

2.1

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Weather Dashboard</title>
<style>
*{margin:0;padding:0;box-sizing:border-box}
body{
  font-family:Arial,Helvetica,sans-serif;
  background:linear-gradient(135deg,#667eea
0%,#764ba2 100%);
  min-height:100vh;
  padding:20px;
}
.container{max-width:900px;margin:0 auto}
.search-box{
  background:#fff;
  padding:20px;
  border-radius:10px;
  margin-bottom:20px;
}
.search-box input{
  width:70%;
  padding:12px;
  border:2px solid #ddd;
```

```
border-radius:4px;
font-size:16px;
}
.search-box button{
width:28%;
padding:12px;
background:#667eea;
color:#fff;
border:none;
border-radius:4px;
cursor:pointer;
font-size:16px;
margin-left:2%;
}
.weather-card{
background:#fff;
padding:30px;
border-radius:10px;
box-shadow:0 10px 30px rgba(0,0,0,0.2);
margin-bottom:20px;
}
.current-weather{
display:flex;
justify-content:space-between;
align-items:center;
}
.temp-display{font-size:60px;font-weight:bold}
.forecast-grid{
```

```
display:grid;
grid-template-columns:repeat(auto-fit,minmax(150px,1fr));
gap:15px;
margin-top:20px;
}
.forecast-item{
background:#f8f9fa;
padding:15px;
border-radius:8px;
text-align:center;
}
.loading{text-align:center;padding:40px;font-size:18px;color:#fff}
.error{
background:#dc3545;
color:#fff;
padding:15px;
border-radius:8px;
margin-bottom:20px;
}
.recent-searches{
display:flex;
gap:10px;
margin-top:10px;
flex-wrap:wrap;
}
.recent-city{
```

```
padding:5px 15px;
background:#e9ecef;
border-radius:20px;
cursor:pointer;
font-size:14px;
}
.recent-city:hover{background:#dee2e6}
</style>
</head>
<body>
<div class="container">
  <h1 style="color:white;text-align:center;margin-
bottom:30px;">Weather Dashboard</h1>
  <div class="search-box">
    <input type="text" id="cityInput"
placeholder="Enter city name..." 
onkeypress="if(event.key==='Enter')
searchWeather()">
    <button
onclick="searchWeather()">Search</button>
    <div class="recent-searches"
id="recentSearches"></div>
  </div>
  <div id="errorMessage"></div>
  <div id="weatherDisplay"></div>
  <div id="forecastDisplay"></div>
</div>
```

```
<script>

const API_KEY = 'b29b0bd15d0ec0aa7af69ec6115ad068';
const weatherDisplay =
document.getElementById("weatherDisplay");
const forecastDisplay =
document.getElementById("forecastDisplay");
const errorMessage =
document.getElementById("errorMessage");
const recentContainer =
document.getElementById("recentSearches");

function getIcon(desc){
  desc = desc.toLowerCase();
  if(desc.includes("cloud")) return "☁";
  if(desc.includes("rain")) return "🌧";
  if(desc.includes("clear")) return "☀";
  if(desc.includes("snow")) return "❄";
  if(desc.includes("storm")) return "⛈";
  return "☀";
}

