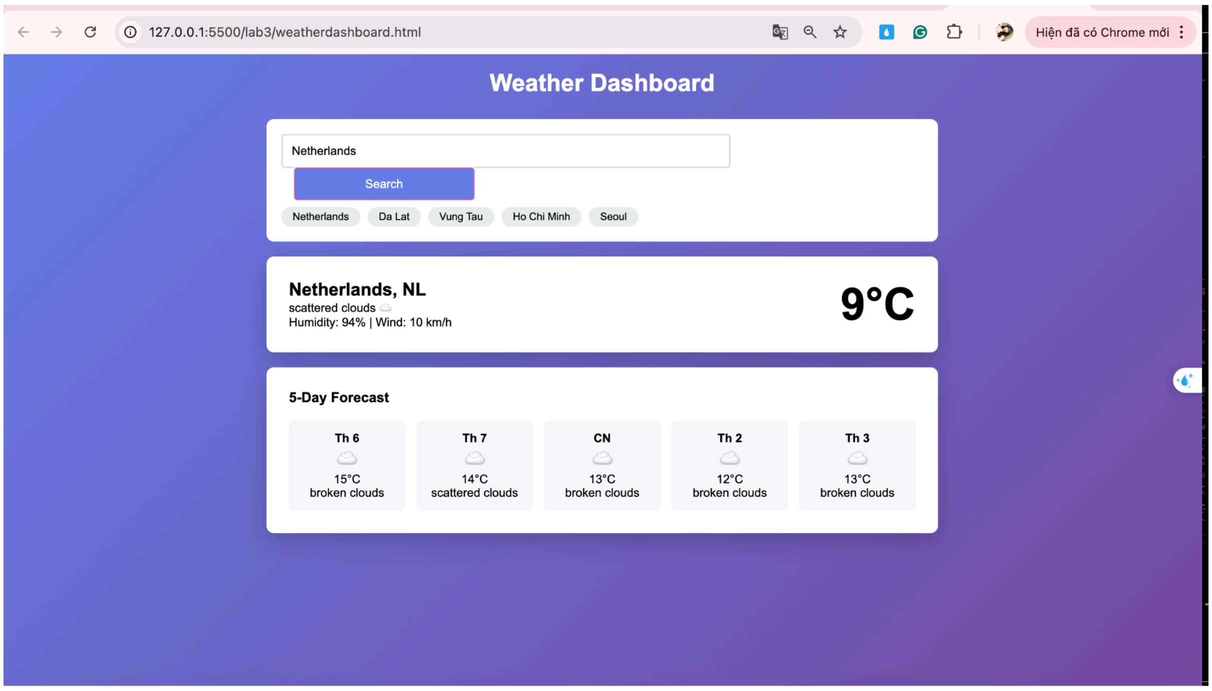
Name: Nguyen Ha Khanh Vy
Student ID: ITITDK21075
Course: Web Application Development Lab
Instructor: Msc. Nguyen Trung Nghia
Email: ntnghia@hcmiu.edu.vn

# Lab 3: Homework Exercise

Date: Nov 7th, 2025

## 2.1 Weather Dashboard

**Output**:



Weather Dashboard

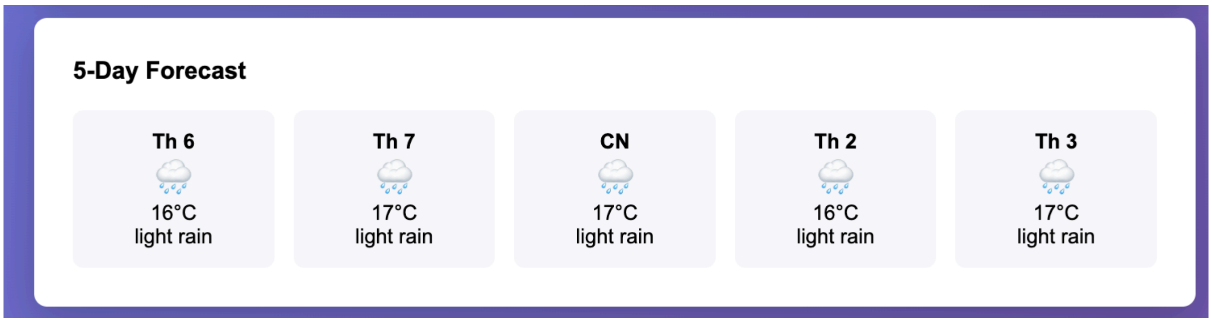
Netherlands

Netherlands, NL
scattered clouds
Humidity: 94% | Wind: 10 km/h
9°C

5-Day Forecast

Th 6 15°C broken clouds
Th 7 14°C scattered clouds
CN 13°C broken clouds
Th 2 12°C broken clouds
Th 3 13°C broken clouds

