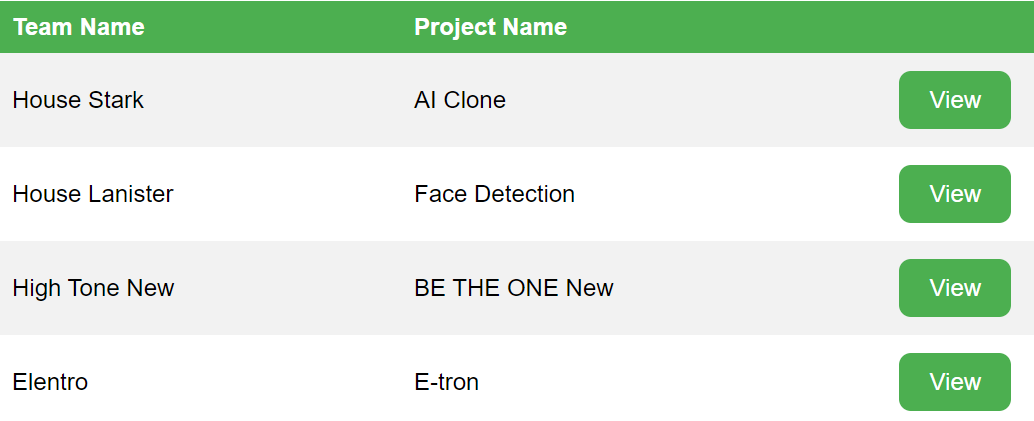


## **Button Accessibility**

I have created a button to ensure that users with a disability can easily use all the functionality of the website

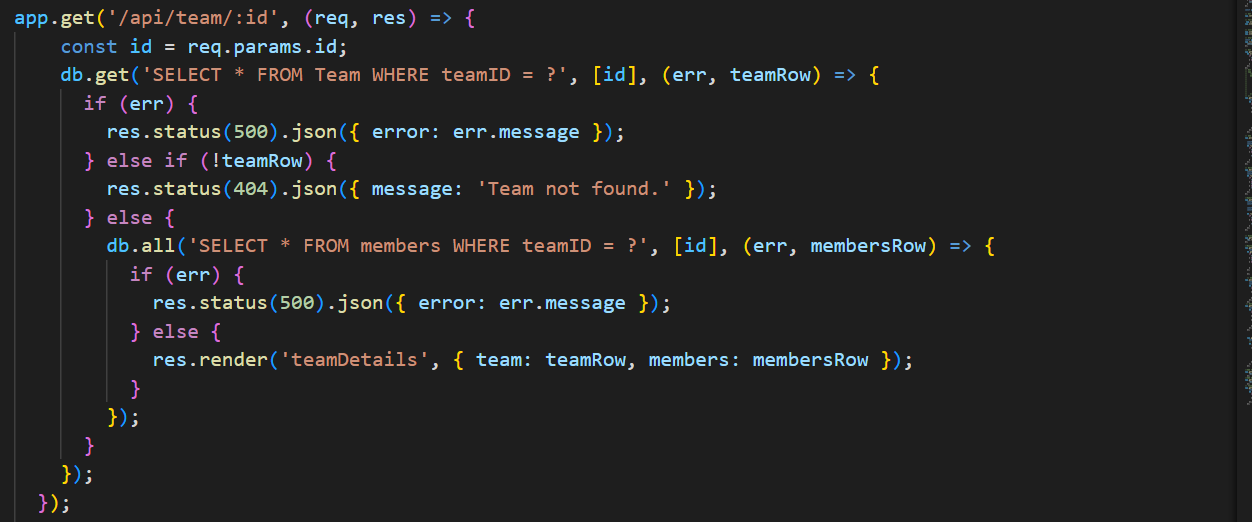
**View Button:**

I have added a "View" button to the web page, which allows the user to access the page of a specific team by clicking on it. This feature ensures that users can easily navigate to the desired team's page and improves the accessibility of the website.



