**A CAPSTONE PROJECT REPORT ON**

**Word Nest**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**BY**

**T.Sudheshna Devi(23501A05I1), P.Nandini(23501A05D5), N.Bhagya Sri(24505A0513), Y.Nihar(23501A05J4), Y.Bhashitha(23501A05J2), T.Anu(23501A05H4)**

**Under the Guidance of**

**Dr. U Sirisha, Ph.D.**

**Assistant Professor**



**PRASAD V POTLURI SIDDHARTHA INSTITUTE OF TECHNOLOGY**

(Permanently affiliated to JNTU Kakinada, Approved by AICTE)

(An NBA & NAAC A+ accredited and ISO 9001:2015 Certified Institution)

**Kanuru, Vijayawada – 520007**

**2024-25**

**PRASAD V POTLURI SIDDHARTHA INSTITUTE OF TECHNOLOGY**

(Permanently affiliated to JNTU Kakinada, Approved by AICTE)

(An NBA & NAAC A+ accredited and ISO 9001:2015 certified institution)

**Kanuru, Vijayawada – 520007**



# CERTIFICATE

This is to certify that the Capstone project report title  **“Word Nest”**  is the bonafied work of  **T.Sudheshna Devi(23501A05I1), P.Nandini(23501A05D5), N.Bhagya Sri(24505A0513), Y.Nihar(23501A05J4), Y.Bhashitha(23501A05J2), T.Anu(23501A05H4)** in partial fulfilment of completing the Academic project in Full Stack Development-I during the academic year 2024-25.

**Signature of the Course Coordinator Signature of the HOD**

## **INDEX**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Content** | **Page No. (s)** |
| 1 | Abstract | 1 |
| 2 | Introduction | 2 |
| 3 | Objectives and Scope of the Project | 3 |
| 4 | Software used - Explanation | 4 |
| 5 | Proposed model | 5 |
| 6 | Sample Code | 6-16 |
| 7 | Result/Output Screen shots | 17-20 |
| 8 | Conclusion | 21 |
| 9 | References (web site URLs) | 22 |

**1. Abstract**

The *Dictionary Application* project aims to develop a user-friendly and efficient software tool that allows users to search for word meanings, synonyms, antonyms, and usage examples. The primary objective of this project is to provide an accessible and interactive digital dictionary that can enhance vocabulary learning and language proficiency, particularly for students and language learners.

This application utilizes a structured database or API to store and retrieve word definitions and related information. The backend logic is designed to ensure fast query processing, while the frontend offers a clean and intuitive interface. Additional features such as voice pronunciation, word-of-the-day, and offline access may also be incorporated to improve the overall user experience.

The project was implemented using technologies such as Python (or HTML/CSS/JavaScript for web-based versions), and tested to ensure stability, accuracy, and ease of use. The Dictionary App stands out as a practical and educational tool, with the potential for future enhancements like multi-language support and AI-based recommendations.

**2. Introduction**

In today's digital world, the need for quick and easy access to information has become essential. One of the most common requirements, especially for students, writers, and language learners, is the ability to look up the meaning of unfamiliar words. Traditionally, printed dictionaries were used, but they are bulky, static, and often lack real-time updates or multimedia support. Even though digital dictionaries exist, many of them are either paid, overloaded with ads, or lack user-friendly design and extra features like example usage, synonyms, or pronunciation. There is a need for a simple, lightweight, and effective dictionary application that offers a clean interface, reliable results, and optional advanced features. Word Nest emerges as a modern, web-based dictionary designed to meet the evolving needs of users who seek an intuitive, interactive, and comprehensive language reference tool.

**3. Objectives and scope of the project**

**Objective of the Project**

The main objective of this project is to design and develop a dictionary application that provides users with:

* Accurate word meanings and definitions.
* Synonyms and antonyms for better vocabulary understanding.
* Example sentences to understand contextual usage.
* Additional features like pronunciation (text-to-speech) and word-of-the-day (if included).
* A simple and responsive user interface for ease of access.

The goal is to make the application educational, easy to navigate, and accessible to all users, especially students and non-native speakers.

**Scope and Relevance**

This project is relevant to both academic and practical real-world use. It demonstrates core programming skills, data handling, and UI design—all important areas in Computer Science. The dictionary app can be expanded with features like offline access, multi-language support, and voice input, making it scalable and future-ready.

In a broader context, the application can support e-learning, assist in communication, and serve as a personal vocabulary-building tool. The project showcases how simple technologies can be used to build useful educational software with real-life applications.