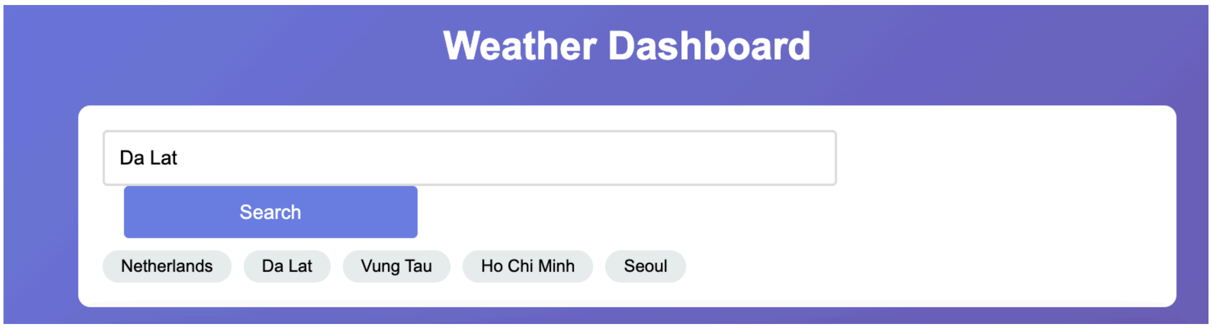
Search finds cities



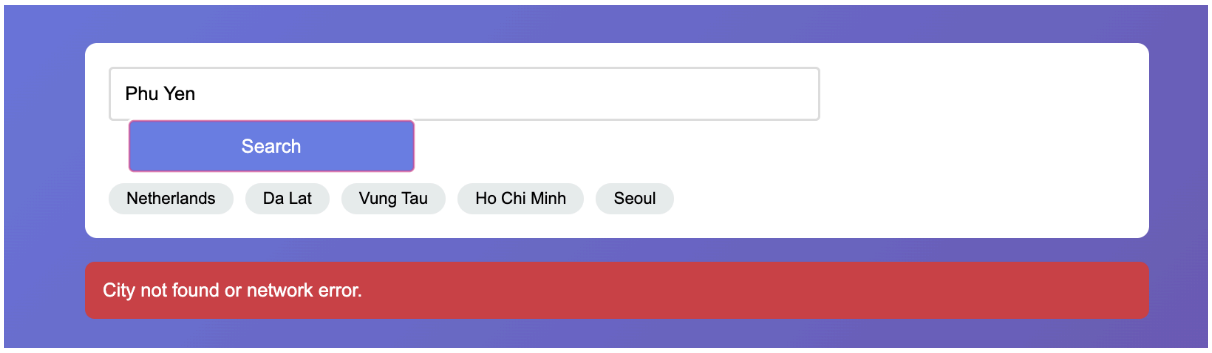
5-Day Forecast

Th 6 16°C light rain
Th 7 17°C light rain
CN 17°C light rain
Th 2 16°C light rain
Th 3 17°C light rain

Forecast shows 5 days

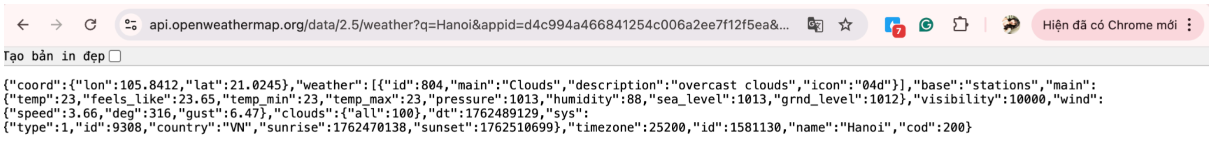


Recent searches save and load



City not found or network error.

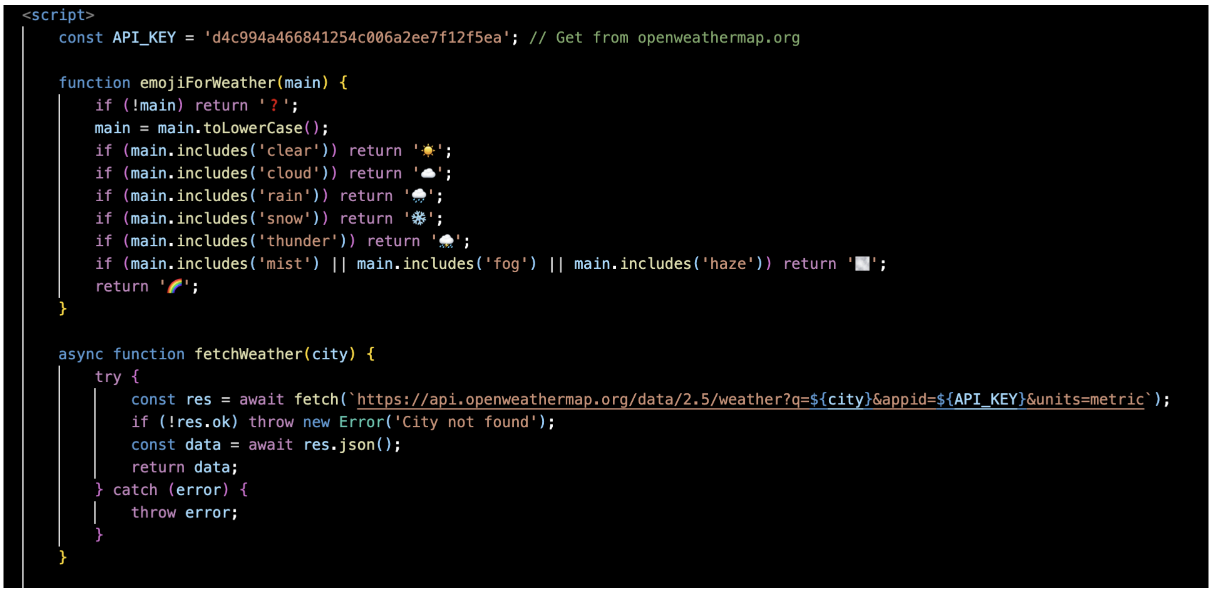
Error messages show for invalid cities


api.openweathermap.org/data/2.5/weather?q=Hanoi&appid=d4c994a466841254c006a2ee7f12f5ea&...
Hiện đã có Chrome mới
Tạo bản in đẹp

```
{"coord":{"lon":105.8412,"lat":21.0245},"weather":[{"id":804,"main":"Clouds","description":"overcast clouds","icon":"04d"}],"base":"stations","main":
{"temp":23,"feels_like":23.65,"temp_min":23,"temp_max":23,"pressure":1013,"humidity":88,"sea_level":1013,"grnd_level":1012},"visibility":10000,"wind":
{"speed":3.66,"deg":316,"gust":6.47},"clouds":{"all":100},"dt":1762489129,"sys":
{"type":1,"id":9308,"country":"VN","sunrise":1762470138,"sunset":1762510699},"timezone":25200,"id":1581130,"name":"Hanoi","cod":200}
```