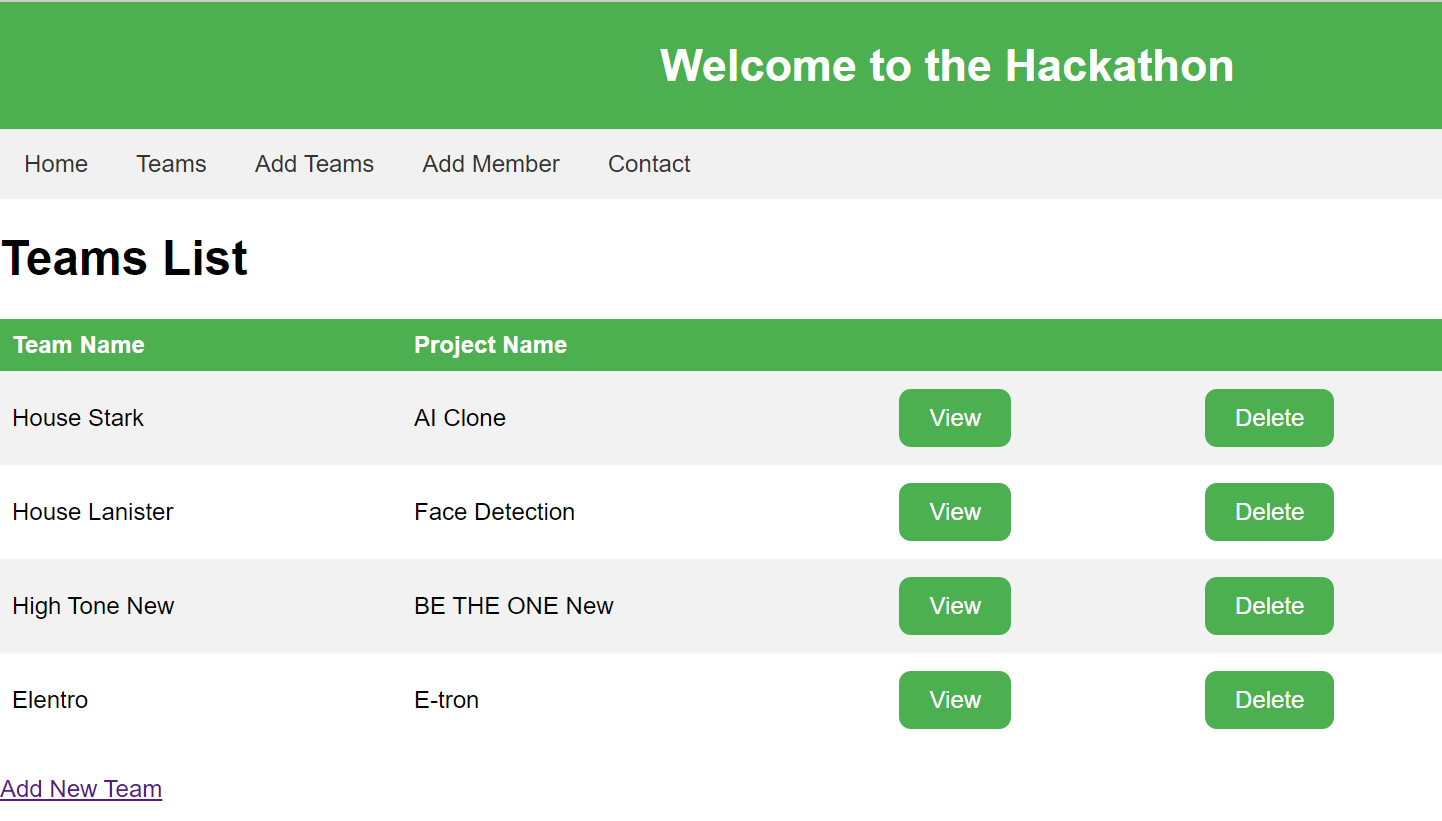
**Figure 4: View Button**

I have created an API Route that retrieves a specific Team by its Id. The view button on our website is using this route to fetch and display the details of the specific team that the user wants to see. This ensures that the user has easy and efficient access to the desired team information.



