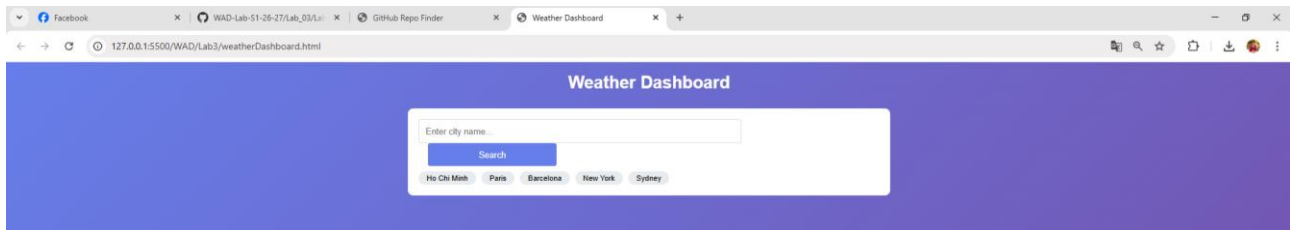


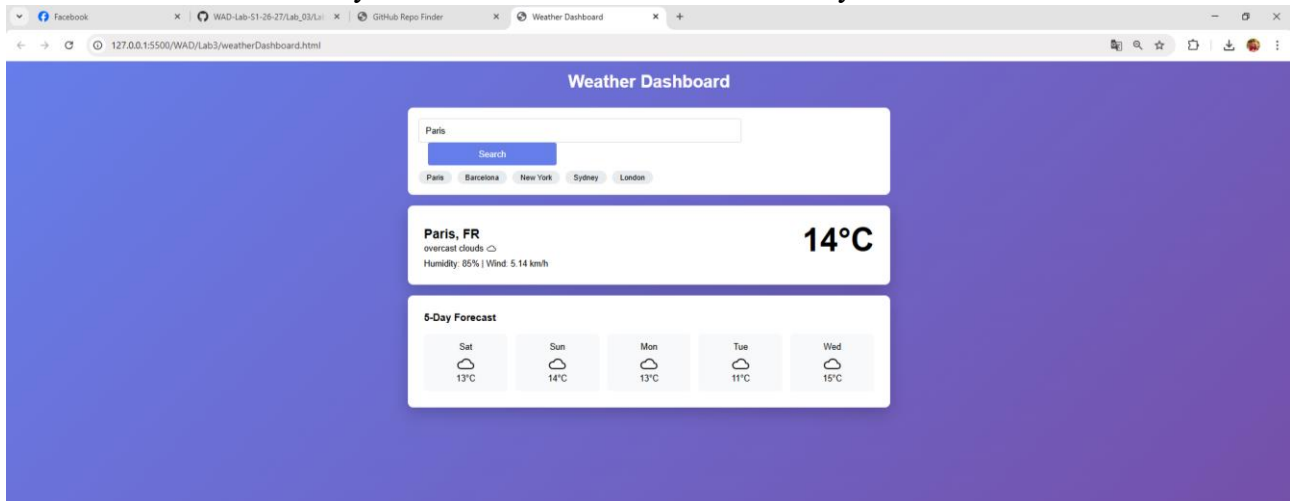
WEB DEVELOPMENT APPLICATION

LAB 3 HOMEWORK

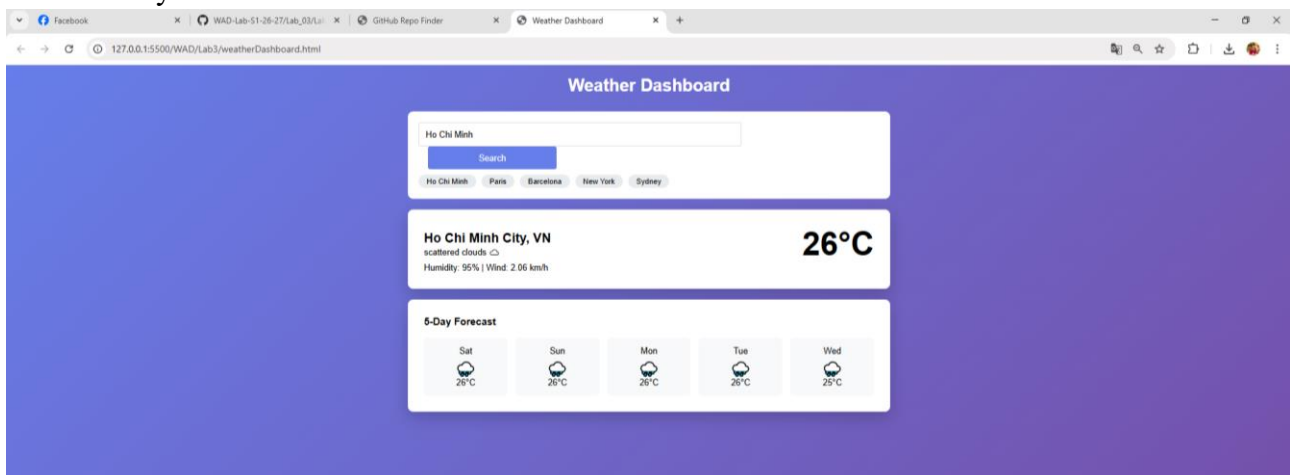
Task 2.1: Weather Dashboard



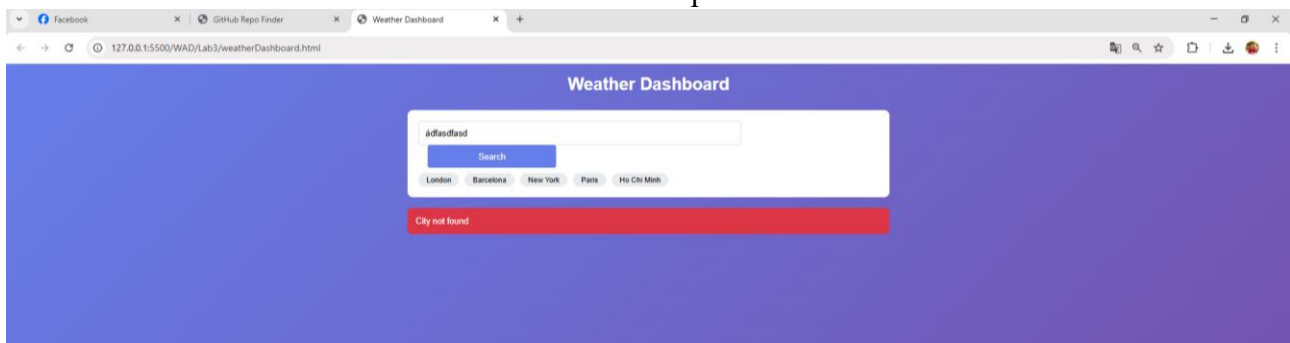
- The search finds city and save 5 cities which have recently been searched.



- When press the search button, the loading state appears.
- The display when searching “Paris” including the current weather displays and forecast shows 5 days.