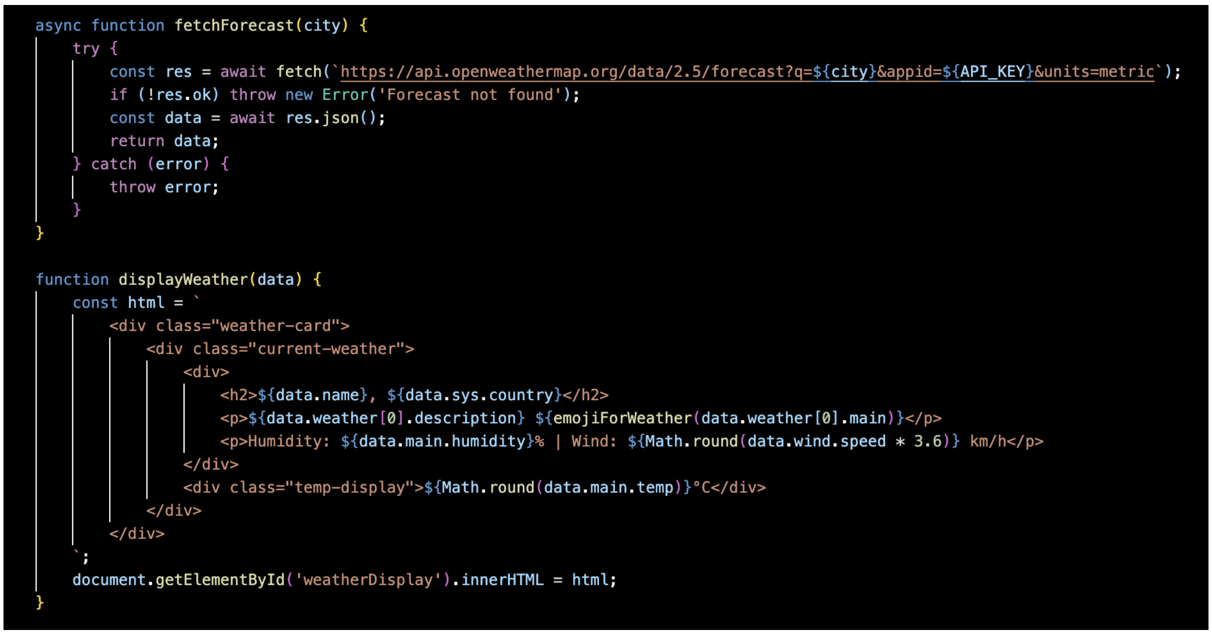
Enter key triggers search

## Checklist:



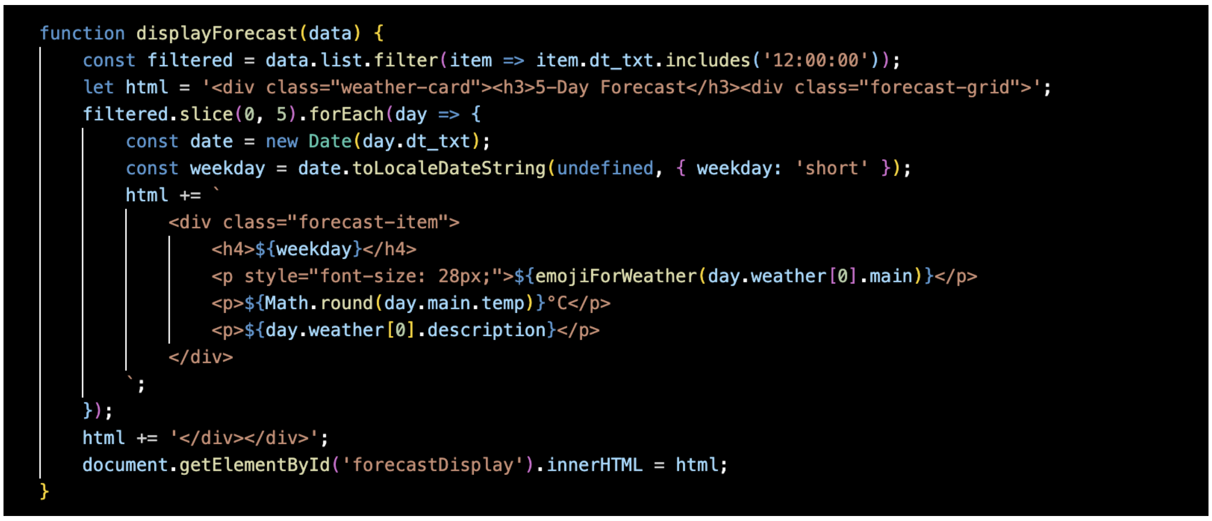
```
<script>
    const API_KEY = 'd4c994a466841254c006a2ee7f12f5ea'; // Get from openweathermap.org

    function emojiForWeather(main) {
        if (!main) return '❓';
        main = main.toLowerCase();
        if (main.includes('clear')) return '☀️';
        if (main.includes('cloud')) return '☁️';
        if (main.includes('rain')) return '🌧️';
        if (main.includes('snow')) return '❄️';
        if (main.includes('thunder')) return '⛈️';
        if (main.includes('mist') || main.includes('fog') || main.includes('haze')) return '🌫️';
        return '🌈';
    }

    async function fetchWeather(city) {
        try {
            const res = await fetch(`https://api.openweathermap.org/data/2.5/weather?q=${city}&appid=${API_KEY}&units=metric`);
            if (!res.ok) throw new Error('City not found');
            const data = await res.json();
            return data;
        } catch (error) {
            throw error;
        }
    }
```

- fetchWeather(): when users input the city name, this function will call the API OpenWeatherMap to get the data JSON about the forecast recently. -> Then get back data to displayWeather().



