**Dictionary Web Application – Code Explanation**

**1. HTML Structure**

html

<!DOCTYPE html>

<html lang="en">

* Declares the document type as **HTML5**.
* Specifies the language as **English** (lang="en"), helping search engines and screen readers.

**2. Head Section (Metadata & Styling)**

html

<head>

<meta charset="UTF-8">

* Defines **UTF-8 character encoding**, ensuring proper display of text in various languages.

html

<meta name="viewport" content="width=device-width, initial-scale=1.0">

* Ensures **responsiveness**, adjusting content layout on different screen sizes.

html

<title>Dictionary Search</title>

* Sets the **page title**, displayed on the browser tab.

html

<script src="https://cdn.tailwindcss.com"></script>

* Imports **Tailwind CSS**, a utility-based styling framework.

**3. CSS Styling**

css

@import url('https://fonts.googleapis.com/css2?family=Nunito:wght@200;300;400;600;700&display=swap');

* Imports **Nunito** font for a modern, friendly look.

css

body {

background-color: #D6E4F0;

font-family: 'Nunito', sans-serif;

}

* Sets **background color** (#D6E4F0 = soft pastel blue).
* Applies **Nunito** font to the page.

css

@keyframes bounce {

0%, 100% {transform: translateY(0);}

50% {transform: translateY(-10px);}

}

* Defines a **bouncing animation** for interactive elements.

css

.search-container {

background-color: rgb(182, 132, 226);

padding: 20px;

border-radius: 24px;

}

* Sets the **search box** background color (**pastel purple**).
* Adds **padding** and **rounded edges** for smooth design.

**4. Main Body Structure**

html

<body class="bg-blue-100 min-h-screen flex flex-col items-center justify-center">

* Defines a **blue background** (bg-blue-100).
* Uses **flexbox** for **vertical centering**.

html

<div class="w-full max-w-lg bg-white p-6 rounded-3xl shadow-md">

* Creates a **responsive white card** (max-w-lg ensures optimal width).
* Adds a **shadow effect** (shadow-md) for depth.

**5. Search Form**

html

<form method="GET" action="" class="mb-4 grid grid-cols-[1fr\_auto] gap-3 items-center">

* Uses **GET method** to send query parameters.
* Applies a **grid layout** (grid-cols-[1fr\_auto]) for a neat button & input alignment.

html

<input type="text" name="word" placeholder="Enter a word"

class="border w-full p-3 rounded-3xl focus:outline-none focus:ring-2 focus:ring-purple-500 input-field text-lg"

required />

* Defines an **input field** for word search.
* Adds **rounded edges** and a **purple focus outline**.

html

<button type="submit" class="bg-[#C3B1E1] text-white py-3 px-6 rounded-3xl text-lg hover:bg-[#C3B1E1]">

Search

</button>

* Creates a **search button** with a **pastel purple background**.

**6. PHP Logic (Fetching Dictionary Data)**

php

if (isset($\_GET['word'])) {

$word = htmlspecialchars($\_GET['word']);

* Checks if a **word** is submitted.
* Uses htmlspecialchars() to **sanitize** input and prevent security issues.

php

$url = "https://api.dictionaryapi.dev/api/v2/entries/en/$word";

$response = file\_get\_contents($url);

* Constructs the **API URL** dynamically using the **searched word**.
* Fetches **dictionary data** using file\_get\_contents().

php

if ($response === FALSE) {

echo "<p class='text-red-500 mt-4'>Error fetching data from API. Please try again later.</p>";

exit();

}

* Handles **API errors**, displaying a **red message** if data fetching fails.

php

$data = json\_decode($response, true);

if ($data === NULL) {

echo "<p class='text-red-500 mt-4'>Invalid response received from API.</p>";

exit();

}

* Converts JSON **API response** into an **associative array**.
* Handles **invalid responses**.

php

foreach ($data as $entry) {

echo "<h2 class='text-2xl font-bold text-blue-800'>$word <span class='text-gray-700'>(" . $entry['meanings'][0]['partOfSpeech'] . ")</span> = <span class='text-black font-bold'>" . $entry['meanings'][0]['definitions'][0]['definition'] . "</span></h2>";

* Loops through the **API response** to display the **word definition**.
* Shows **part of speech** and **meaning**.

php

if (!empty($entry['phonetics'][0]['text'])) {

echo "<p><strong>Pronunciation:</strong> " . $entry['phonetics'][0]['text'] . "</p>";

}

* Displays **phonetic transcription** if available.

php

if (!empty($entry['phonetics'][0]['audio'])) {

echo "<audio controls class='mt-2'><source src='" . $entry['phonetics'][0]['audio'] . "' type='audio/mpeg'>Your browser does not support the audio element.</audio>";

}

* Adds an **audio pronunciation** feature for clarity.

**7. Contact Button(?)**

html

<a href="mailto:heliabrs40@gmail.com"

class="fixed bottom-16 right-8 bg-blue-500 text-white rounded-3xl p-3 shadow-lg hover:bg-blue-600 transition bounce-animation text-md"

title="Email Us">

* Adds a **floating contact button** with an **email link**.
* Includes a **hover animation** (bounce-animation).