**4. Software used-Explanation**

**Programming Languages**

* **HTML**: Used to create the structure and content of the web pages such as login, signup, and dictionary screens.
* **CSS**: Used to design and style the user interface with custom layouts, buttons, and gradients for a modern look.
* **JavaScript**: Used for handling dynamic interactions, form validation, page navigation, and API requests.

**Frameworks / Libraries**

* **None explicitly used in the code**, but:
  + **Web APIs** such as fetch() are used for calling the external dictionary API.
  + **Optional**: The React logo is displayed on the welcome and thank-you screens for design purposes, but React is not implemented in the code.

**Databases**

* **Local Storage (Web Storage API)**:
  + Used to temporarily store user signup information such as username, password, email, and phone number.
  + Data persists across sessions within the same browser.

**Tools (IDEs, Platforms, APIs)**

* **VS Code (or any text editor)**: Used for writing and managing HTML, CSS, and JavaScript code.
* **Web Browser**: Acts as the runtime environment where the application is tested and executed.
* [**Free Dictionary API**](https://dictionaryapi.dev/): Used to fetch real-time definitions, phonetics, examples, synonyms, and antonyms for searched words.

**5. Proposed Model**

The proposed model for Word Nest focuses on creating a robust, user-friendly, and responsive online dictionary platform. The model consists of two main components: the functional features and the user interface (UI) design. Together, these aim to provide a smooth, informative, and engaging experience for users of all age groups and backgrounds.

**1. Key Features:**

Real-Time Search:

* A predictive search bar that offers real-time suggestions based on user input.
* Supports spelling corrections and partial word matches.

Word Details Page:

* Displays word definitions, part of speech, and pronunciation (with audio).
* Shows synonyms, antonyms, usage in sentences, and origin/etymology.
* Visual icons for pronunciation, audio playback, and language roots.

Responsive & Accessible Design:

* Fully compatible with mobile, tablet, and desktop devices.
* Adheres to accessibility standards (WAI-ARIA), making it usable by people with disabilities.

**2. User Interface Design**

Homepage:

* Clean layout with a large central search bar.
* Quick access to Word of the Day, trending words, and tips.

Word Result Layout:

* Card-style display with collapsible sections for meanings, examples, and synonyms.
* Icons and typography are used for clarity and ease of reading.

**6. Sample Code**

* **Html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

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

<title>Happy Nest Dictionary</title>

<link rel="stylesheet" href="style.css" />

</head>

<body>

<!-- Auth Screen -->

<div class="frame" id="auth-screen">

<h1>Welcome to <span class="highlight">Happy Nest</span></h1>

<p>Login or Sign Up to continue</p>

<button class="next-btn" onclick="showLogin()">Login</button>

<button class="next-btn" onclick="showSignup()">Sign Up</button>

</div>

<!-- Login Screen -->

<div class="frame hidden" id="login-screen">

<h2>Login</h2>

<input type="text" id="login-username" placeholder="Username" />

<input type="password" id="login-password" placeholder="Password" />

<div class="btn-row">

<button class="next-btn" onclick="authScreenBack()">Back</button>

<button class="next-btn" onclick="login()">Login</button>

</div>

</div>

<!-- Signup Screen -->

<div class="frame hidden" id="signup-screen">

<h2>Sign Up</h2>

<input type="text" id="signup-username" placeholder="Choose Username" />

<input type="email" id="signup-email" placeholder="Enter Gmail" />

<input type="tel" id="signup-phone" placeholder="Enter Phone Number" />

<input type="password" id="signup-password" placeholder="Choose Password" />

<input type="password" id="signup-repassword" placeholder="Re-enter Password" />

<div class="btn-row">

<button class="next-btn" onclick="authScreenBack()">Back</button>

<button class="next-btn" onclick="signup()">Sign Up</button>

</div>

</div>

<!-- Welcome Screen -->

<div class="frame hidden" id="welcome-screen">

<img src="https://upload.wikimedia.org/wikipedia/commons/thumb/a/a7/React-icon.svg/512px-React-icon.svg.png" alt="Logo" class="logo" />