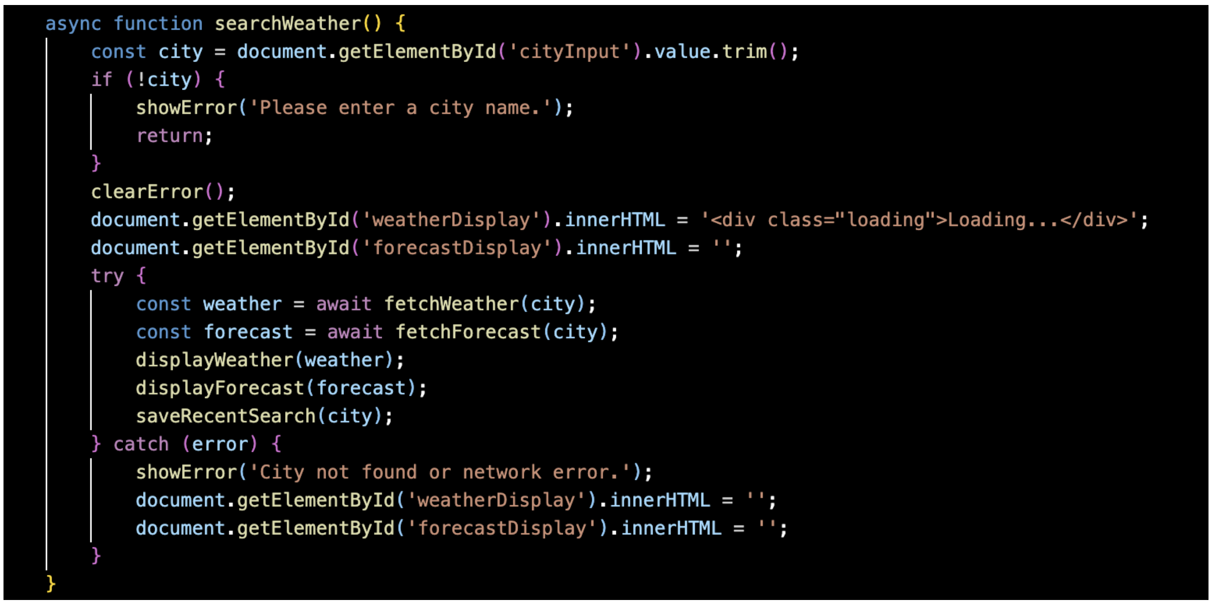
```
async function fetchForecast(city) {
    try {
        const res = await fetch(`https://api.openweathermap.org/data/2.5/forecast?q=${city}&appid=${API_KEY}&units=metric`);
        if (!res.ok) throw new Error('Forecast not found');
        const data = await res.json();
        return data;
    } catch (error) {
        throw error;
    }
}

function displayWeather(data) {
    const html = `
        <div class="weather-card">
            <div class="current-weather">
                <div>
                    <h2>${data.name}, ${data.sys.country}</h2>
                    <p>${data.weather[0].description} ${emojiForWeather(data.weather[0].main)}</p>
                    <p>Humidity: ${data.main.humidity}% | Wind: ${Math.round(data.wind.speed * 3.6)} km/h</p>
                </div>
                <div class="temp-display">${Math.round(data.main.temp)}°C</div>
            </div>
        </div>
    `;
    document.getElementById('weatherDisplay').innerHTML = html;
}
```

- fetchForecast(): Call the 5-days forecast API and files one forecast per day. Passes the data to displayForecast()



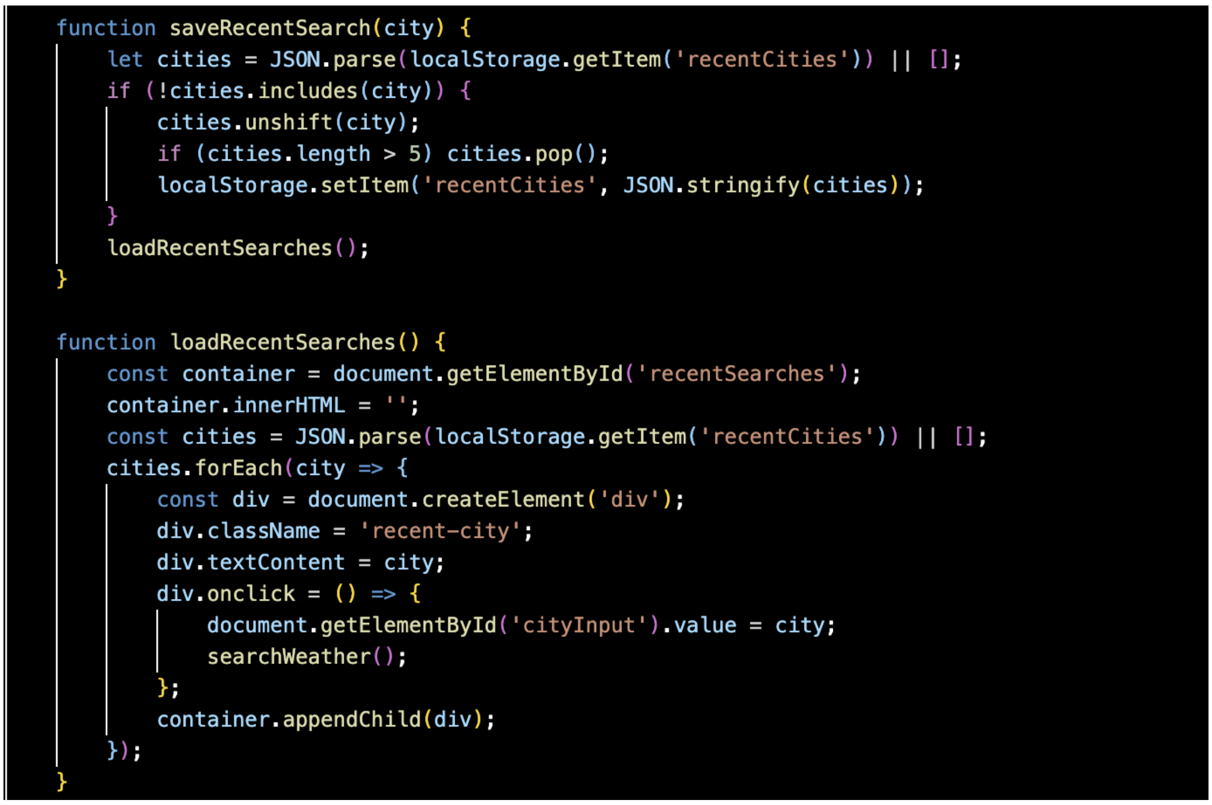
```javascript
function displayForecast(data) {
    const filtered = data.list.filter(item => item.dt_txt.includes('12:00:00'));
    let html = '<div class="weather-card"><h3>5-Day Forecast</h3><div class="forecast-grid">';
    filtered.slice(0, 5).forEach(day => {
        const date = new Date(day.dt_txt);
        const weekday = date.toLocaleDateString(undefined, { weekday: 'short' });
        html += `
            <div class="forecast-item">
                <h4>${weekday}</h4>
                <p style="font-size: 28px;">${emojiForWeather(day.weather[0].main)}</p>
                <p>${Math.round(day.main.temp)}°C</p>
                <p>${day.weather[0].description}</p>
            </div>
        `;
    });
    html += '</div></div>';
    document.getElementById('forecastDisplay').innerHTML = html;
}
```