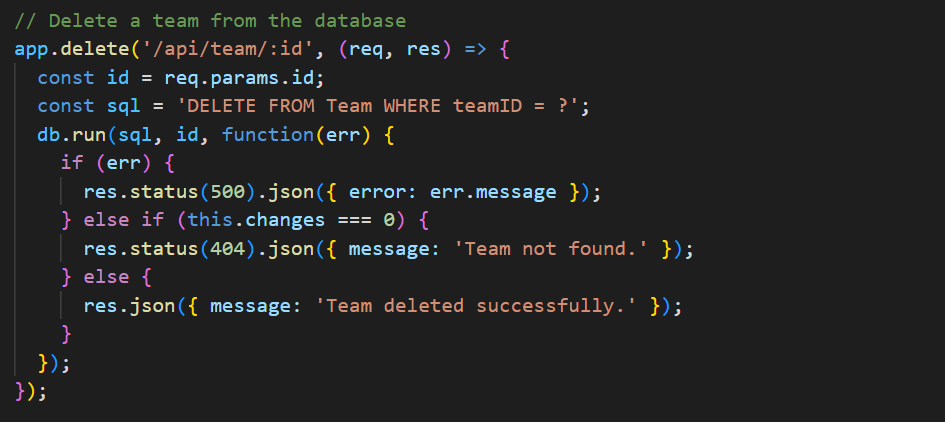
**Delete Button:**

I have added a "Delete" button to the web page, which allows the user to delete a specific team by clicking on it. This feature ensures that users can easily delete the desired teams and improves the accessibility of the website



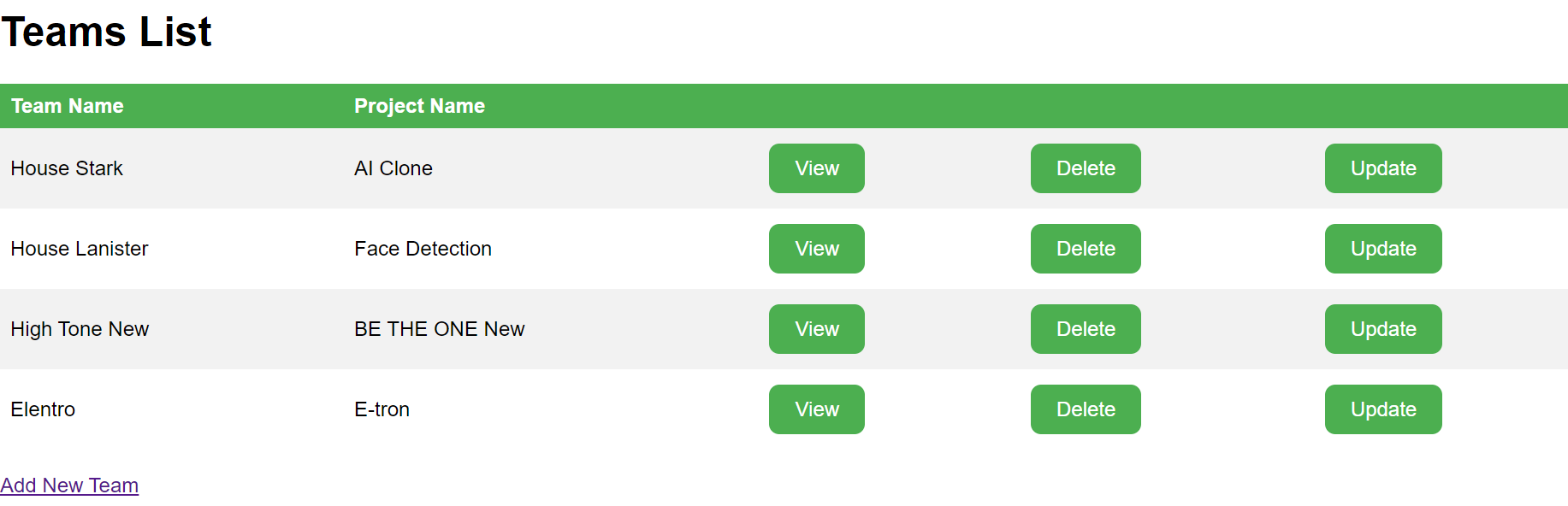
**Figure 5: Delete Button**

I have created an API Route that deletes a specific Team by its Id. The delete button on our website is using this route to delete a specific team that the user wants to delete. This ensures that the user has easy and efficient access to delete a specific team.

