

# **Reddit Browser - Progressive Building**

Build Step-by-Step with Real Features

Tech Stack Fundamentals Session - CSE 24 Batch

19th December 2025

# Contents

<b>1</b>	<b>Step 0: Setup Local Server (Important!)</b>	<b>5</b>
1.1	Option A: Python (Recommended) . . . . .	5
1.2	Option B: VS Code Live Server . . . . .	5
1.3	Option C: Node.js . . . . .	5
1.4	Critical Warning . . . . .	5
<b>2</b>	<b>Step 1: Get Data from Reddit API (JavaScript Only)</b>	<b>6</b>
2.1	Step 1.1: Setup . . . . .	6
2.2	Step 1.2: Your Task . . . . .	6
2.3	Step 1.3: Test . . . . .	7
<b>3</b>	<b>Step 2: Display Post Titles in HTML List</b>	<b>8</b>
3.1	Step 2.1: Add HTML Container . . . . .	8
3.2	Step 2.2: Your Task . . . . .	8
3.3	Step 2.3: Expected Output . . . . .	9
<b>4</b>	<b>Step 3: Show Posts as Cards with Images</b>	<b>10</b>
4.1	Step 3.1: Update HTML . . . . .	10
4.2	Step 3.2: Your Task . . . . .	11
4.3	Step 3.3: Expected Output . . . . .	12
<b>5</b>	<b>Step 4: Add Custom Subreddit Input</b>	<b>13</b>
5.1	Step 4.1: Update HTML . . . . .	13
5.2	Step 4.2: Your Task . . . . .	14
5.3	Step 4.3: Test It . . . . .	15
<b>6</b>	<b>Step 5: Custom Styling (Make It Beautiful)</b>	<b>16</b>
6.1	Step 5.1: Your Task . . . . .	16
6.2	Step 5.2: Update HTML Structure . . . . .	19
6.3	Step 5.3: Test . . . . .	19
<b>7</b>	<b>Step 6 (Optional): Add Video Playback</b>	<b>21</b>
7.1	Step 6.1: Detect Video Posts . . . . .	21
7.2	Step 6.2: Your Task . . . . .	21
<b>8</b>	<b>Final Project Summary</b>	<b>23</b>
<b>9</b>	<b>How to Submit Your Work</b>	<b>24</b>
9.1	Step 1: Create Submission Folder . . . . .	24
9.2	Step 2: Save Your File . . . . .	24
9.3	Step 3: Commit and Push . . . . .	24
9.4	Step 4: Verify . . . . .	25
9.5	Submission Checklist . . . . .	26

<b>10 Troubleshooting</b>	<b>27</b>
10.1 CORS Error - Posts don't appear . . . . .	27
10.2 Posts don't appear (after local server) . . . . .	27
10.3 Images broken . . . . .	27
10.4 Search not working . . . . .	27
10.5 Cannot push to GitHub . . . . .	27
<b>11 Additional Resources</b>	<b>28</b>
11.1 Helpful Links . . . . .	28
11.2 Try These Subreddits . . . . .	28
<b>12 What's Next?</b>	<b>29</b>

# Project Philosophy

**Instead of learning concepts in isolation, you'll build a complete website incrementally.**

Each step adds one feature to your existing project. By the end, you have a fully functional Reddit browser.

## **Why this approach?**

- See real results at each step
- Understand how HTML, CSS, JS work together
- Build confidence progressively
- No boring "just learn theory" sections
- Deploy working features immediately

# 1 Step 0: Setup Local Server (Important!)

**Read this FIRST before coding!**

You need to run a local server to avoid CORS (Cross-Origin Resource Sharing) errors when fetching Reddit data.

## 1.1 Option A: Python (Recommended)

Open terminal in your project folder and run:

```
1 # Python 3
2 python -m http.server 8000
3
4 # Python 2
5 python -m SimpleHTTPServer 8000
```

Then open your browser: <http://localhost:8000>

**Why?** This solves CORS errors and allows API calls to Reddit!

## 1.2 Option B: VS Code Live Server

1. Install “Live Server” extension in VS Code
2. Right-click `index.html`
3. Click “Open with Live Server”

Automatically opens at <http://localhost:5500>

## 1.3 Option C: Node.js

```
1 npx http-server
```

Then open: <http://localhost:8080>

## 1.4 Critical Warning

**DO NOT open file directly!**

- WRONG: Double-click `index.html`
- WRONG: Open with `file://`
- CORRECT: Run local server first
- CORRECT: Then access via <http://localhost:8000>

If you see this error: “Access to fetch has been blocked by CORS policy”