- displayForecast(): loop through forecast items, take one entry per day, and create 5 small forecast boxes with emoji icons and temperatures.



```javascript
async function searchWeather() {
    const city = document.getElementById('cityInput').value.trim();
    if (!city) {
        showError('Please enter a city name.');
        return;
    }
    clearError();
    document.getElementById('weatherDisplay').innerHTML = '<div class="loading">Loading...</div>';
    document.getElementById('forecastDisplay').innerHTML = '';
    try {
        const weather = await fetchWeather(city);
        const forecast = await fetchForecast(city);
        displayWeather(weather);
        displayForecast(forecast);
        saveRecentSearch(city);
    } catch (error) {
        showError('City not found or network error.');
        document.getElementById('weatherDisplay').innerHTML = '';
        document.getElementById('forecastDisplay').innerHTML = '';
    }
}
```

- searchWeather(): the main controller function. Gets the city name from input -> call clearError() -> show a “Lading…” message -> runs both fetchWeather and fetchForecast () -> display result or calls showError() if something fails-> also saves the searc to localStorage.

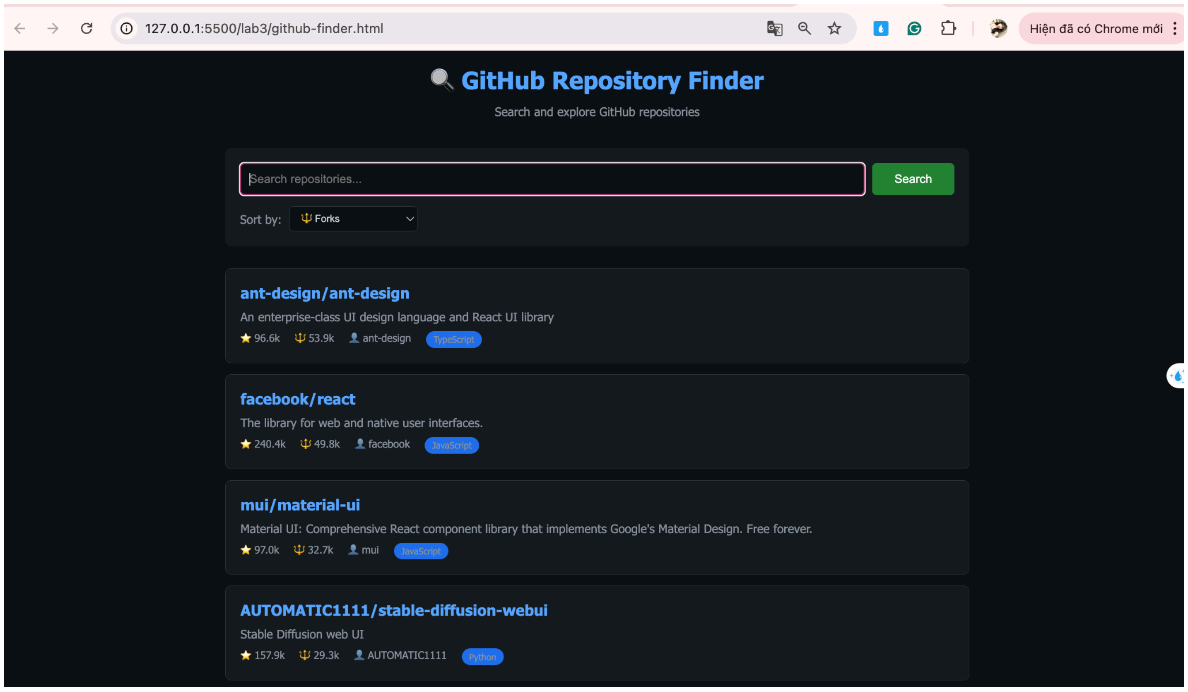


```javascript
function saveRecentSearch(city) {
    let cities = JSON.parse(localStorage.getItem('recentCities')) || [];
    if (!cities.includes(city)) {
        cities.unshift(city);
        if (cities.length > 5) cities.pop();
        localStorage.setItem('recentCities', JSON.stringify(cities));
    }
    loadRecentSearches();
}

function loadRecentSearches() {
    const container = document.getElementById('recentSearches');
    container.innerHTML = '';
    const cities = JSON.parse(localStorage.getItem('recentCities')) || [];
    cities.forEach(city => {
        const div = document.createElement('div');
        div.className = 'recent-city';
        div.textContent = city;
        div.onclick = () => {
            document.getElementById('cityInput').value = city;
            searchWeather();
        };
        container.appendChild(div);
    });
}
```

- saveRecentSearch(): read old research from localStorage when the app starts -> creates small clickable buttons -> clicking one trigger a new search for that city.

