Exercise 11

Wish Wall

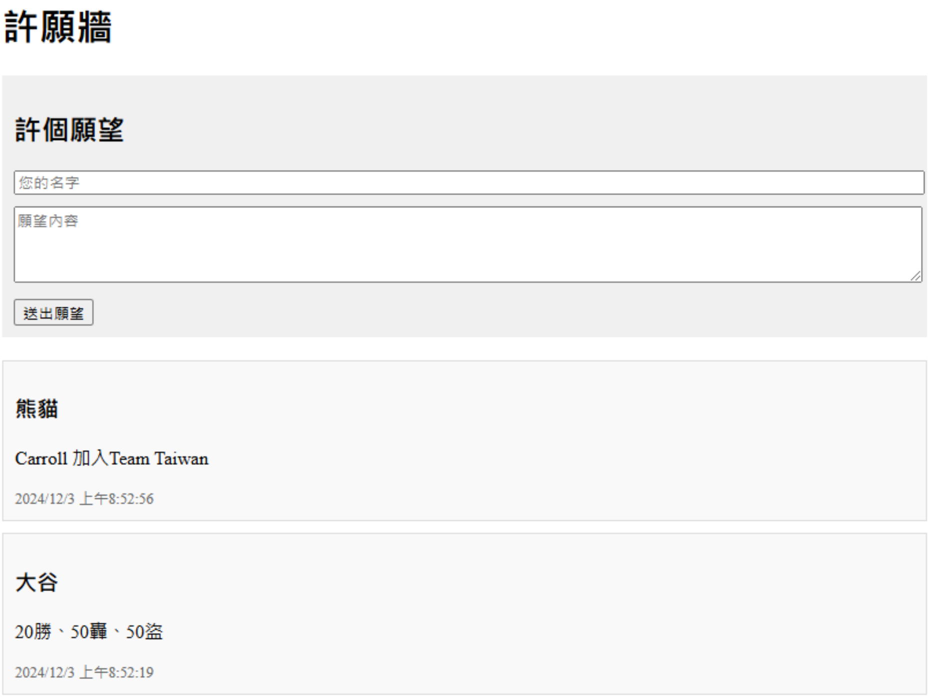
Please create a “wish wall (許願牆)” web application where users can share their wishes. This exercise helps you understand the basics of client-server communication.

Backend:

- A partial Node.js code is provided.
- Please Implement GET /api/wishes:
  - Return all wishes
  - Status code: 200
- Please Implement POST /api/wishes:
  - Accept: { name, message }
  - Add timestamp
  - Status code: 201 for success, 400 for errors

Frontend:

An HTML file (index.html) is provided. Please use this provided frontend code to test your API



許願牆

熊貓

Carroll 加入Team Taiwan

2024/12/3 上午8:52:56

大谷

20勝、50轟、50盜

2024/12/3 上午8:52:19

Testing Steps:

- Save revised backend code as wishes.js
- Run backend: node wishes.js
- Open index.html in browser
- Test both posting and viewing wishes

Requirements:

- Backend successfully handles GET requests
- Backend successfully handles POST requests
- Frontend can display wishes
- Frontend can submit new wishes
- Timestamps are displayed correctly

Submission:

Please submit “wishes.js” only.

Hints:

- Suggest testing your backend first using Postman.
- Please check browser console (F12) if any error occurs.

Note:

- Why do we need CORS? (This feature is provided)

  CORS (Cross-Origin Resource Sharing) is a security feature implemented by web browsers. It prevents web pages from making requests to a different domain (網域) than the one that served the web page. In our assignment:

  - Frontend: Running from a file (file://)
  - Backend: Running on http://localhost:3000
  - Without CORS headers, the browser would block these cross-origin requests
- Why do we need OPTIONS handling? (This feature is provided)

  When frontend makes certain types of HTTP requests (like POST with JSON), browsers automatically send an OPTIONS request first. This "preflight" request checks if the server allows: