

Objective:

To test that the web application correctly defends against Cross-Site Request Forgery (CSRF) attacks.

Method:

1. Launched the frontend (<http://localhost:3000>) and logged in normally.
2. Opened the browser developer console.
3. Manually deleted the CSRF token from the app's memory using:

```
delete window.axios.defaults.headers.common["X-CSRF-Token"]
```

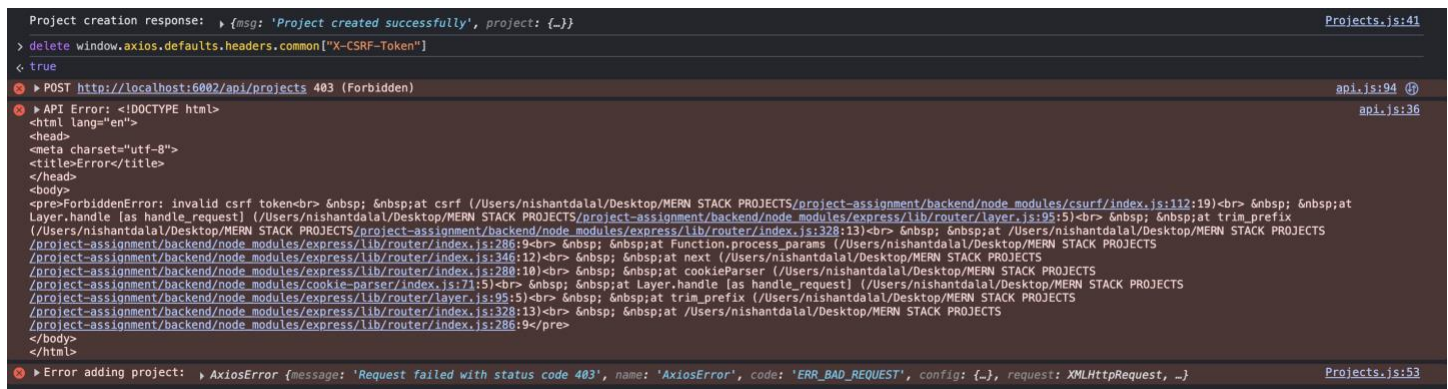
4. Attempted to perform a protected action (e.g., creating a project).
 5. Observed the server response:
- 403 ForbiddenError: invalid csrf token

Result:

The server rejected the request due to the missing CSRF token. This confirms that CSRF protection is successfully enforced, and the web app is secure against CSRF attacks.

Tools Used:

- Browser Developer Tools (Console tab)
- React frontend
- Axios (for HTTP requests)

A screenshot of a web browser's developer console. The top part shows a successful POST request to 'http://localhost:6002/api/projects' with a response of {msg: 'Project created successfully', project: {...}}. Below this, the 'delete window.axios.defaults.headers.common["X-CSRF-Token"]' code is executed. The next entry is a 403 Forbidden error from the same endpoint. The console shows the raw HTML response, which is an error page with a title 'Error' and a body containing the message 'ForbiddenError: invalid csrf token'. The error is also shown as an AxiosError object at the bottom.

```
Project creation response: {msg: 'Project created successfully', project: {...}} Projects.js:41
> delete window.axios.defaults.headers.common["X-CSRF-Token"]
< true
> POST http://localhost:6002/api/projects 403 (Forbidden) api.js:94
> API Error: <!DOCTYPE html> api.js:36
<html lang="en">
<head>
<meta charset="utf-8">
<title>Error</title>
</head>
<body>
<pre>ForbiddenError: invalid csrf token<br> <script> </script>
Layer.handle [as handle_request] (/Users/nishantdal/Desktop/MERN STACK PROJECTS/project-assignment/backend/node_modules/express/lib/router/layer.js:95:15)
/Users/nishantdal/Desktop/MERN STACK PROJECTS/project-assignment/backend/node_modules/express/lib/router/index.js:328:13)
/project-assignment/backend/node_modules/express/lib/router/index.js:286:9)
/project-assignment/backend/node_modules/express/lib/router/index.js:346:12)
/project-assignment/backend/node_modules/express/lib/router/index.js:280:10)
/project-assignment/backend/node_modules/cookie-parser/index.js:71:5)
/project-assignment/backend/node_modules/cookie-parser/index.js:95:5)
/project-assignment/backend/node_modules/express/lib/router/index.js:328:13)
/project-assignment/backend/node_modules/express/lib/router/index.js:286:9)
</body>
</html>
> Error adding project: AxiosError {message: 'Request failed with status code 403', name: 'AxiosError', code: 'ERR_BAD_REQUEST', config: {...}, request: XMLHttpRequest, ...} Projects.js:53
```

CSRF Protection Testing – Success Case

Method:

1. Refreshed the app to allow it to fetch the CSRF token from the backend.

2. Performed a protected action (e.g., creating a new project) without modifying the CSRF token.

3. Observed that the request succeeded with a 200 OK status.

Result:

When the CSRF token is present, the request is accepted and processed by the server. This confirms that the application handles CSRF tokens correctly in a valid session.