**Solution:** You forgot to run the local server! Go back and follow Option A, B, or C above.

## 2 Step 1: Get Data from Reddit API (JavaScript Only)

**Goal:** Fetch Reddit data and print it to browser console.

**Time:** 15 minutes

**What you'll do:**

1. Create HTML file with a script
2. Fetch hardcoded URL from Reddit
3. Console.log the results
4. See raw data structure

### 2.1 Step 1.1: Setup

Create index.html:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Reddit Browser</title>
5 </head>
6 <body>
7   <h1>Reddit Browser - Step 1</h1>
8   <p>Open Developer Tools (F12) and check Console tab</p>
9
10  <script>
11    // We'll write code here
12  </script>
13 </body>
14 </html>
```

### 2.2 Step 1.2: Your Task

In the <script> section, write code to:

1. Create a hardcoded URL to Reddit API:

```
1 const url = 'https://www.reddit.com/r/programming.json';
2
```

2. Use fetch to get the data:

```
1 // TODO: Write async function that:
2 // 1. Fetches the URL
3 // 2. Converts to JSON
4 // 3. Logs to console with console.log()
5
```

3. Call the function when page loads:

```
1 window.addEventListener('load', () => {  
2   // TODO: Call your fetch function  
3 });  
4
```

## 2.3 Step 1.3: Test

1. Make sure you started your local server (Step 0)!
2. Open `http://localhost:8000` in browser
3. Press F12 to open Developer Tools
4. Go to “Console” tab
5. You should see the JSON data printed
6. Expand the data and explore the structure
7. Find: `data.children` contains the posts

### Success criteria:

Local server is running

Page loads in browser

Console shows data

Data structure visible

No red errors

## 3 Step 2: Display Post Titles in HTML List

**Goal:** Show Reddit post titles on the webpage (as a simple list).

**Time:** 20 minutes

**Building on Step 1:** We now take the data we fetched and display it.

### 3.1 Step 2.1: Add HTML Container

Update your HTML body:

```
1 <body>
2   <h1>Reddit Browser - Step 2</h1>
3
4   <!-- Container to display posts -->
5   <div id="posts-container">
6     <!-- Posts will appear here -->
7   </div>
8
9   <script>
10     // Previous code + new code
11   </script>
12 </body>
```

### 3.2 Step 2.2: Your Task

Update your JavaScript:

1. Keep your fetch function from Step 1
2. Create a new function to display posts:

```
1 function displayPosts(posts) {
2   // TODO:
3   // 1. Get the container with id 'posts-container'
4   // 2. Loop through posts array
5   // 3. Create <li> element for each post title
6   // 4. Add to container
7 }
8
```

3. Modify your fetch function to call displayPosts:

```
1 // Instead of console.log(data)
2 // TODO: Call displayPosts(data.data.children)
3
```



### 3.3 Step 2.3: Expected Output

Your page should display:

Reddit Browser - Step 2

- Post Title 1
- Post Title 2
- Post Title 3
- ...

#### Success criteria:

Posts appear on page

At least 10 posts visible

Titles are readable

No JavaScript errors

## 4 Step 3: Show Posts as Cards with Images

**Goal:** Make it look professional with post cards including images, upvotes, comments.

**Time:** 25 minutes

**What changes:**

- Replace list with card layout
- Add upvotes and comments count
- Add author name
- Add thumbnail image
- Better styling

### 4.1 Step 3.1: Update HTML

```
1 <body>
2   <h1>Reddit Browser - Step 3</h1>
3
4   <div id="posts-container" class="posts-grid">
5     <!-- Post cards will appear here -->
6   </div>
7
8   <style>
9     .posts-grid {
10       display: grid;
11       grid-template-columns: repeat(auto-fill, minmax(300px, 1
fr));
12       gap: 20px;
13       padding: 20px;
14     }
15
16     .post-card {
17       border: 1px solid #ddd;
18       border-radius: 8px;
19       padding: 15px;
20       background: white;
21       box-shadow: 0 2px 8px rgba(0,0,0,0.1);
22     }
23
24     .post-image {
25       width: 100%;
26       height: 200px;
27       object-fit: cover;
28       border-radius: 4px;
29       margin-bottom: 10px;
```

```

30     }
31
32     .post-title {
33         font-size: 1.1rem;
34         font-weight: bold;
35         margin-bottom: 10px;
36     }
37
38     .post-stats {
39         font-size: 0.9rem;
40         color: #666;
41     }
42 </style>
43
44 <script>
45     // Code goes here
46 </script>
47 </body>

```

## 4.2 Step 3.2: Your Task

Update displayPosts function:

```

1 function displayPosts(posts) {
2     const container = document.getElementById('posts-container');
3     container.innerHTML = ''; // Clear previous
4
5     posts.forEach(post => {
6         const data = post.data;
7
8         // TODO: Create HTML for each post card:
9         // <div class="post-card">
10        //   <img src={thumbnail} class="post-image">
11        //   <div class="post-title">{title}</div>
12        //   <div class="post-stats">
13        //     {ups} upvotes | {num_comments} comments | by u/{
author}
14        //   </div>
15        // </div>
16
17        // TODO: Add to container
18    });
19 }

```

Key data fields:

- data.title — Post title

- `data.ups` — Upvotes
- `data.num_comments` — Comments
- `data.author` — Username
- `data.thumbnail` — Image URL

### 4.3 Step 3.3: Expected Output

Beautiful grid of post cards with:

- Thumbnail images
- Post titles
- Stats (upvotes, comments, author)

**Success criteria:**

Cards display in grid

Images show correctly

Stats visible

Responsive (works on mobile)

No broken images

## 5 Step 4: Add Custom Subreddit Input

**Goal:** Let user type subreddit name and search for it.

**Time:** 20 minutes

**What changes:**

- Add input field for subreddit
- Add search button
- Change hardcoded URL to dynamic
- Support multiple subreddits

### 5.1 Step 4.1: Update HTML

Add search box at top:

```
1 <body>
2   <h1>Reddit Browser - Step 4</h1>
3
4   <!-- Search Box -->
5   <div id="search-box" class="search-box">
6     <input
7       type="text"
8       id="subreddit-input"
9       placeholder="Enter subreddit (e.g., programming)"
10      value="programming"
11    >
12    <button onclick="loadPosts()">Search</button>
13  </div>
14
15  <!-- Loading indicator -->
16  <div id="loading" style="display: none; text-align: center;">
17    Loading posts...
18  </div>
19
20  <!-- Posts container -->
21  <div id="posts-container" class="posts-grid">
22  </div>
23
24  <style>
25    .search-box {
26      display: flex;
27      gap: 10px;
28      padding: 20px;
29      max-width: 500px;
30      margin: 0 auto;
31    }
```

```

32
33     .search-box input {
34         flex: 1;
35         padding: 10px;
36         font-size: 16px;
37         border: 1px solid #ddd;
38         border-radius: 4px;
39     }
40
41     .search-box button {
42         padding: 10px 20px;
43         background-color: #FF4500;
44         color: white;
45         border: none;
46         border-radius: 4px;
47         cursor: pointer;
48         font-weight: bold;
49     }
50
51     .search-box button:hover {
52         background-color: #FF5722;
53     }
54
55     /* Previous CSS... */
56 </style>
57
58 <script>
59     // Code goes here
60 </script>
61 </body>

```

## 5.2 Step 4.2: Your Task

Update JavaScript:

1. Create `loadPosts()` function that:

```

1 function loadPosts() {
2     // TODO:
3     // 1. Get subreddit from input field
4     // 2. Show loading div
5     // 3. Build URL: https://www.reddit.com/r/{subreddit}.json
6     // 4. Fetch from URL
7     // 5. Convert to JSON
8     // 6. Call displayPosts()
9     // 7. Hide loading div
10 }
11

```

2. Add error handling:

```
1 // If fetch fails, show error message
2
```

3. Auto-load on page load:

```
1 window.addEventListener('load', () => {
2     loadPosts(); // Load 'programming' by default
3 });
4
```

### 5.3 Step 4.3: Test It

1. Load page — should show programming posts
2. Type “learnprogramming” in input
3. Click Search — should update posts
4. Try other subreddits: “webdev”, “python”, etc.

#### Success criteria:

Input field visible

Search button works

Different subreddits load correctly

Loading message appears

Error handling works for invalid subreddits

## 6 Step 5: Custom Styling (Make It Beautiful)

**Goal:** Professional looking design with colors, fonts, layout.

**Time:** 30 minutes

**What to improve:**

- Header styling
- Color scheme
- Typography
- Spacing
- Hover effects
- Responsive design

### 6.1 Step 5.1: Your Task

Create professional CSS:

```
1 * {
2     margin: 0;
3     padding: 0;
4     box-sizing: border-box;
5 }
6
7 body {
8     font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
9     background-color: #0f1419;
10    color: #818384;
11    line-height: 1.6;
12 }
13
14 header {
15     background: linear-gradient(135deg, #FF4500 0%, #FF6B35 100%);
16     color: white;
17     padding: 30px;
18     text-align: center;
19     box-shadow: 0 2px 10px rgba(0,0,0,0.3);
20 }
21
22 header h1 {
23     font-size: 2.5rem;
24     margin: 0;
25 }
26
27 .search-box {
```