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

```
}

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

function displayWeather(data){
  const icon = getIcon(data.weather[0].main);
  weatherDisplay.innerHTML = `

<div class="weather-card">
  <div class="current-weather">
    <div>
      <h2>${data.name}, ${data.sys.country}</h2>
      <p>${data.weather[0].description}
${icon}</p>
      <p>Humidity: ${data.main.humidity}% | Wind:
${data.wind.speed} km/h</p>
    </div>
    <div class="temp-
display">${Math.round(data.main.temp)}°C</div>
  </div>
</div>`;
```

```
}

function displayForecast(data){

  const days = {};
  data.list.forEach(item=>{
    const date = item.dt_txt.split(" ")[0];
    const hour = item.dt_txt.split(" ")[1];
    if(hour==="12:00:00" && !days[date])
      days[date]=item;
  });
  forecastDisplay.innerHTML = `<div class="weather-card"><h3>5-Day Forecast</h3>
<div class="forecast-grid">
${Object.values(days).slice(0,5).map(d=>{
  const day = new
Date(d.dt_txt).toLocaleDateString("en", {weekday:"short"});
  const icon = getIcon(d.weather[0].main);
  return `<div class="forecast-item">
<h4>${day}</h4>
<p>${icon}</p>
<p>${Math.round(d.main.temp)}°C</p>
</div>`;
}).join("")}
</div></div>`;
}
```

```
async function searchWeather(){
  const city =
document.getElementById("cityInput").value.trim();
  if(!city) return;
  weatherDisplay.innerHTML = '<div
class="loading">Loading...</div>';
  forecastDisplay.innerHTML = "";
  clearError();
  try{
    const weatherData = await fetchWeather(city);
    const forecastData = await fetchForecast(city);
    displayWeather(weatherData);
    displayForecast(forecastData);
    saveRecentSearch(city);
  }catch(err){
    showError(err.message);
    weatherDisplay.innerHTML = "";
  }
}

function saveRecentSearch(city){
  let cities =
JSON.parse(localStorage.getItem("recentCities")) ||
[];
  cities = [city,
...cities.filter(c=>c.toLowerCase() !== city.toLowerCase())];
  if(cities.length>5) cities=cities.slice(0,5);
```

```
localStorage.setItem("recentCities", JSON.stringify(cities));
  loadRecentSearches();
}

function loadRecentSearches(){
  const cities =
JSON.parse(localStorage.getItem("recentCities")) || [];
  recentContainer.innerHTML = cities.map(c=>`<div
class="recent-city"
onclick="selectCity('${c}')">${c}</div>`).join("");
}

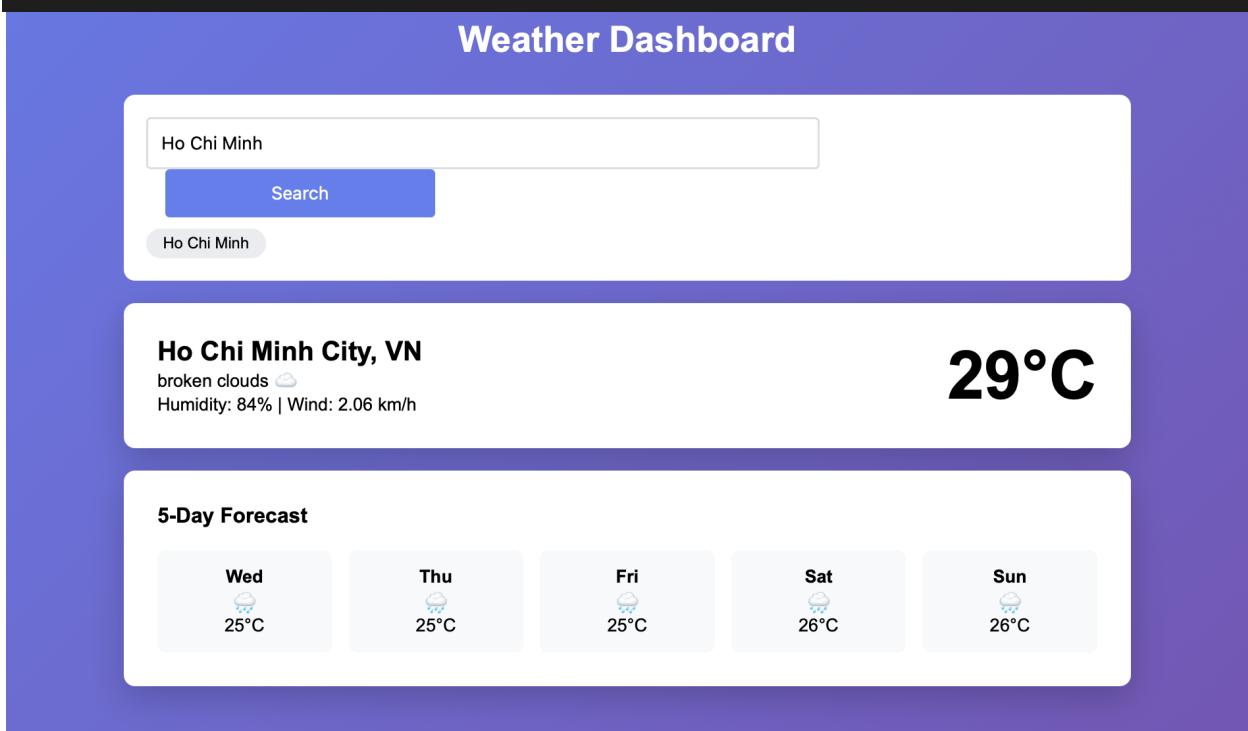
function selectCity(city){
  document.getElementById("cityInput").value = city;
  searchWeather();
}

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

function clearError(){errorMessage.innerHTML=""}

loadRecentSearches();
```

```
</script>
</body>
</html>
```



