

Query:

We (an IT department for a K12 school district) have help information needing to be centrally organized into a simple, searchable system. I was thinking about a clever means of allowing users to search for help based on keywords. I was thinking they'd have a field they'd start typing in and as they type in keywords it would search a flat file that would consist of a URL (to a Google Doc or a webpage with the info they're looking for), a title for that information (to appear in their search results), and keywords for that entry. They would simply start typing and as they do the system would actively search for keywords and sort, say, the top 10 results by keyword hit count. I was hoping this could be relatively simple though, such as a HTML with inline JavaScript and perhaps the flat file as a separate, easily editable test file. I don't want to complicate this with a MySQL database.

ChatGPT Response:

You're on the right track. What you're describing is totally doable using just **HTML**, **JavaScript**, and a simple **flat file** (like a JSON or CSV file). There's no need for a full backend or database, and it'll be lightweight, fast, and easy to maintain.

Here's a breakdown of how you could build this:

Basic Architecture

1. **Flat File:** A JSON file containing all your help entries.
 2. **HTML Page:** Contains the search box and a div for displaying results.
 3. **JavaScript:** Loads the flat file, listens for input, does a keyword match, and displays ranked results.
-

Step 1: Example JSON Flat File (help_data.json)

```
[
  {
    "title": "How to Reset Student Passwords",
    "url": "https://docs.google.com/some-doc",
    "keywords": ["reset", "password", "student", "login"]
  },
  {
    "title": "Projector Not Displaying",
    "url": "https://docs.google.com/projector-issue",
    "keywords": ["projector", "display", "screen", "HDMI"]
  },
  {
    "title": "Connecting to WiFi",
    "url": "https://docs.google.com/wifi-guide",
    "keywords": ["wifi", "internet", "connection", "network"]
  }
]
```

Keep it editable in any text or code editor.

Step 2: Basic HTML Page (index.html)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>IT Help Search</title>
  <style>
    body { font-family: Arial, sans-serif; margin: 40px; }
    #results a { display: block; margin-bottom: 10px; }
    #results span { font-size: 0.9em; color: gray; }
  </style>
</head>
<body>

<h2>Search IT Help</h2>
<input type="text" id="searchBox" placeholder="Start typing..." autocomplete="off"
style="width: 300px; padding: 8px;">
<div id="results"></div>

<script>
  let helpData = [];

  // Load the JSON data
  fetch('help_data.json')
    .then(response => response.json())
    .then(data => helpData = data);

  document.getElementById('searchBox').addEventListener('input', function () {
    const query = this.value.toLowerCase().trim();
    const queryWords = query.split(/\s+/);

    if (!query) {
      document.getElementById('results').innerHTML = '';
      return;
    }

    const results = helpData.map(entry => {
      let hits = 0;
      queryWords.forEach(q => {
        if (entry.keywords.some(k => k.toLowerCase().includes(q))) {
          hits++;
        }
      });
      return { ...entry, hits };
    })
    .filter(entry => entry.hits > 0)
    .sort((a, b) => b.hits - a.hits)
    .slice(0, 10);

    const resultHtml = results.map(r =>
      `<a href="${r.url}" target="_blank">${r.title}<br><span>${r.keywords.join(',
' )}</span></a>`
    ).join('');

    document.getElementById('results').innerHTML = resultHtml || '<p>No results
found.</p>';
```

```
});  
</script>  
  
</body>  
</html>
```



How to Maintain It

- **Updating entries:** Just edit the `help_data.json` file. You could even host that file in Google Drive or GitHub if needed.
 - **Deployment:** Host this HTML on your internal site or share it via your school's intranet.
-



Bonus Ideas (Optional Enhancements)

- Highlight matched keywords.
 - Add category tags (e.g. "hardware", "software").
 - Add "last updated" dates in the entries.
 - Sync with Google Sheets + Apps Script → auto-generate the JSON file.
-

Would you like me to help you build this as a zip file or generate a full working version you can test immediately?