Project creation response: `{msg: 'Project created successfully', project: {...}}`

Projects.js:41

Project Manager

Register CandidateCandidatesPro

Candidate added successfully!

Candidates

NameEmailSkills (comma-separatedAdd Candidate

Name: John DoeEmail: johndoe@example.comSkills: JavaScript, React, Node.js

Name: NishantEmail: dalalnishant0207@gmail.comSkills: JavaScript, React, Node.js

Name: Test UserEmail: user@gmail.comSkills: Python, Java, Kotlin

Name: User 2Email: testmail@gmail.comSkills: React, Express

Name: NishantEmail: nishant@gmail.comSkills: Reactjs, Node.js, Express

Name: Gourav RathilEmail: gourav@gmail.comSkills: Nodejs, Reactjs, Backend, Full Stack

Name: Test User 3Email: testuser@gmail.comSkills: Python, Flask

Name: Test User 4Email: testuser4@gmail.comSkills: None

CSRF Protection – Invalid Token Test

Method:

1. Overwrote the CSRF token in the browser using:

```
window.axios.defaults.headers.common["X-CSRF-Token"] = "invalid-token-123"
```

2. Tried performing a protected POST action.

3. Server responded with 403 Forbidden.

Result:

The server detected the tampered CSRF token and blocked the request. This confirms protection against forged or expired tokens.

```
> window.axios.defaults.headers.common["X-CSRF-Token"] = "invalid-token-123"
< 'invalid-token-123'
➤ POST http://localhost:6002/api/projects 403 (Forbidden) api.js:94
➤ API Error: <!DOCTYPE html> api.js:36
<html lang="en">
<head>
<meta charset="utf-8">
<title>Error</title>
</head>
<body>
<pre>ForbiddenError: invalid csrf token<br> &nbsp;&nbsp;&nbsp;at csrf (/Users/nishantdal/Desktop/MERN STACK PROJECTS/project-assignment/backend/node_modules/csrf/index.js:112:19)<br> &nbsp;&nbsp;&nbsp;at Layer.handle [as handle_request] (/Users/nishantdal/Desktop/MERN STACK PROJECTS/project-assignment/backend/node_modules/express/lib/router/layer.js:95:5)<br> &nbsp;&nbsp;&nbsp;at trim_prefix (/Users/nishantdal/Desktop/MERN STACK PROJECTS/project-assignment/backend/node_modules/express/lib/router/index.js:328:13)<br> &nbsp;&nbsp;&nbsp;at /Users/nishantdal/Desktop/MERN STACK PROJECTS/project-assignment/backend/node_modules/express/lib/router/index.js:336:12)<br> &nbsp;&nbsp;&nbsp;at next (/Users/nishantdal/Desktop/MERN STACK PROJECTS/project-assignment/backend/node_modules/express/lib/router/index.js:280:10)<br> &nbsp;&nbsp;&nbsp;at cookieParser (/Users/nishantdal/Desktop/MERN STACK PROJECTS/project-assignment/backend/node_modules/cookie-parser/index.js:71:5)<br> &nbsp;&nbsp;&nbsp;at Layer.handle [as handle_request] (/Users/nishantdal/Desktop/MERN STACK PROJECTS/project-assignment/backend/node_modules/express/lib/router/layer.js:95:5)<br> &nbsp;&nbsp;&nbsp;at trim_prefix (/Users/nishantdal/Desktop/MERN STACK PROJECTS/project-assignment/backend/node_modules/express/lib/router/index.js:328:13)<br> &nbsp;&nbsp;&nbsp;at /Users/nishantdal/Desktop/MERN STACK PROJECTS/project-assignment/backend/node_modules/express/lib/router/index.js:286:9</pre>
</body>
</html>
➤ Error adding project: > AxiosError {message: 'Request failed with status code 403', name: 'AxiosError', code: 'ERR_BAD_REQUEST', config: {_, request: XMLHttpRequest, ...} Projects.js:53
```

Objective:

To ensure that the frontend provides **clear feedback** to users when a CSRF error occurs, such as during an expired session or invalid/missing token.

Test Performed:

1. Logged into the application normally.
2. Opened browser console and **manually deleted the CSRF token** using:

```
delete axios.defaults.headers.common["X-CSRF-Token"]
```

3. Attempted to create a project (a protected action).
4. The backend returned a **403 Forbidden** error due to the invalid/missing CSRF token.
5. The frontend **displayed an alert** to the user with the message:

```
Session expired or invalid request. Please refresh the page.
```

Result:

The application correctly detects CSRF errors and **alerts the user** with a friendly message, instead of silently failing or showing a raw error.