2.2

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>GitHub Repo Finder</title>
  <style>
    *{margin:0;padding:0;box-sizing:border-box}
    body{
      font-family: 'Segoe
      UI', Tahoma, Geneva, Verdana, sans-serif;
```

```
background:#0d1117;
color:#c9d1d9;
padding:20px;
}
.container{max-width:1000px;margin:0 auto}
.header{text-align:center;margin-bottom:40px}
.header h1{color:#58a6ff;margin-bottom:10px}
.search-container{
background:#161b22;
padding:20px;
border-radius:8px;
margin-bottom:30px;
}
.search-box{display:flex;gap:10px;margin-
bottom:15px}
.search-box input{
flex:1;
padding:12px;
background:#0d1117;
border:1px solid #30363d;
border-radius:6px;
color:#c9d1d9;
font-size:16px;
}
.search-box button{
padding:12px 30px;
background:#238636;
color:white;
```

```
border:none;
border-radius:6px;
cursor:pointer;
font-size:16px;
}
.search-box button:hover{background:#2ea043}
.filters{display:flex;gap:10px;align-items:center}
.filters label{color:#8b949e}
.filters select{
padding:8px 12px;
background:#0d1117;
border:1px solid #30363d;
border-radius:6px;
color:#c9d1d9;
}
.repo-card{
background:#161b22;
border:1px solid #30363d;
border-radius:8px;
padding:20px;
margin-bottom:15px;
transition:transform 0.2s;
}
.repo-card:hover{transform:translateY(-2px);border-color:#58a6ff}
.repo-name{
color:#58a6ff;
```

```
    font-size:20px;
    font-weight:bold;
    text-decoration:none;
}
.repo-name:hover{text-decoration:underline}
.repo-description{color:#8b949e;margin:10px 0}
.repo-meta{display:flex;gap:20px;font-size:14px;color:#8b949e}
.language-badge{
    display:inline-block;
    padding:4px 10px;
    background:#1f6feb;
    border-radius:12px;
    font-size:12px;
}
.loading{text-align:center;padding:40px;color:#8b949e}
.error{
    background:#da3633;
    color:white;
    padding:15px;
    border-radius:6px;
    margin-bottom:20px;
}
.load-more{text-align:center;margin:30px 0}
.load-more button{
    padding:12px 40px;
    background:#21262d;
```

```
border:1px solid #30363d;
color:#c9d1d9;
border-radius:6px;
cursor:pointer;
font-size:16px;
}
.load-more button:hover{background:#30363d}
</style>
</head>
<body>
<div class="container">
<div class="header">
<h1>🔍 GitHub Repository Finder</h1>
<p style="color:#8b949e;">Search and explore
GitHub repositories</p>
</div>
<div class="search-container">
<div class="search-box">
<input type="text" id="searchInput"
placeholder="Search repositories..." 
onkeypress="if(event.key==='Enter') 
performSearch()">
<button
onclick="performSearch()">Search</button>
</div>
<div class="filters">
<label>Sort by:</label>
```