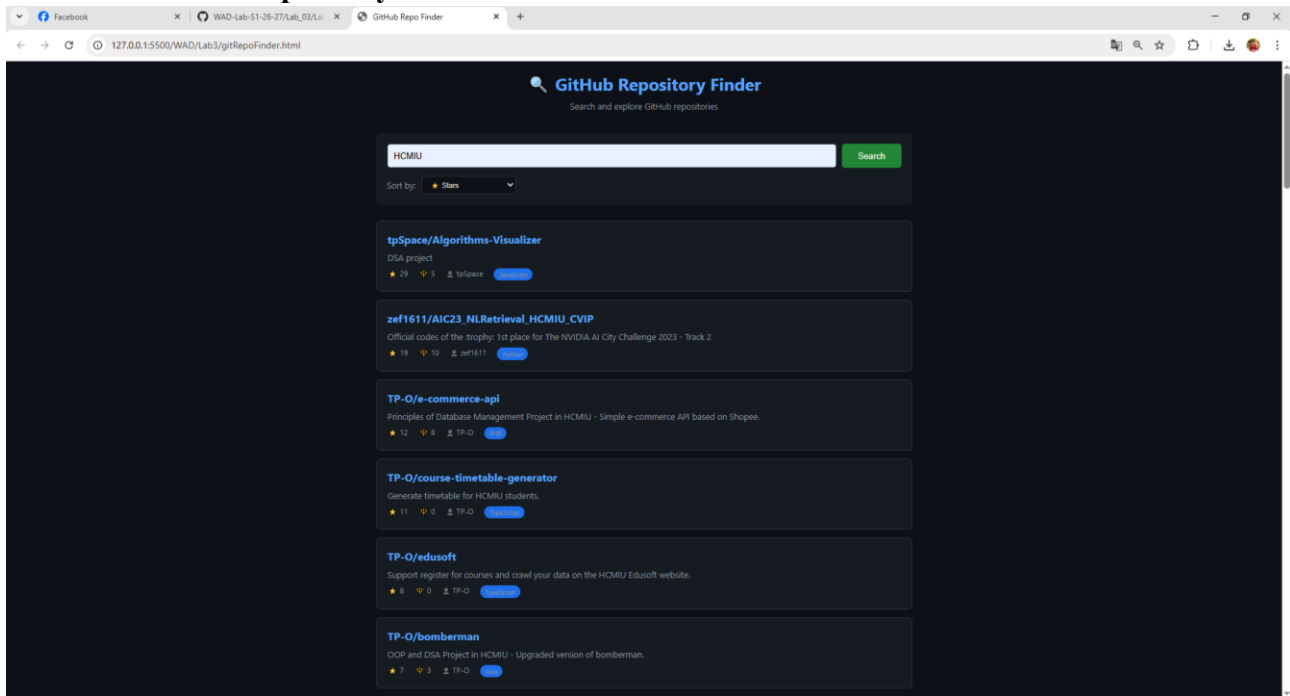


- The recent search “London” is removed and updated with Ho Chi Minh.

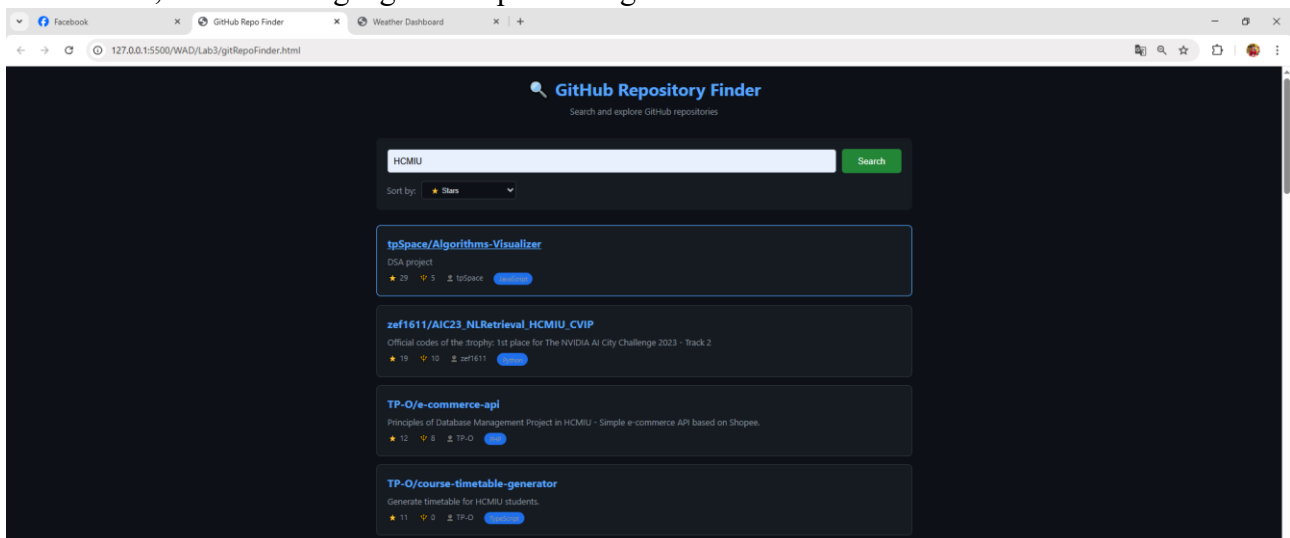


- The error message shows for invalid cities. It is said that the city is not found.