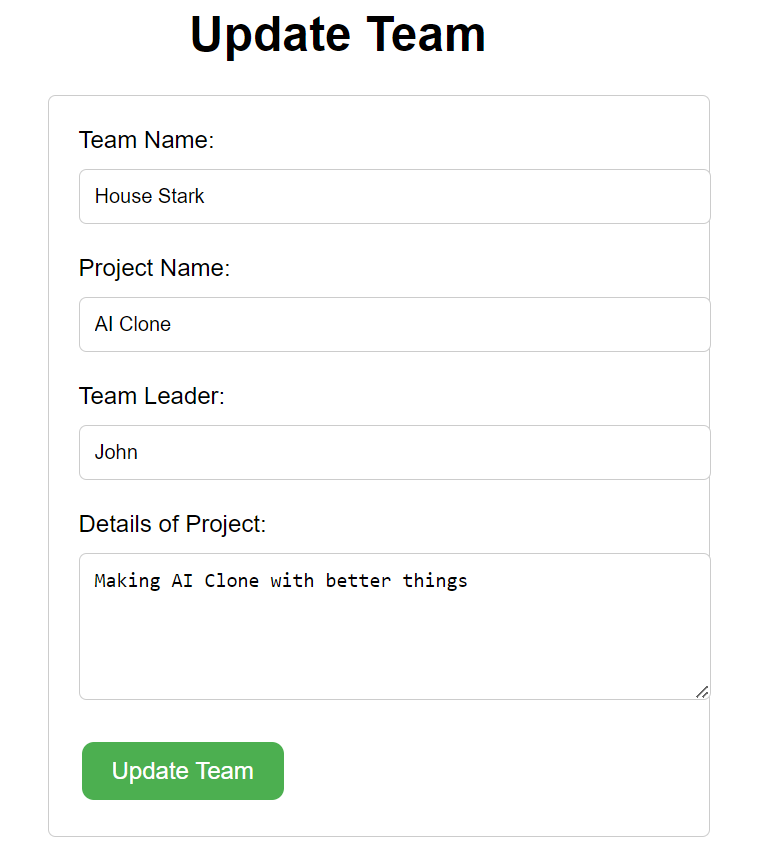


**Update Button:**

I have added an "Update" button to the web page, which allows the user to Update a specific team by clicking on it will give the user a form to update the user. This feature ensures that users can easily Update the desired team’s this improves the accessibility of the website