```
    <select id="sortSelect"
onchange="performSearch()">
    <option value="stars">★ Stars</option>
    <option value="forks">⚡ Forks</option>
    <option value="updated">🕒 Recently
    Updated</option>
  </select>
</div>
</div>
<div id="errorMessage"></div>
<div id="repoList"></div>
<div id="loadMoreContainer"></div>
</div>

<script>
  let currentPage = 1;
  let currentQuery = "";
  let currentSort = "stars";

  async function searchRepositories(query, sort =
"stars", page = 1) {
    const url =
`https://api.github.com/search/repositories?q=${enc
odeURIComponent(query)}&sort=${sort}&order=desc&pag
e=${page}&per_page=10`;
    const res = await fetch(url);
    if (!res.ok) throw new Error("Failed to fetch
from GitHub API");
  }
</script>
```

```
    return res.json();
}

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

    repos.forEach(repo => {
        const card = document.createElement("div");
        card.className = "repo-card";
        card.innerHTML = `

            <a href="${repo.html_url}"
target="_blank" class="repo-
name">${repo.full_name}</a>
            <p class="repo-
description">${repo.description || "No
description"}</p>
            <div class="repo-meta">
                <span>★
${formatNumber(repo.stargazers_count)}</span>
                <span>⚡
${formatNumber(repo.forks_count)}</span>
                ${repo.language ? `<span
class="language-badge">${repo.language}</span>` :
""}
            </div>
        
```

```
        `;
        repoList.appendChild(card);
    });
}

function formatNumber(num) {
    if (num >= 1000) return (num /
1000).toFixed(1) + "k";
    return num;
}

async function performSearch() {
    const query =
document.getElementById("searchInput").value.trim()
;
    if (!query) return;
    currentQuery = query;
    currentSort =
document.getElementById("sortSelect").value;
    currentPage = 1;
    clearError();
    document.getElementById("repoList").innerHTML
= '<div class="loading">Loading...</div>';
    document.getElementById("loadMoreContainer").innerHTML
= "";
    try {
```

```
        const data = await
searchRepositories(query, currentSort,
currentPage);
        if (data.items.length === 0) {
            showError("No repositories found.");
            return;
        }
        displayRepositories(data.items);
        showLoadMoreButton(data.total_count > 10);
    } catch (err) {
        showError(err.message);
    }
}

async function loadMore() {
    currentPage++;

document.getElementById("loadMoreContainer").innerHTML = '<div class="loading">Loading more...</div>';
    try {
        const data = await
searchRepositories(currentQuery, currentSort,
currentPage);
        displayRepositories(data.items, true);
        showLoadMoreButton(data.items.length ===
10);
    } catch (err) {
        showError(err.message);
    }
}
```

```
        }
    }

    function showLoadMoreButton(show) {
        const container =
document.getElementById("loadMoreContainer");
        container.innerHTML = show
            ? '<button onclick="loadMore()">Load More
Results</button>'
            : "";
    }

    function showError(message) {

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

    function clearError() {

document.getElementById("errorMessage").innerHTML =
"";
    }
}

</script>
</body>
</html>
```

# GitHub Repository Finder

Search and explore GitHub repositories

Search

Sort by: Stars

**dtgazmuri/WebAppsII-Lab3**  
No description  
★ 2 ⚡ 0 Kotlin

**begletsoid/Lab3WebApp**  
Lab3 webApp  
★ 0 ⚡ 0 Java

**jackOne63/webapp\_lab3**  
No description  
★ 0 ⚡ 0 Python

**QuocAnhDiEm/WebApp\_lab3**  
No description  
★ 0 ⚡ 0 HTML

**THMinh26/WebApp\_Lab3**  
No description  
★ 0 ⚡ 0 HTML