<h1>Hello! Welcome to <span class="highlight">Happy Nest</span></h1>

<button class="next-btn" onclick="showDictionary()">Next</button>

</div>

<!-- Dictionary App Screen -->

<div class="frame hidden" id="dictionary-screen">

<div class="container">

<div class="top-bar">

<button class="logout-btn" onclick="logout()">Logout</button>

<button class="next-btn" onclick="showTeam()">Our Team</button>

</div>

<h1>Dictionary</h1>

<form id="search-box" class="search-box">

<input type="text" id="search-input" placeholder="Search a word" />

<button type="submit">🔍</button>

</form>

<p class="errTxt"></p>

<div class="word-details">

<div class="word-info">

<div class="word-name">

<h2 id="word-txt">play</h2>

<p>

<span id="type-txt">noun </span>

<span id="phonetic-txt">/pleɪ/</span>

</p>

</div>

<button type="button" id="sound-btn">🔊</button>

</div>

<p class="definition" id="definition-txt">Activity for amusement only.</p>

<div class="blockquote" id="example-elem">

<h3>Examples</h3>

<p>Children learn through play.</p>

</div>

<div class="blockquote" id="synonyms-elem">

<h3>Synonyms</h3>

<p>play</p>

</div>

<div class="blockquote" id="antonyms-elem">

<h3>Antonyms</h3>

<p>some word</p>

</div>

</div>

</div>

</div>

<!-- Our Team Screen -->

<div class="frame hidden" id="team-screen">

<h2>Our Team</h2>

<ul style="text-align: left; list-style-type: disc; margin-left: 20px;">

<li><span class="highlight">23501A05I1</span> (T. Sudheshna Devi)</li>

<li><span class="highlight">23501A05D5</span> (P. Nandini)</li>

<li><span class="highlight">24505A0513</span> (N. Bhagya Sri)</li>

<li><span class="highlight">23501A05J4</span> (Y. Nihar)</li>

<li><span class="highlight">23501A05J2</span> (Y. Bhashitha)</li>

<li><span class="highlight">23501A05H4</span> (T. Anu)</li>

</ul>

<button class="next-btn" onclick="showDictionary()">Back</button>

</div>

<!-- Thank You Screen -->

<div class="frame hidden" id="thankyou-screen">

<img src="https://upload.wikimedia.org/wikipedia/commons/thumb/a/a7/React-icon.svg/512px-React-icon.svg.png" alt="Logo" class="logo" />

<h1>Thank You Very Much <br /> For Choosing <span class="highlight">Happy Nest</span>!</h1>

</div>

<script>

const authScreen = document.getElementById("auth-screen");

const loginScreen = document.getElementById("login-screen");

const signupScreen = document.getElementById("signup-screen");

const welcomeScreen = document.getElementById("welcome-screen");

const dictionaryScreen = document.getElementById("dictionary-screen");

const thankYouScreen = document.getElementById("thankyou-screen");

const teamScreen = document.getElementById("team-screen");

const signupUsername = document.getElementById("signup-username");

const signupPassword = document.getElementById("signup-password");

const signupRepassword = document.getElementById("signup-repassword");

const signupEmail = document.getElementById("signup-email");

const signupPhone = document.getElementById("signup-phone");

const loginUsername = document.getElementById("login-username");

const loginPassword = document.getElementById("login-password");

const searchBox = document.getElementById("search-box");

const searchInput = document.getElementById("search-input");

const wordTxt = document.getElementById("word-txt");

const typeTxt = document.getElementById("type-txt");

const phoneticTxt = document.getElementById("phonetic-txt");

const soundBtn = document.getElementById("sound-btn");

const definitionTxt = document.getElementById("definition-txt");

const exampleElem = document.getElementById("example-elem");

const synonymsElem = document.getElementById("synonyms-elem");

const antonymsElem = document.getElementById("antonyms-elem");

const wordDetailsElem = document.querySelector(".word-details");

const errTxt = document.querySelector(".errTxt");

const audio = new Audio();

function showLogin() {

authScreen.classList.add("hidden");

loginScreen.classList.remove("hidden");

}

function showSignup() {

authScreen.classList.add("hidden");

signupScreen.classList.remove("hidden");

}

function authScreenBack() {

loginScreen.classList.add("hidden");

signupScreen.classList.add("hidden");

authScreen.classList.remove("hidden");

}

function signup() {

const username = signupUsername.value;

const password = signupPassword.value;

const repassword = signupRepassword.value;

const email = signupEmail.value;

const phone = signupPhone.value;

if (username && password && repassword && email && phone) {

if (password !== repassword) {

alert("Passwords do not match.");

return;

}

localStorage.setItem("happyNestUser", JSON.stringify({ username, password, email, phone }));

alert("Signup successful! Please login.");

signupScreen.classList.add("hidden");

authScreen.classList.remove("hidden");

