**Report: Troubleshooting CORS Issues in GitHub Codespaces Environment**

**Author:** Khadija Mahmoud  
**Project:** Coworking Space Web Application  
**Environment:** GitHub Codespaces  
**Frontend:** React  
**Backend:** Flask

**1. Problem Description**

While developing the Coworking Space web application in GitHub Codespaces, I encountered persistent CORS (Cross-Origin Resource Sharing) errors when the frontend attempted to send POST requests to the Flask backend, specifically to the /api/register route.

**2. Observed Errors**

* Access to fetch at ... has been blocked by CORS policy
* No 'Access-Control-Allow-Origin' header is present on the requested resource
* net::ERR\_FAILED during preflight requests
* 401 status on preflight requests
* WebSocket connection failure: wss://...3000.app.github.dev/ws

**3. What I Tried**

**a. CORS Configuration in Flask**

A screenshot of a computer code

AI-generated content may be incorrect.I added a CORS configuration with the correct origin URLs:

Figure 1: CORS Configuration

**b. Included OPTIONS Method Handling**

In the /register route:

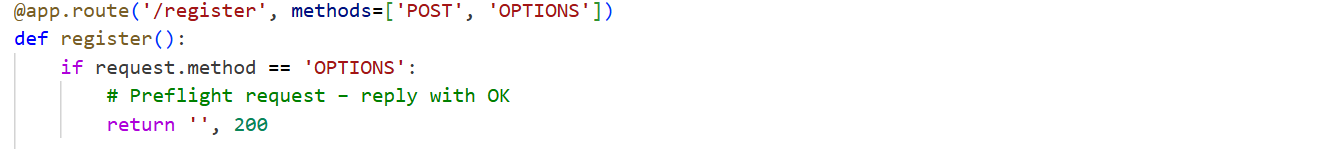


Figure 2: Method Handling

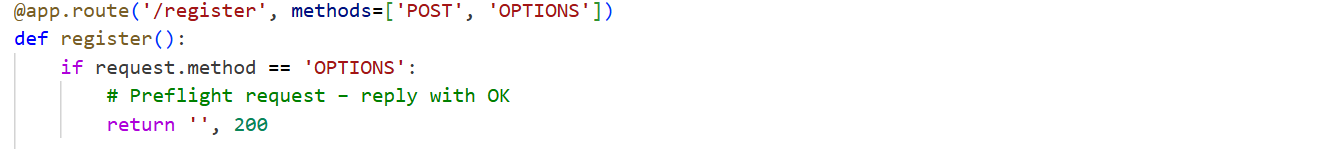
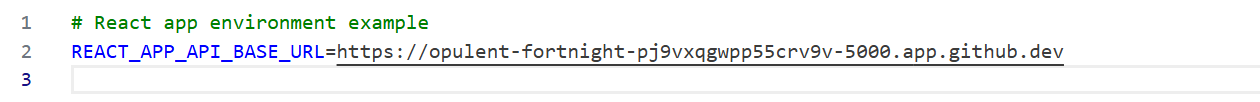
**c. Enabled CORS Headers Manually**

A screenshot of a computer code

AI-generated content may be incorrect.I added an @after\_request handler to ensure headers are set:

Figure 3: After\_Request Handler

d. **Frontend Configuration**



And in my request:

A computer code on a white background

AI-generated content may be incorrect.

Figure 4: Credentials “Include”

**4. Why It’s Not Working**

Despite the correct CORS configuration and frontend code, the following may be causing the issue:

* **GitHub Codespaces Tunneling**: The URL rewrites (\*.3000.app.github.dev and \*.5000.app.github.dev) introduce unexpected redirections that fail CORS preflight checks.
* **Missing CORS Headers for Preflight (OPTIONS)**: While OPTIONS is handled in /register, Codespaces may send unexpected headers or the response may lack the proper Access-Control-Allow-Origin.
* **WebSocket Limitations**: Codespaces restrict WebSocket traffic, especially cross-port (5000 ↔ 3000) interactions.
* **Network Tunnel Interception**: GitHub may rewrite or block headers as part of its tunneling layer for Codespaces.

**5. Conclusion and Next Steps**

Throughout the project timeline, I consistently completed my assigned tasks early and was prepared to integrate them. However, I was unable to proceed due to dependencies on parts I was waiting to receive from teammates. I have consistently waited for my teammates to complete their parts in order to integrate the full system. Unfortunately, I only received the necessary components from them on the morning of Sunday, 13/07/2025, which significantly limited the time available for testing, debugging, and deploying the application on GitHub Codespaces.

I believe that if I had received the complete project earlier, I would have had sufficient time to properly configure and deploy the backend and frontend without the persistent CORS and networking errors. The delay left me with little room to resolve the complex environment-specific issues introduced by Codespaces. In the future, I will ensure earlier communication and internal deadlines with collaborators to avoid time crunches close to major deliverables. Despite the setback, I have documented the issues thoroughly and am prepared to deploy in a more stable environment (e.g., Render or Railway) once integration is fully functional.

Next Steps:

* Move development to a local environment using tools like ngrok for cross-origin testing, bypassing Codespaces networking restrictions.
* Explore deploying the app on Render, Railway, or Heroku to create a more stable environment.
* Communicate with teammates earlier and set stricter internal deadlines to allow enough time for testing and integration.

The GitHub repository containing the Codespace project can be accessed at: <https://github.com/khadija-mahmoud-17/coworking-space-app>