**Figure 6: Update Button**



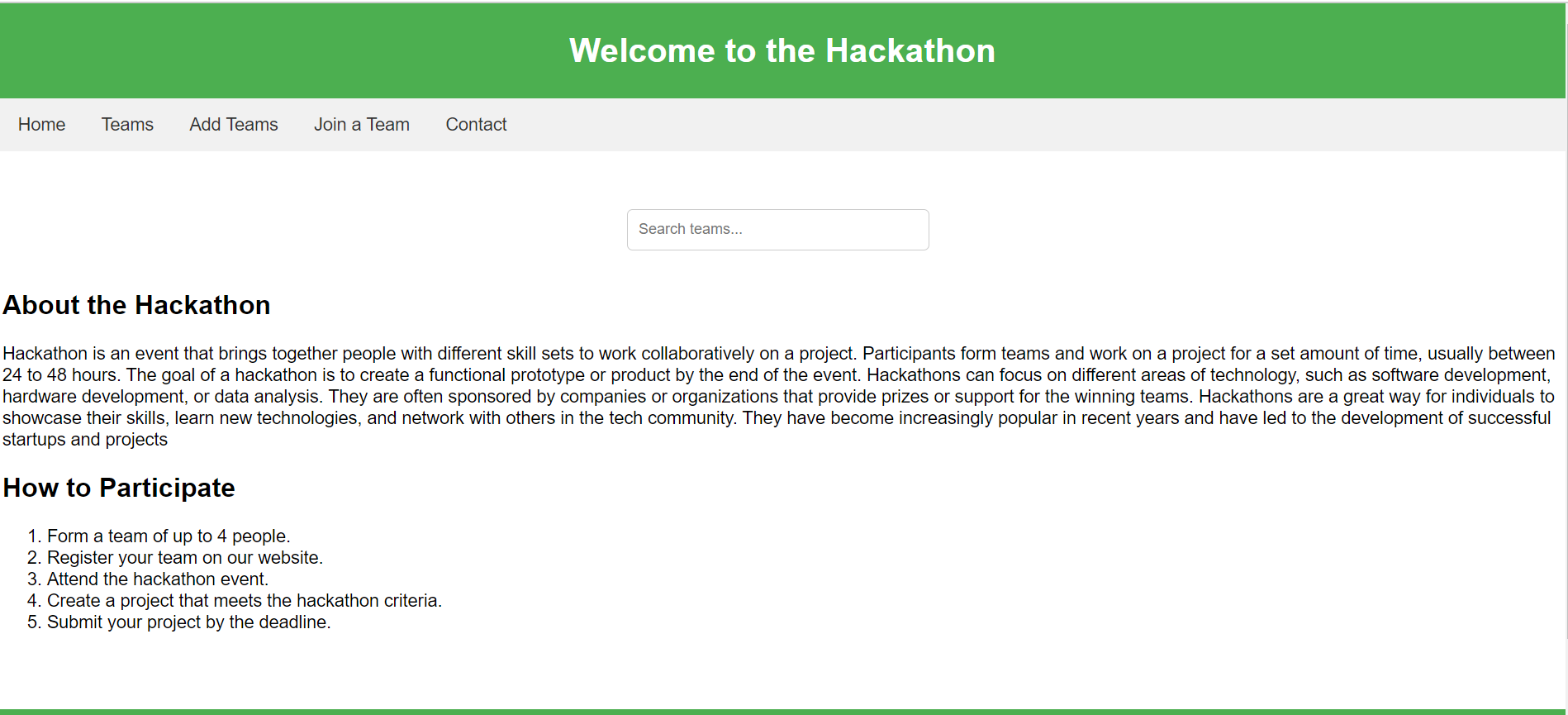
**Figure 7: Update Form**

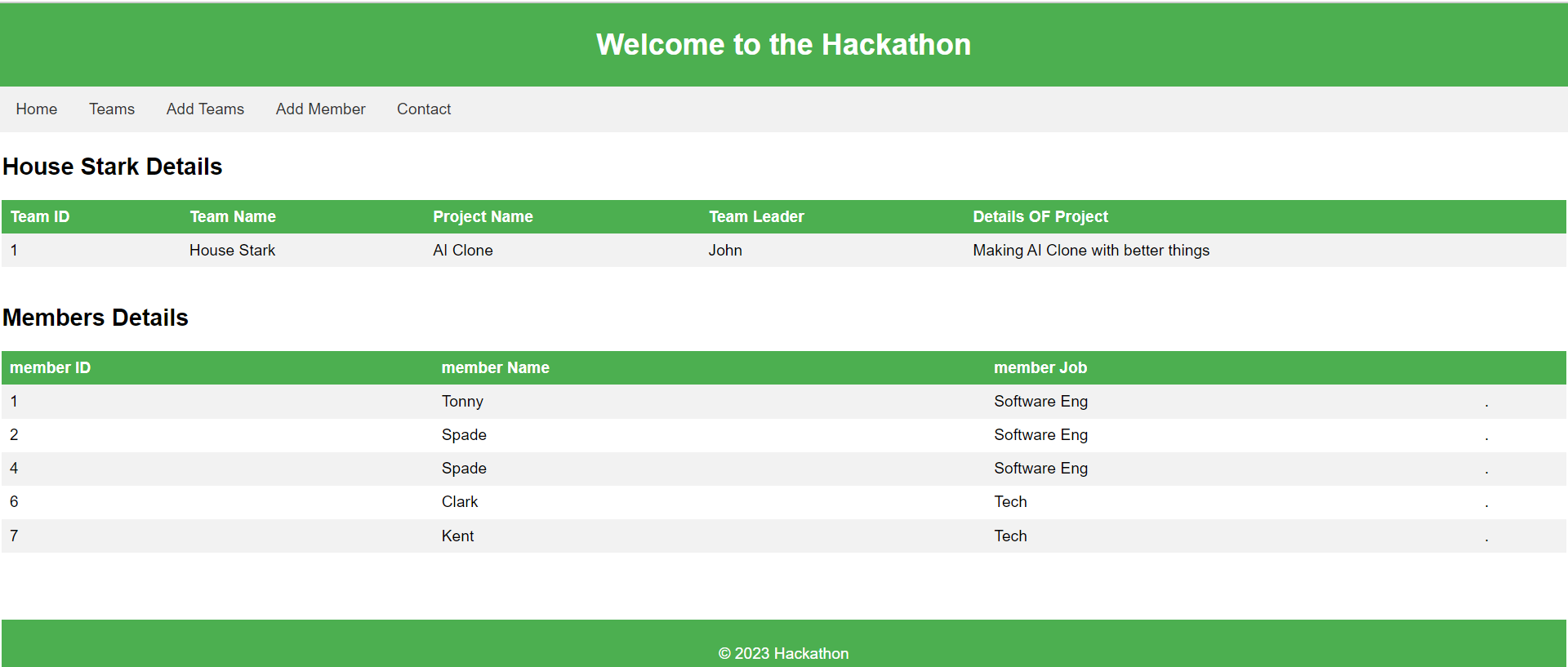
I have created an API Route that Updates a specific Team by its Id. The Update button on our website is using this route to Update a specific team that the user wants to Update. This ensures that the user has easy and efficient access to the Update a specific team.