## 2.2 Github Finder

**Output**:



127.0.0.1:5500/lab3/github-finder.html

Hiện đã có Chrome mới

🔍 GitHub Repository Finder

Search and explore GitHub repositories

Search repositories...

Search

Sort by: Forks

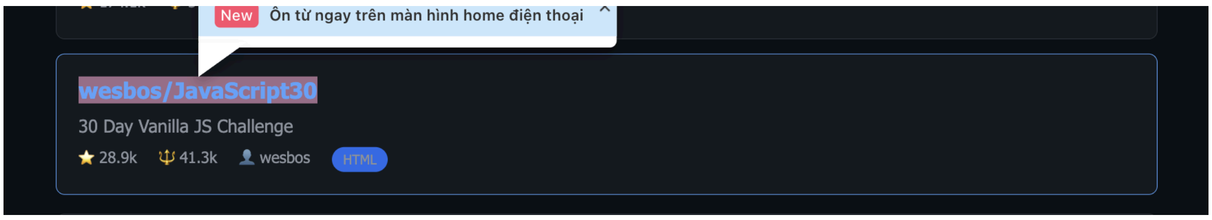
**ant-design/ant-design**
An enterprise-class UI design language and React UI library
★ 96.6k 53.9k ant-design TypeScript

**facebook/react**
The library for web and native user interfaces.
★ 240.4k 49.8k facebook JavaScript

**mui/material-ui**
Material UI: Comprehensive React component library that implements Google's Material Design. Free forever.
★ 97.0k 32.7k mui JavaScript

**AUTOMATIC1111/stable-diffusion-webui**
Stable Diffusion web UI
★ 157.9k 29.3k AUTOMATIC1111 Python

Search returns results

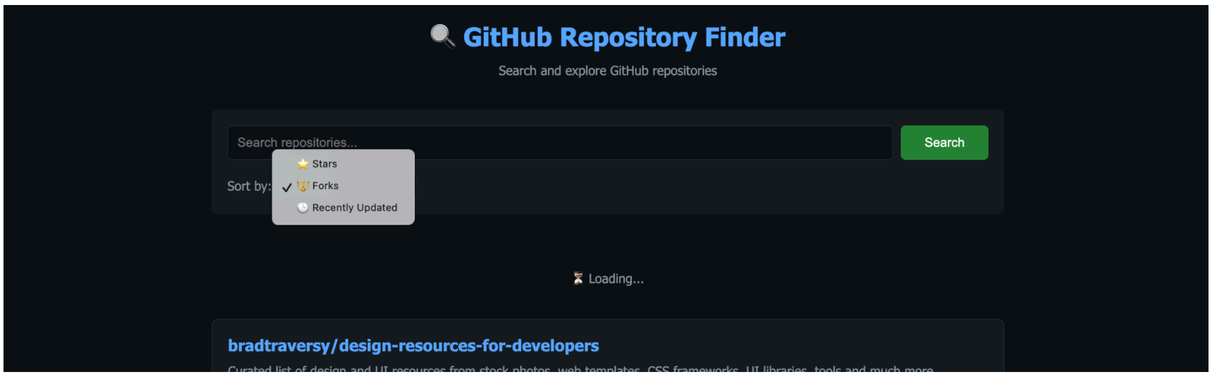

New Ôn từ ngay trên màn hình home điện thoại

wesbos/JavaScript30

30 Day Vanilla JS Challenge

★ 28.9k 41.3k wesbos HTML

Links open GitHub pages


GitHub Repository Finder

Search and explore GitHub repositories

Search repositories... Search

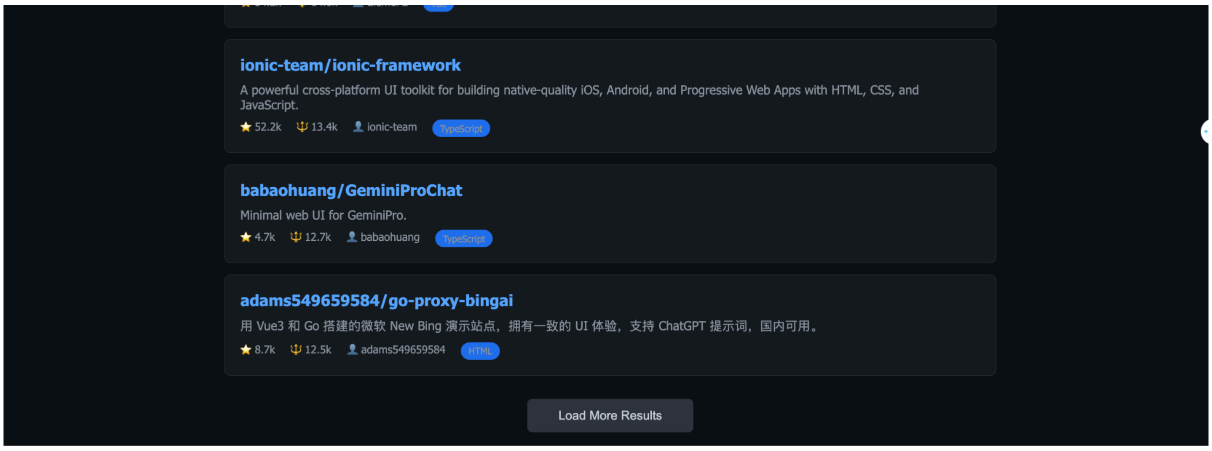
Sort by: Stars [✓ Forks] Recently Updated

Loading...

bradtraversy/design-resources-for-developers

Curated list of design and UI resources from stock photos, web templates, CSS frameworks, UI libraries, tools and much more

Sort dropdown works


ionic-team/ionic-framework

A powerful cross-platform UI toolkit for building native-quality iOS, Android, and Progressive Web Apps with HTML, CSS, and JavaScript.

