

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>English Dictionary</title>
  <script src="https://cdn.tailwindcss.com"></script>
</head>
<body class="min-h-screen bg-gray-100 flex items-center justify-center p-4">
  <div class="w-full max-w-3xl bg-white shadow-md rounded-lg">
    <div class="p-4 border-b">
      <h1 class="text-2xl font-semibold text-blue-600">English Dictionary</h1>
      <p class="text-green-600 font-semibold pt-2">Search for word meanings, synonyms, and
examples</p>
    </div>
    <div class="p-4">
      <!-- Input and Button -->
      <div class="flex space-x-2 mb-4">
        <input
          id="wordInput"
          type="text"
          placeholder="Enter a word"
          class="border border-gray-300 rounded px-3 py-2 w-full"
        />
        <button
          id="searchButton"
          class="bg-blue-600 text-white px-4 py-2 rounded hover:bg-blue-700"
        >
          Search
        </button>
      </div>
      <!-- Error Message -->
      <p id="errorMessage" class="text-red-500 mb-4 hidden"></p>
      <!-- Word Data -->
      <div id="wordData" class="hidden">
        <h2 class="text-2xl text-green-600 font-semibold mb-2" id="wordTitle"></h2>
        <p class="text-gray-600 mb-4" id="phonetic"></p>
        <div id="meanings"></div>
      </div>
    </div>
  </div>

  <script>
    document.getElementById('searchButton').addEventListener('click', searchWord);

    async function searchWord() {
      const wordInput = document.getElementById('wordInput');
      const errorMessage = document.getElementById('errorMessage');
      const wordDataDiv = document.getElementById('wordData');
      const wordTitle = document.getElementById('wordTitle');
      const phonetic = document.getElementById('phonetic');

```

```

const meanings = document.getElementById('meanings');

const word = wordInput.value.trim();

// Reset UI
errorMessage.classList.add('hidden');
wordDataDiv.classList.add('hidden');
wordTitle.textContent = '';
phonetic.textContent = '';
meanings.innerHTML = '';

if (!word) return;

try {
  const response = await fetch(`https://api.dictionaryapi.dev/api/v2/entries/en/${word}`);
  const data = await response.json();

  if (!response.ok || !data[0]) {
    errorMessage.textContent = 'Word not found. Please try another word.';
    errorMessage.classList.remove('hidden');
    return;
  }

  // Display Word Data
  const firstEntry = data[0];
  wordTitle.textContent = firstEntry.word;
  phonetic.textContent = firstEntry.phonetic || '';

  firstEntry.meanings.forEach(meaning => {
    const meaningDiv = document.createElement('div');
    meaningDiv.classList.add('mb-4');

    const partOfSpeech = document.createElement('h3');
    partOfSpeech.textContent = meaning.partOfSpeech;
    partOfSpeech.classList.add('text-blue-600', 'font-semibold', 'mb-2');
    meaningDiv.appendChild(partOfSpeech);

    // Definitions
    const defList = document.createElement('ul');
    defList.classList.add('list-disc', 'pl-5', 'space-y-2');
    meaning.definitions.forEach(def => {
      const listItem = document.createElement('li');
      listItem.textContent = def.definition;
      defList.appendChild(listItem);
    });
    meaningDiv.appendChild(defList);

    // Synonyms
    if (meaning.synonyms && meaning.synonyms.length > 0) {
      const synonyms = document.createElement('p');
      synonyms.textContent = 'Synonyms: ' + meaning.synonyms.slice(0, 5).join(', ');
    }
  });
}

```

```
        synonyms.classList.add('text-green-600', 'mt-2');
        meaningDiv.appendChild(synonyms);
    }

    meanings.appendChild(meaningDiv);
});

wordDataDiv.classList.remove('hidden');
} catch (err) {
    errorMessage.textContent = 'An error occurred. Please try again later.';
    errorMessage.classList.remove('hidden');
}
}
</script>
</body>
</html>
```