```

28     background-color: #1a1a1b;
29     border: 1px solid #343536;
30     border-radius: 8px;
31     padding: 15px;
32     margin: 20px auto;
33     max-width: 600px;
34     display: flex;
35     gap: 10px;
36 }
37
38 .search-box input {
39     flex: 1;
40     background-color: #030303;
41     color: #d7dad9;
42     border: 1px solid #343536;
43     padding: 12px;
44     border-radius: 4px;
45     font-size: 16px;
46 }
47
48 .search-box input::placeholder {
49     color: #818384;
50 }
51
52 .search-box button {
53     background-color: #FF4500;
54     color: white;
55     padding: 12px 25px;
56     border: none;
57     border-radius: 4px;
58     cursor: pointer;
59     font-weight: bold;
60     transition: 0.3s;
61 }
62
63 .search-box button:hover {
64     background-color: #FF6B35;
65     transform: translateY(-2px);
66     box-shadow: 0 4px 12px rgba(255, 69, 0, 0.3);
67 }
68
69 .posts-grid {
70     display: grid;
71     grid-template-columns: repeat(auto-fill, minmax(300px, 1fr));
72     gap: 20px;
73     padding: 20px;
74     max-width: 1400px;

```

```

75     margin: 0 auto;
76 }
77
78 .post-card {
79     background-color: #1a1a1b;
80     border: 1px solid #343536;
81     border-radius: 8px;
82     overflow: hidden;
83     transition: all 0.3s;
84     cursor: pointer;
85 }
86
87 .post-card:hover {
88     border-color: #FF4500;
89     box-shadow: 0 8px 20px rgba(255, 69, 0, 0.2);
90     transform: translateY(-4px);
91 }
92
93 .post-image {
94     width: 100%;
95     height: 180px;
96     object-fit: cover;
97     background-color: #343536;
98 }
99
100 .post-content {
101     padding: 15px;
102 }
103
104 .post-title {
105     font-size: 1rem;
106     font-weight: bold;
107     color: #d7dadac;
108     margin-bottom: 10px;
109     line-height: 1.4;
110     min-height: 40px;
111 }
112
113 .post-stats {
114     font-size: 0.85rem;
115     color: #818384;
116     display: flex;
117     gap: 15px;
118     flex-wrap: wrap;
119 }
120
121 footer {

```

```

122     text-align: center;
123     padding: 20px;
124     color: #818384;
125     border-top: 1px solid #343536;
126     margin-top: 40px;
127 }
128
129 @media (max-width: 768px) {
130     header h1 {
131         font-size: 1.8rem;
132     }
133
134     .posts-grid {
135         grid-template-columns: 1fr;
136     }
137
138     .search-box {
139         flex-direction: column;
140     }
141 }

```

## 6.2 Step 5.2: Update HTML Structure

Wrap post card content properly:

```

1 <!-- Update the post card HTML in displayPosts() -->
2 <div class="post-card">
3     
4     <div class="post-content">
5         <div class="post-title">{title}</div>
6         <div class="post-stats">
7             <span>        {ups}</span>
8             <span>        {num_comments}</span>
9             <span>by u/{author}</span>
10        </div>
11    </div>
12 </div>

```

## 6.3 Step 5.3: Test

Design looks professional

Colors are consistent

Hover effects work

Mobile responsive

No layout issues

## 7 Step 6 (Optional): Add Video Playback

**Goal:** Play videos when posts have video content.

**Time:** 15 minutes (optional)

**Challenge:** Some Reddit posts contain videos. Detect and play them.

### 7.1 Step 6.1: Detect Video Posts

Check if post has video:

```
1 // Check if post has video
2 if (data.media && data.media.reddit_video) {
3     // It's a video - use video player
4 } else if (data.thumbnail) {
5     // It's an image - use img tag
6 }
```

### 7.2 Step 6.2: Your Task

Update displayPosts to handle videos:

```
1 function displayPosts(posts) {
2     const container = document.getElementById('posts-container');
3     container.innerHTML = '';
4
5     posts.forEach(post => {
6         const data = post.data;
7
8         let mediaHTML = '';
9
10        // TODO: Check if video exists
11        // If yes: create <video> tag
12        // If no: create <img> tag
13
14        const cardHTML = `
15            <div class="post-card">
16                ${mediaHTML}
17                <div class="post-content">
18                    <div class="post-title">${data.title}</div>
19                    <div class="post-stats">
20                        ${data.ups} | ${data.
21num_comments}
22                    </div>
23                </div>
24            `;
25
```