★ 52.2k 13.4k ionic-team TypeScript

babaohuang/GeminiProChat

Minimal web UI for GeminiPro.

★ 4.7k 12.7k babaohuang TypeScript

adams549659584/go-proxy-bingai

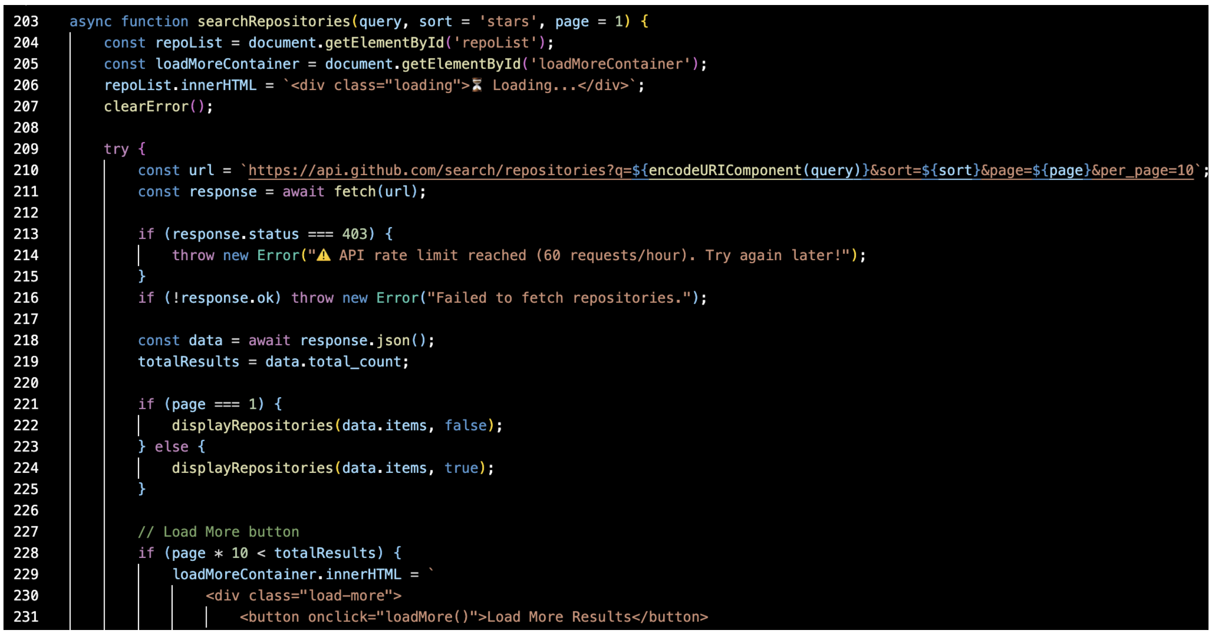
用 Vue3 和 Go 搭建的微软 New Bing 演示站点，拥有一致的 UI 体验，支持 ChatGPT 提示词，国内可用。