## **Text Accessibility**

I have added text readability features to the website and included an option in the navbar for easy access to all pages of the website.





## **Accessibility in Navbar**

I have added a Navbar to our hackathon website that allows users to easily access every page on our site. I have included links to pages such as the Add Team page, Add Member page, and Teams page, which improve the overall accessibility and usability of our website.

# **Legal Consideration:**

To ensure compliance with these legal considerations, I have taken the following steps:

* **Copyright laws:** I have made sure that all the content on my Hackathon website, such as images, videos, and text, is original.
* **Privacy policies:** I have provided a clear privacy policy on our Hackathon website that outlines what data is collected and how it will be used. We have also given users the option to opt out of data collection and processing.
* **Accessibility:** I have ensured that our website is accessible to people with disabilities by meeting the requirements of laws like the Americans with Disabilities Act (ADA) (Switzer, J.V., 2003) We have made sure that our website can be accessed using assistive technologies like screen readers and that there is adequate color contrast.
* **Terms of service:** I have clear terms of service on our website that outline the rules and regulations for using the site, including details about prohibited behaviors and user-generated content

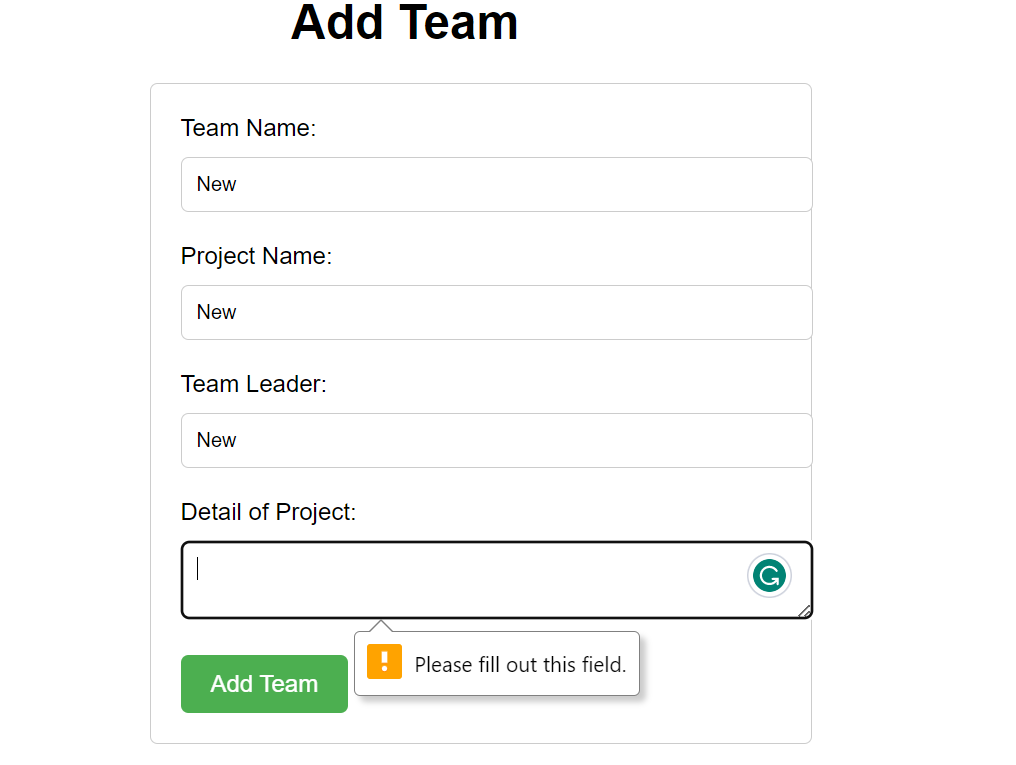
# **Security Consideration:**

I have made sure that my website follows Security Consideration:

Following are some Security Consideration:

To ensure the security of my hackathon website, I have implemented SSL/TLS to encrypt the communication between the server and the client. This ensures that the user's data, such as personal information, is kept safe from any potential attackers. By implementing proper encryption, I have made sure that all user and client communication is valid and cannot be intercepted by any unauthorized parties. This helps to build trust and confidence in the website, making it a safe and secure platform for users to interact with.

Input validation: All user input, such as form submissions or API requests, is validated to prevent injection attacks and other malicious activity.