} else {

alert("Please fill all fields.");

}

}

function login() {

const username = loginUsername.value;

const password = loginPassword.value;

const storedUser = JSON.parse(localStorage.getItem("happyNestUser"));

if (storedUser && storedUser.username === username && storedUser.password === password) {

loginScreen.classList.add("hidden");

welcomeScreen.classList.remove("hidden");

} else {

alert("Invalid credentials. Try again.");

}

}

function showDictionary() {

welcomeScreen.classList.add("hidden");

teamScreen.classList.add("hidden");

dictionaryScreen.classList.remove("hidden");

}

function logout() {

dictionaryScreen.classList.add("hidden");

thankYouScreen.classList.remove("hidden");

}

function showTeam() {

dictionaryScreen.classList.add("hidden");

teamScreen.classList.remove("hidden");

}

async function getWordDetails(word) {

const res = await fetch(https://api.dictionaryapi.dev/api/v2/entries/en/${word});

const data = await res.json();

const wordData = data[0];

const phonetics = wordData.phonetics || [];

let phoneticText = "", phoneticAudio = "";

for (const phonetic of phonetics) {

if (phonetic.text && !phoneticText) phoneticText = phonetic.text;

if (phonetic.audio && !phoneticAudio) phoneticAudio = phonetic.audio;

if (phoneticText && phoneticAudio) break;

}

const meaning = wordData.meanings[0];

return {

word: word.toLowerCase(),

phonetic: { text: phoneticText, audio: phoneticAudio },

speechPart: meaning.partOfSpeech,

definition: meaning.definitions[0].definition,

synonyms: meaning.synonyms,

antonyms: meaning.antonyms,

example: meaning.definitions[0].example || ""

};

}

searchBox.addEventListener("submit", async e => {

e.preventDefault();

if (searchInput.value.trim() === "") {

errTxt.textContent = "Please Enter a Word";

} else {

wordDetailsElem.classList.remove("active");

try {

errTxt.textContent = "";

const wordDetails = await getWordDetails(searchInput.value);

wordTxt.textContent = wordDetails.word;

typeTxt.textContent = wordDetails.speechPart;

phoneticTxt.textContent = wordDetails.phonetic.text;

audio.src = wordDetails.phonetic.audio;

definitionTxt.textContent = wordDetails.definition;

exampleElem.querySelector("p").textContent = wordDetails.example;

synonymsElem.querySelector("p").textContent = wordDetails.synonyms.join(", ");

antonymsElem.querySelector("p").textContent = wordDetails.antonyms.join(", ");

exampleElem.style.display = wordDetails.example === "" ? "none" : "block";

synonymsElem.style.display = wordDetails.synonyms.length === 0 ? "none" : "block";

antonymsElem.style.display = wordDetails.antonyms.length === 0 ? "none" : "block";

wordDetailsElem.classList.add("active");

} catch {

errTxt.textContent = "Word Not Found";

}

}

});

soundBtn.addEventListener("click", () => {

audio.play();

});

</script>

</body>

</html>

* **Css**

body {

margin: 0;

font-family: Arial, sans-serif;

background: linear-gradient(135deg, #cce5ff, #e6f0ff);

background-image: url('https://www.shutterstock.com/shutterstock/videos/3576520755/thumb/1.jpg?ip=x480');

background-repeat: no-repeat;

background-position: center center;

background-attachment: fixed;

background-size: cover;

display: flex;

justify-content: center;

align-items: center;

min-height: 100vh;

}

.frame {

background: rgba(255, 255, 255, 5.0);

padding: 40px;

border-radius: 20px;

box-shadow: 0 10px 30px rgba(0, 0, 0, 0.2);

width: 90%;

max-width: 500px;

text-align: center;

backdrop-filter: blur(10px);

}

.hidden {

display: none;

}

.highlight {

color: #0099ff;

}

input {

display: block;

width: 90%;

padding: 12px;

margin: 15px auto;

border: 1px solid #ccc;

border-radius: 8px;

}

.next-btn, .back-btn, .logout-btn {

padding: 10px 20px;

margin: 10px 5px;

border: none;

background: #0099ff;

color: white;

border-radius: 8px;

cursor: pointer;

}

.logout-btn {

float: right;

margin: 0 0 20px 0;

}

.logo {

width: 80px;

margin-bottom: 20px;

}

.word-details {

text-align: left;

margin-top: 20px;