★ 8.7k 12.5k adams549659584 HTML

Load More Results

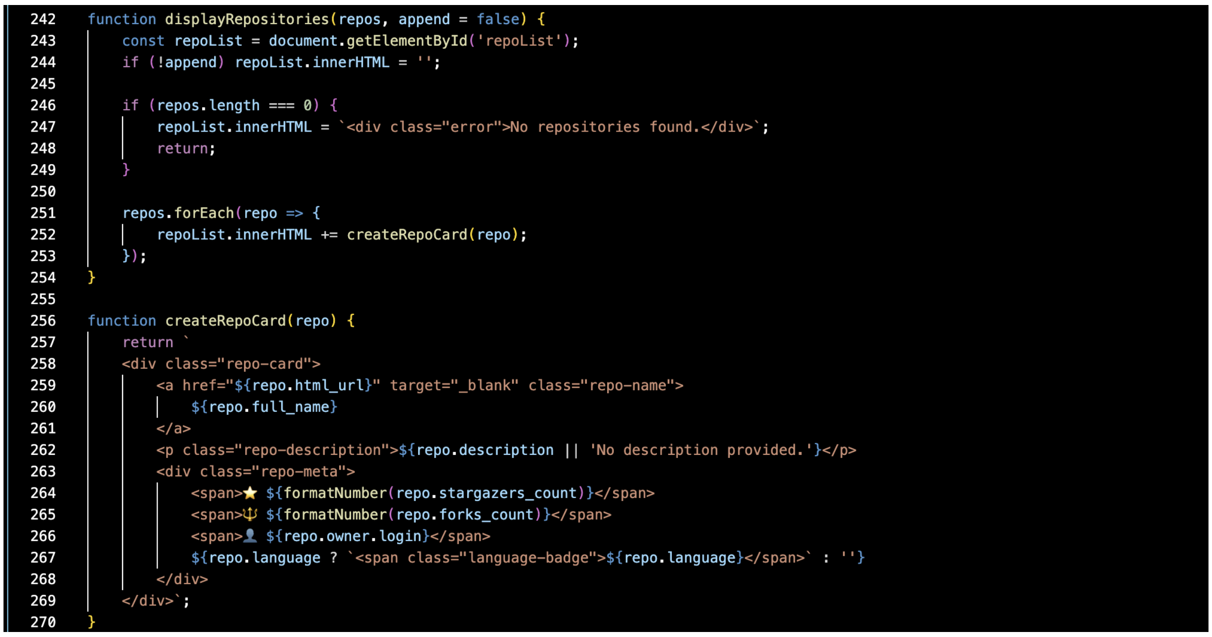
Load More fetches next page

## Checklist:



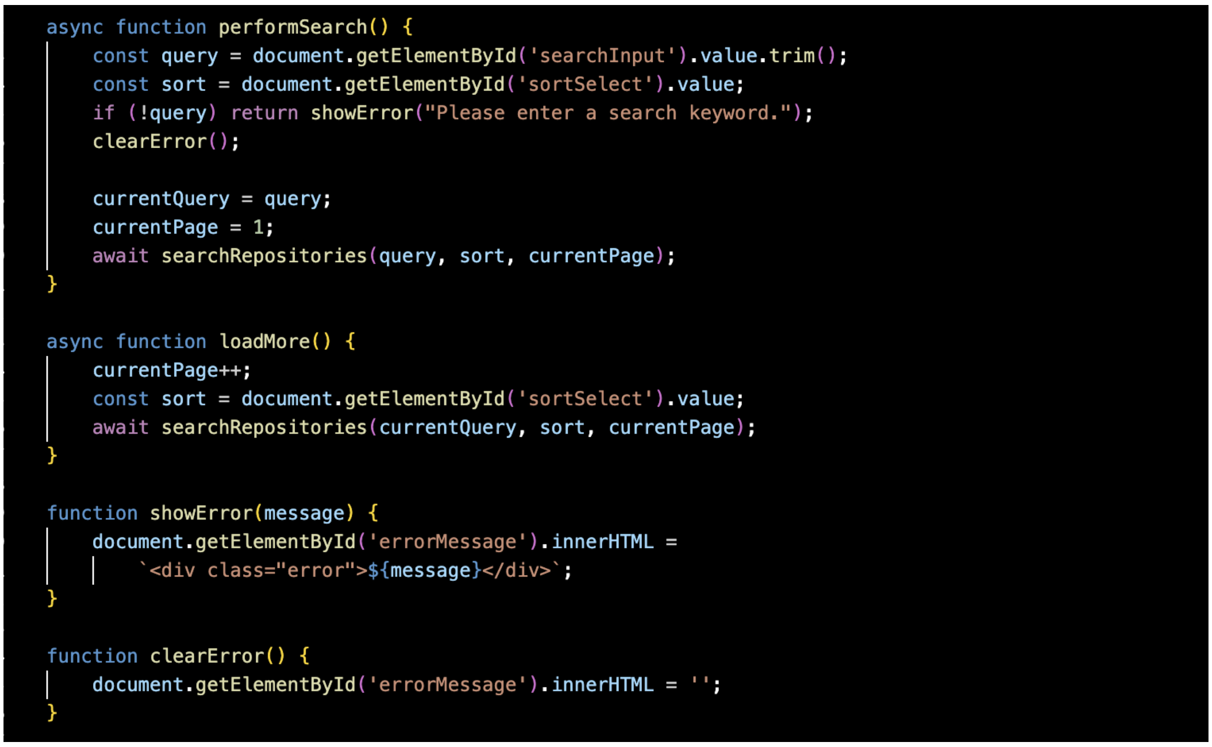
```
async function searchRepositories(query, sort = 'stars', page = 1) {
    const repoList = document.getElementById('repoList');
    const loadMoreContainer = document.getElementById('loadMoreContainer');
    repoList.innerHTML = `<div class="loading">⏳ Loading...</div>`;
    clearError();

    try {
        const url = `https://api.github.com/search/repositories?q=${encodeURIComponent(query)}&sort=${sort}&page=${page}&per_page=10`;
        const response = await fetch(url);

        if (response.status === 403) {
            throw new Error("⚠️ API rate limit reached (60 requests/hour). Try again later!");
        }
        if (!response.ok) throw new Error("Failed to fetch repositories.");

        const data = await response.json();
        totalResults = data.total_count;

        if (page === 1) {
            displayRepositories(data.items, false);
        } else {
            displayRepositories(data.items, true);
        }

        // Load More button
        if (page * 10 < totalResults) {
            loadMoreContainer.innerHTML = `
                <div class="load-more">
                    <button onclick="loadMore()">Load More Results</button>
```

- searchRespositorie: Use the GitHub Search AOI, with a search keyword, sort option and page number. Return JSON (items[]) to displayRespositories().



```
function displayRepositories(repos, append = false) {
    const repoList = document.getElementById('repoList');
    if (!append) repoList.innerHTML = '';

    if (repos.length === 0) {
        repoList.innerHTML = `<div class="error">No repositories found.</div>`;
        return;
    }

    repos.forEach(repo => {
        repoList.innerHTML += createRepoCard(repo);
    });
}

function createRepoCard(repo) {
    return `
    <div class="repo-card">
        <a href="${repo.html_url}" target="_blank" class="repo-name">
            ${repo.full_name}
        </a>
        <p class="repo-description">${repo.description || 'No description provided.'}</p>
        <div class="repo-meta">
            <span>⭐ ${formatNumber(repo.stargazers_count)}</span>
            <span>🔱 ${formatNumber(repo.forks_count)}</span>
            <span>👤 ${repo.owner.login}</span>
            ${repo.language ? `<span class="language-badge">${repo.language}</span>` : ''}
        </div>
    </div>`;
}
```

- displayRepositories(): take the list of repositories and loops through them -> for each one calls createRepoCard() to build HTML -> if append = false, replace old result, if true, adds new results below.



```javascript
async function performSearch() {
    const query = document.getElementById('searchInput').value.trim();
    const sort = document.getElementById('sortSelect').value;
    if (!query) return showError("Please enter a search keyword.");
    clearError();

    currentQuery = query;
    currentPage = 1;
    await searchRepositories(query, sort, currentPage);
}

async function loadMore() {
    currentPage++;
    const sort = document.getElementById('sortSelect').value;
    await searchRepositories(currentQuery, sort, currentPage);
}

function showError(message) {
    document.getElementById('errorMessage').innerHTML =
        `<div class="error">${message}</div>`;
}

function clearError() {
    document.getElementById('errorMessage').innerHTML = '';
}
```

- performSearch(): get user input and selected sort option -> resets currentPage = 1 -> calls searchRespositories() -> display result -> shows “Load More” button if there are additional pages.