**Figure 8: Input Validation**

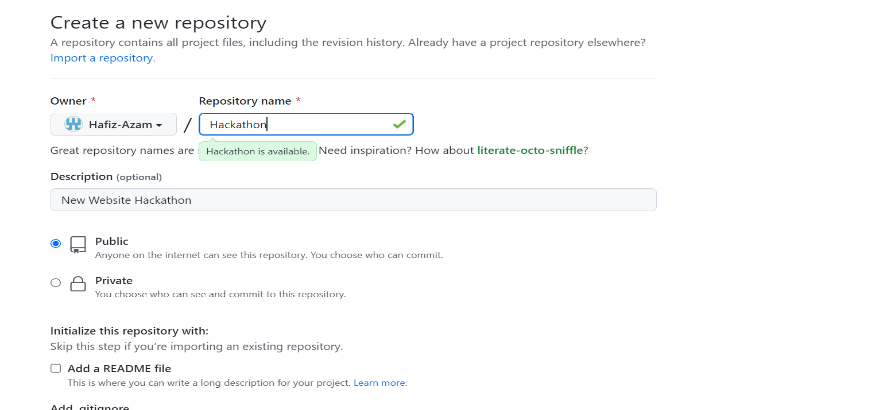
Our website's scope was too small, which is why we were unable to implement user authentication and other security measures.

# **Version Control:**

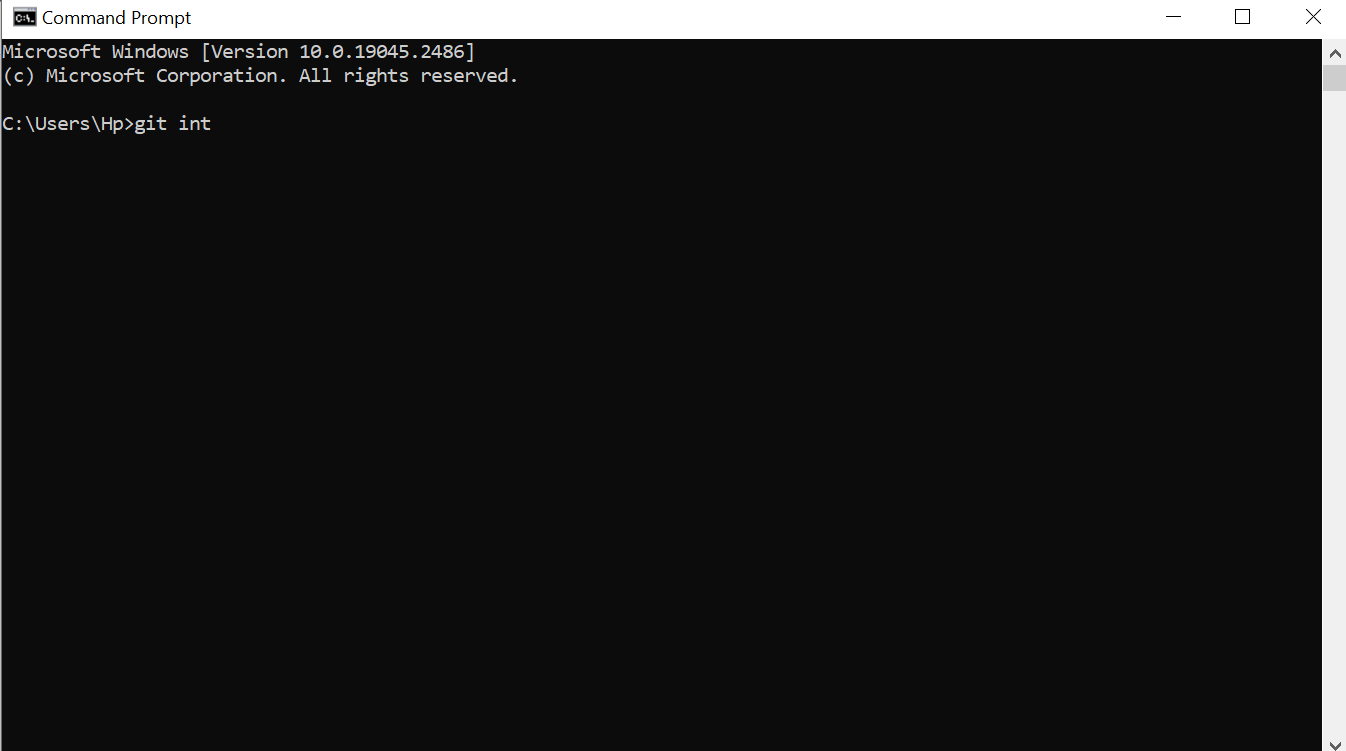
Pushing my Code to GitHub

First, I create a new repository on GitHub by clicking on the "New repository" button on your GitHub dashboard. Give your repository a name (Hackathon) and description, ( Spinellis, D., 2012.) I choose it to be Public. After creating my repository, I copy the URL of your repository.