```
26         container.innerHTML += cardHTML;
27     });
28 }
```

## 8 Final Project Summary

**You've built a complete Reddit Browser with:**

- HTML structure
- API integration (Reddit data)
- JavaScript to fetch and display
- Beautiful CSS styling
- Search functionality
- Responsive design
- Error handling
- (Optional) Video support

**Total Time:** 2-3 hours

**Core Concepts Learned:**

- Fetching data from APIs
- DOM manipulation
- Event handling
- CSS Grid layout
- Responsive design
- Error handling
- Full-stack thinking (frontend + API)
- Running local servers

## 9 How to Submit Your Work

### 9.1 Step 1: Create Submission Folder

Navigate to the submissions folder:

```
1 # Clone repository
2 git clone https://github.com/SMILE-LABS/tech-stack-fundamentals-
  cse24.git
3 cd tech-stack-fundamentals-cse24
4
5 # Go to Task 02 submissions
6 cd "Task 02/submissions"
7
8 # Create folder with your name
9 mkdir Your_First_Name_Your_Last_Name
10 cd Your_First_Name_Your_Last_Name
```

**Examples:**

Himath\_Jayasinghe/  
Anusha\_Perera/  
Kamal\_Silva/

### 9.2 Step 2: Save Your File

Save your index.html in your folder:

Task 02/submissions/Himath\_Jayasinghe/index.html

### 9.3 Step 3: Commit and Push

```
1 # Go back to repo root
2 cd ../../..
3
4 # Stage all changes
5 git add .
6
7 # Commit with message
8 git commit -m "Task 2: Reddit Browser - Your Name"
9
10 - Fetches Reddit API
11 - Displays posts in grid
12 - Search functionality
13 - Responsive design"
14
15 # Push to GitHub
16 git push
```



## 9.4 Step 4: Verify

Go to GitHub and check:

1. Your folder appears in Task 02/submissions/
2. Your index.html is inside
3. GitHub shows your commit

**Done!**

## 9.5 Submission Checklist

Before pushing, verify:

Folder name is correct: `First_Name_Last_Name`

File name is: `index.html`

Location: `Task 02/submissions/Your_Name/index.html`

Website loads in browser

Can search different subreddits

Posts display with images

No console errors (F12)

Mobile responsive (test on phone)

Git commit has clear message

## 10 Troubleshooting

### 10.1 CORS Error - Posts don't appear

- Error message: “Access to fetch has been blocked by CORS policy”
- Solution: You forgot to run local server (Step 0)!
- Do NOT double-click index.html
- Use Python, Live Server, or Node.js to run local server

### 10.2 Posts don't appear (after local server)

- Open browser console (F12)
- Check for red errors
- Make sure URL is correct
- Check Reddit API is responding

### 10.3 Images broken

- Some posts don't have images (normal)
- Add `onerror` handler to hide broken images
- Use placeholder image if needed

### 10.4 Search not working

- Check subreddit name spelling
- Try a popular one: “programming”, “webdev”, “python”
- Check console for errors

### 10.5 Cannot push to GitHub

```
1 # Check your remote
2 git remote -v
3
4 # If wrong, set correct remote
5 git remote set-url origin https://github.com/SMILE-LABS/tech-stack-
   fundamentals-cse24.git
6
7 # Then push
8 git push
```

## 11 Additional Resources

### 11.1 Helpful Links

- MDN Web Docs: <https://developer.mozilla.org>
- JavaScript.info: <https://javascript.info>
- Reddit API: <https://www.reddit.com/dev/api>
- CSS Grid: <https://css-tricks.com/snippets/css/complete-guide-grid/>

### 11.2 Try These Subreddits

- [r/programming](#)
- [r/webdev](#)
- [r/learnprogramming](#)
- [r/javascript](#)
- [r/python](#)
- [r/memes](#)
- [r/funny](#)
- [r/todayilearned](#)

## 12 What's Next?

After completing this project:

1. Learn React for more complex apps
2. Build backend with Node.js or Python
3. Deploy on Vercel or Netlify
4. Add more features (filtering, bookmarks, etc.)
5. Join open-source projects
6. Build portfolio projects

**You're officially a web developer!**

---

**Questions?** Check the cheatsheets in `/resources` folder.

**Deadline:** 26th December 2025

**Have fun coding!**