

Digimon Fan Website Guide

1 Introduction

- Overview: Build a Digimon Fan Portal to showcase Digimon, random generator, and fan submissions.
- Purpose: Educational project and fan engagement.

2 Website Features

- Homepage: Latest Digimon, navigation menu, search bar
- Digimon Showcase: Cards layout, filter by type/level
- Random Digimon Generator: Button shows a random Digimon
- Fan Submission Form: For logged-in users to submit new Digimon
- User Registration/Login: Secure access to fan features

3 Technologies Used

- HTML: Structure of pages
- CSS: Styling, responsiveness, animations
- JavaScript: Interactivity (search, random generator, form validation)

4 Step-by-Step Guide

1. Setup project folder (`index.html` , `style.css` , `script.js`)
2. HTML Structure: Header, Main section, Footer
3. CSS Styling: Grid layout, colors, hover effects, font styling
4. JavaScript Functionality: Dynamic display, search/filter, random generator, form validation

5 Sample Code

HTML (index.html)

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Digimon Fan Portal</title>
<link rel="stylesheet" href="style.css">
</head>
<body>
<header>
<h1>Digimon Fan Portal</h1>
<nav>
<ul>
<li><a href="#showcase">Showcase</a></li>
<li><a href="#random">Random Digimon</a></li>
```

```

<li><a href="#submit">Submit Digimon</a></li>
</ul>
</nav>
</header>
<main>
<section id="showcase">
<h2>Digimon Showcase</h2>
<div class="digimon-grid" id="digimonGrid"></div>
</section>
<section id="random">
<h2>Random Digimon Generator</h2>
<button id="randomBtn">Get Random Digimon</button>
<div id="randomDisplay"></div>
</section>
<section id="submit">
<h2>Submit Your Digimon</h2>
<form id="digimonForm">
<input type="text" id="name" placeholder="Name" required>
<input type="text" id="type" placeholder="Type" required>
<button type="submit">Submit</button>
</form>
</section>
</main>
<script src="script.js"></script>
</body>
</html>

```

CSS (style.css)

```

body { font-family: Arial, sans-serif; margin: 0; padding: 0; }
header { background-color: #ffcc00; padding: 20px; text-align: center; }
nav ul { list-style: none; display: flex; justify-content: center; gap: 20px; }
.digimon-grid { display: grid; grid-template-columns: repeat(auto-fit,
minmax(150px, 1fr)); gap: 20px; padding: 20px; }
.digimon-card { background: #fff; padding: 10px; text-align: center; border-
radius: 10px; box-shadow: 0 0 5px rgba(0,0,0,0.2); }

```

JavaScript (script.js)

```

const digimons = [
  { name: 'Agumon', type: 'Reptile' },
  { name: 'Gabumon', type: 'Reptile' },
  { name: 'Patamon', type: 'Mammal' }
];

```

```

const digimonGrid = document.getElementById('digimonGrid');
digimons.forEach(d => {
const div = document.createElement('div');
div.className = 'digimon-card';
div.innerHTML = `

### ${d.name}</h3><p>${d.type}</p>`; digimonGrid.appendChild(div); }); document.getElementById('randomBtn').addEventListener('click', () => { const rand = digimons[Math.floor(Math.random() * digimons.length)]; document.getElementById('randomDisplay').innerHTML = `${rand.name}</h3><p>${rand.type}</p>`; }); document.getElementById('digimonForm').addEventListener('submit', e => { e.preventDefault(); const name = document.getElementById('name').value; const type = document.getElementById('type').value; digimons.push({ name, type }); alert(`${name} added! Refresh the page to see it in showcase.`); });


```

6 Additional Info

- Responsive and mobile-friendly
- Expandable with images, evolution chains, and filters
- Optional: Use JSON file or database for storing Digimon data