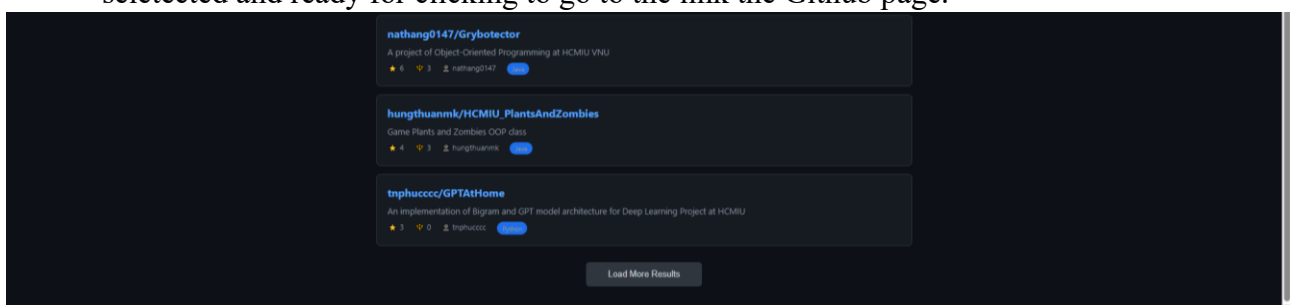
Task 2.2: Github Repository Finder



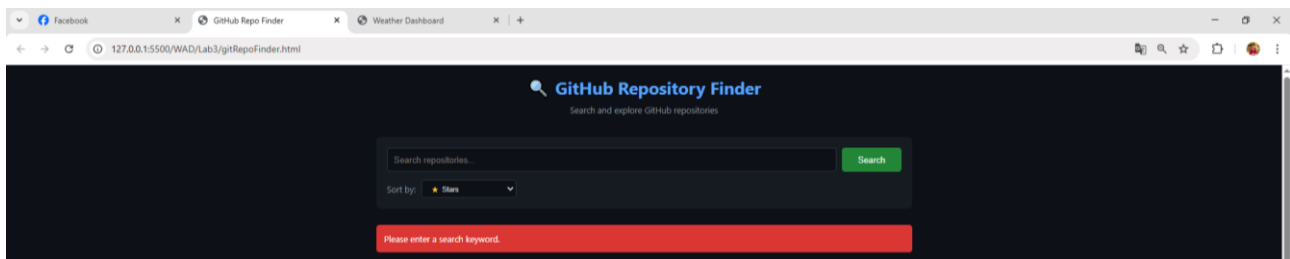
- When entering HCMIU, there are some results displayed which are related to the input given in the search field.
- The repository is displayed with all information including the name of the repo, the author or user, and what language this repo is using.



- When the mouse moves to the repo, it will pop up and underline, which means this repo is selected and ready for clicking to go to the link the Github page.



- The load more result button: When clicking this button, the webpage will display more results also relating to the word given in the search field.



- If the input of search field is empty, there is an error message that announce the user to enter the search keyword.

```
try {
  const response = await fetch(url);
  if (response.status === 403) {
    throw new Error("Rate limit exceeded. Please wait or use a GitHub API token.");
  }
  if (!response.ok) {
    throw new Error("Failed to fetch data from GitHub API.");
  }

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

  if (data.items.length === 0) {
    showError("No repositories found.");
    return [];
  }

  return data.items;
} catch (error) {
  showError(error.message);
  return [];
}
```

Error handling

- Using try-catch to handle errors.
- If GitHub returns a 403 error, that means you hit the rate limit. Therefore, we throw an error to be caught. Moreover, we also handle some HTTP status to make sure it works successfully.
- If there is an error, in catch will show that error message like the image above.

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

Display the error message in the red box

```
function clearError() {
  // TODO: Clear error message
  const errorDiv = document.getElementById('errorMessage');
  errorDiv.innerHTML = '';
}
```

Remove any error message before a new request