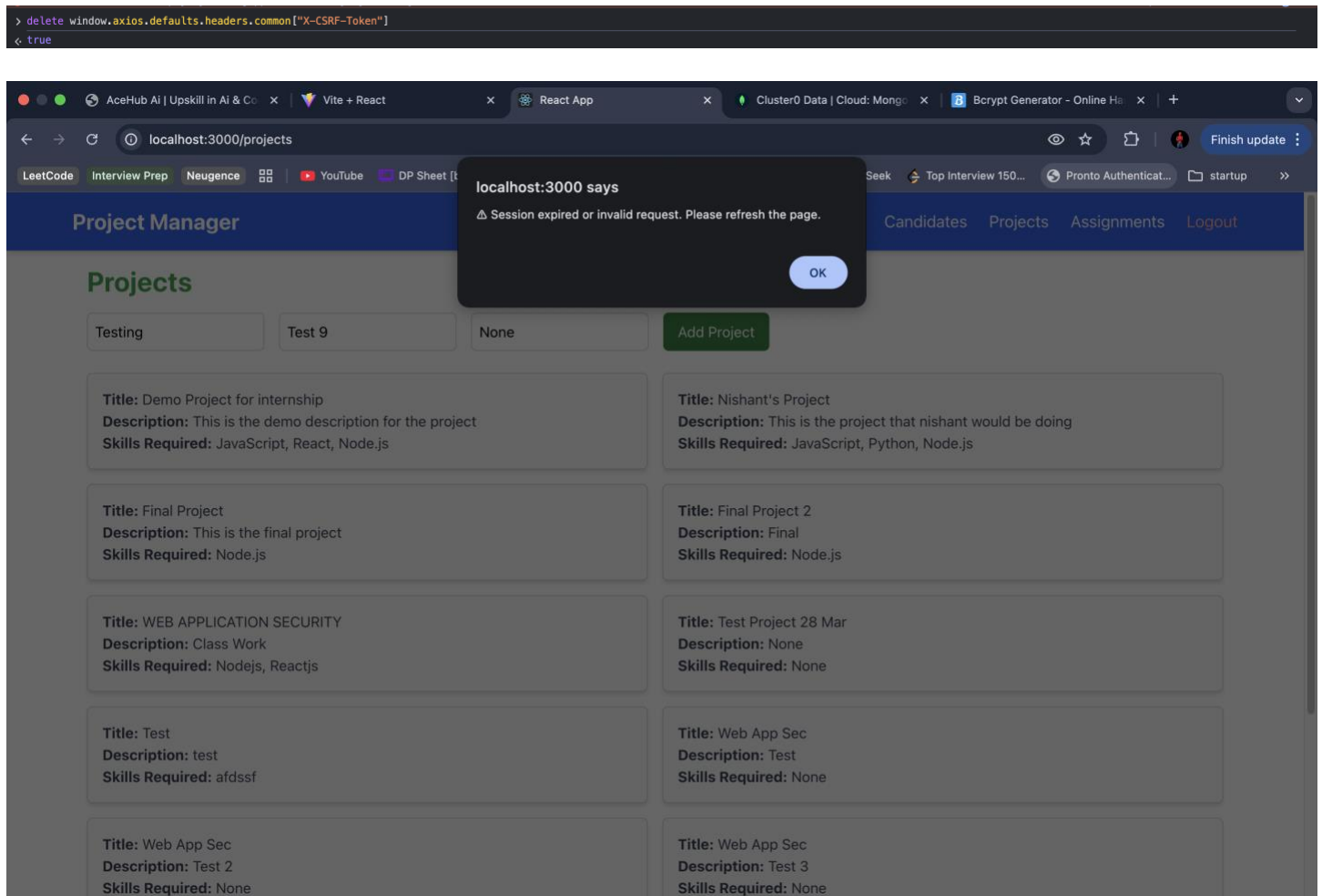
This improves:

- Security awareness for users
- Debugging clarity
- Overall user experience

Tools Used:

- Chrome DevTools

- Axios error interceptor in React frontend
- alert() for real-time feedback



Conclusion

Through a series of controlled browser-based tests, the application was found to be well-protected against CSRF (Cross-Site Request Forgery) attacks. Attempts to send requests without or with invalid CSRF tokens were correctly rejected by the server. Additionally, the frontend provided clear user feedback when such errors occurred. These tests confirm the implementation of strong CSRF defense mechanisms using cookies, Axios, and Express middleware.