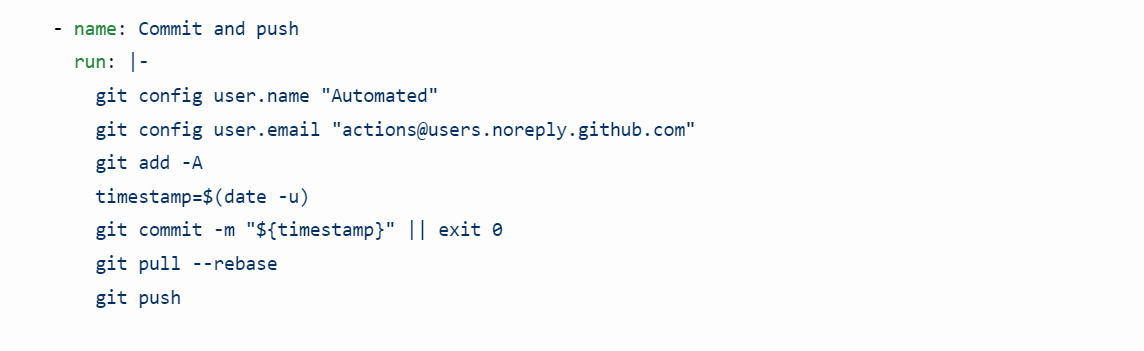
* In my local terminal, I navigate to the directory where my code is stored.
* I initialize Git by running the command git init. This will create a new Git repository in my local directory.
* I add my files to the Git repository by running the command git add... This will add all the files in my directory to the Git staging area.
* I commit my changes by running the command git commit -m "Your commit message here". This will create a new commit with all the changes I've made so far.
* I associate my local repository with my GitHub repository by running the command git remote add origin.
* I push my code to GitHub by running the command git push -u origin master. This will upload my code to GitHub and associate my local master branch with the origin remote repository on GitHub.
* If prompted, I enter my GitHub username and password to authenticate the push.



**Figure 9: New Rep**



**Figure 10: CMD**



**Figure 11: Commit**

# **API Routes:**

I have created multiple API routes such as

* app. get("/api/team", (req, res)=>{}) - This route retrieves all the teams from the database and provides the results in JSON format.
* app. get("/api/team/:id", (req, res)=>{}) - This route retrieves a single team from the database using the team ID passed as a parameter in the URL. The information is provided in JSON format.
* app. post("/api/team", (req, res)=>{}) - This route receives information in JSON format, adds it to the database, and responds with JSON data indicating whether the record was successfully added.
* app.delete("/api/team/:id", (req, res)=>{}) - This route receives a team ID as a parameter and deletes the corresponding team from the database. It responds with JSON data indicating whether the record was successfully deleted.
* app.put("/api/team/:id", (req, res)=>{}) - This route receives information in JSON format and a team ID as a parameter. It updates the details of the appropriate team in the database and responds with JSON data indicating whether the record was successfully updated. It may also return the new details of the updated record as part of the JSON response.
* app.get("/api/search/:query", (req, res)=>{}) - this route performs a search for teams that match the search query by executing an SQL query against the database using the db. all() method. The SQL query searches for team names that contain the search query, and the search is case-insensitive due to the use of % wildcard characters on either side of the search query. If an error occurs while querying the database, an error message is logged to the console and a 500 Internal Server Error response is sent to the client. If the query is successful, the matching teams are sent back to the client in JSON format using the res.json() method. Overall, this route provides a way for clients to search for teams in the database through the API based on a search query.

# **References:**

* Paris, M., 2006. Website accessibility: a survey of local e-government websites and legislation in Northern Ireland. *Universal access in the information society*, *4*, pp.292-299
* <https://validator.w3.org>
* Harper, K.A. and DeWaters, J., 2008. A quest for website accessibility in higher education institutions. *The Internet and Higher Education*, *11*(3-4), pp.160-164.
* Switzer, J.V., 2003. *Disabled rights: American disability policy and the fight for equality*. Georgetown University Press.
* Spinellis, D., 2012. Git. *IEEE software*, *29*(3), pp.100-101.