}

.blockquote {

background: #f0f8ff;

border-left: 4px solid #0099ff;

padding: 10px;

margin-top: 10px;

}

.search-box {

display: flex;

gap: 10px;

justify-content: center;

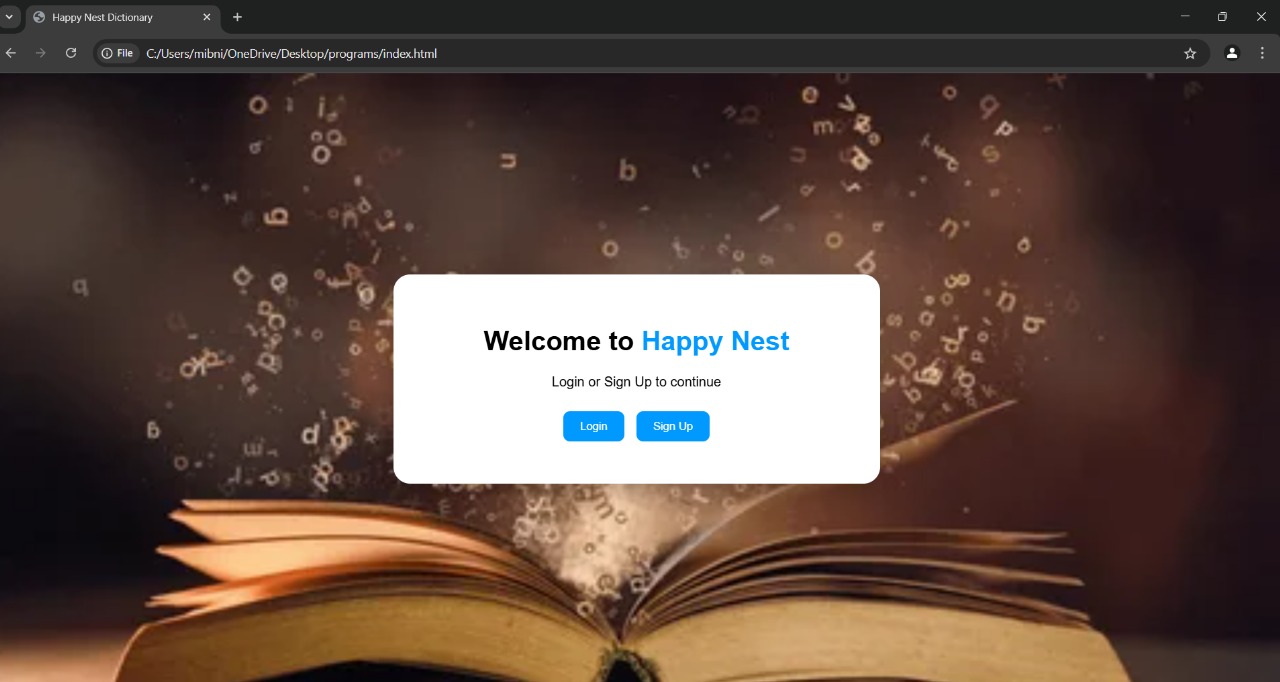
}

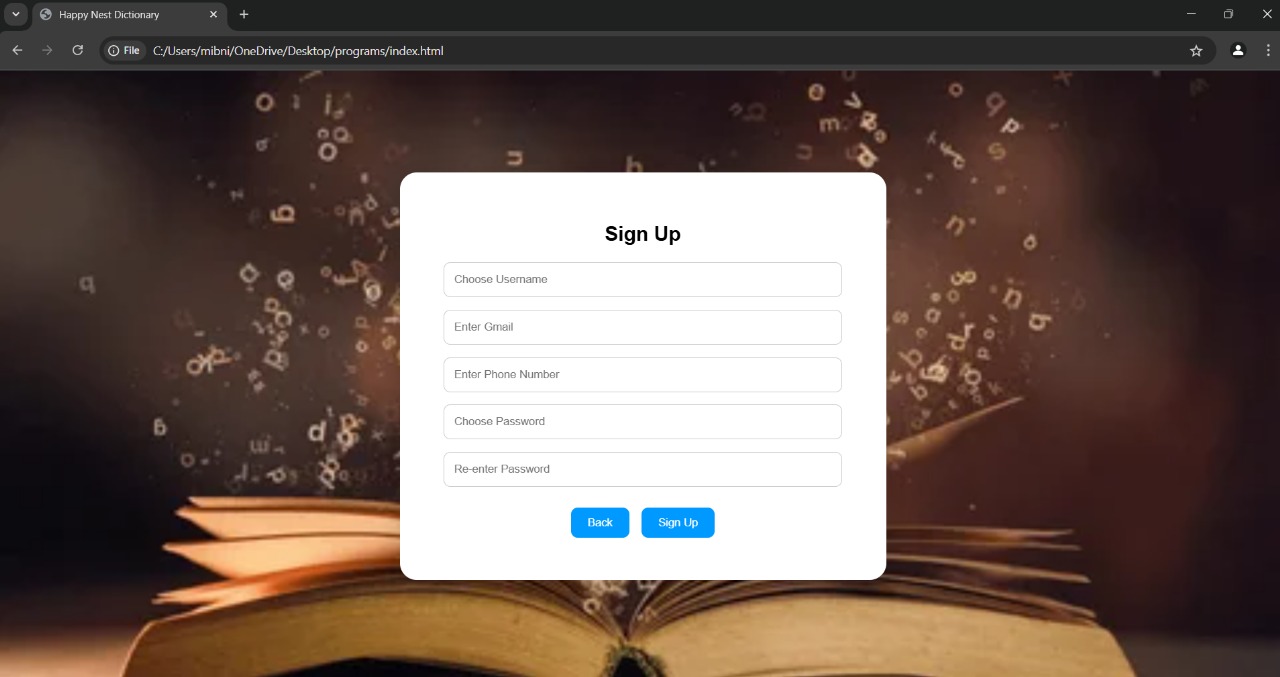
.search-box input {

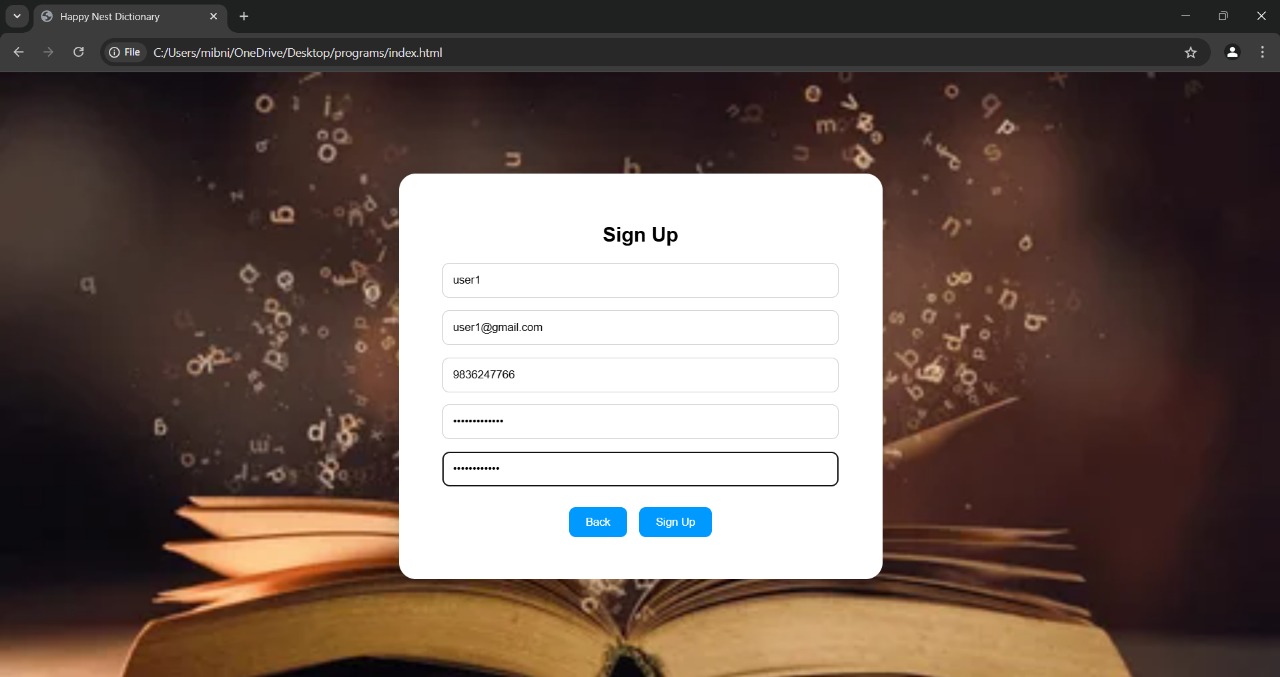
flex: 1;

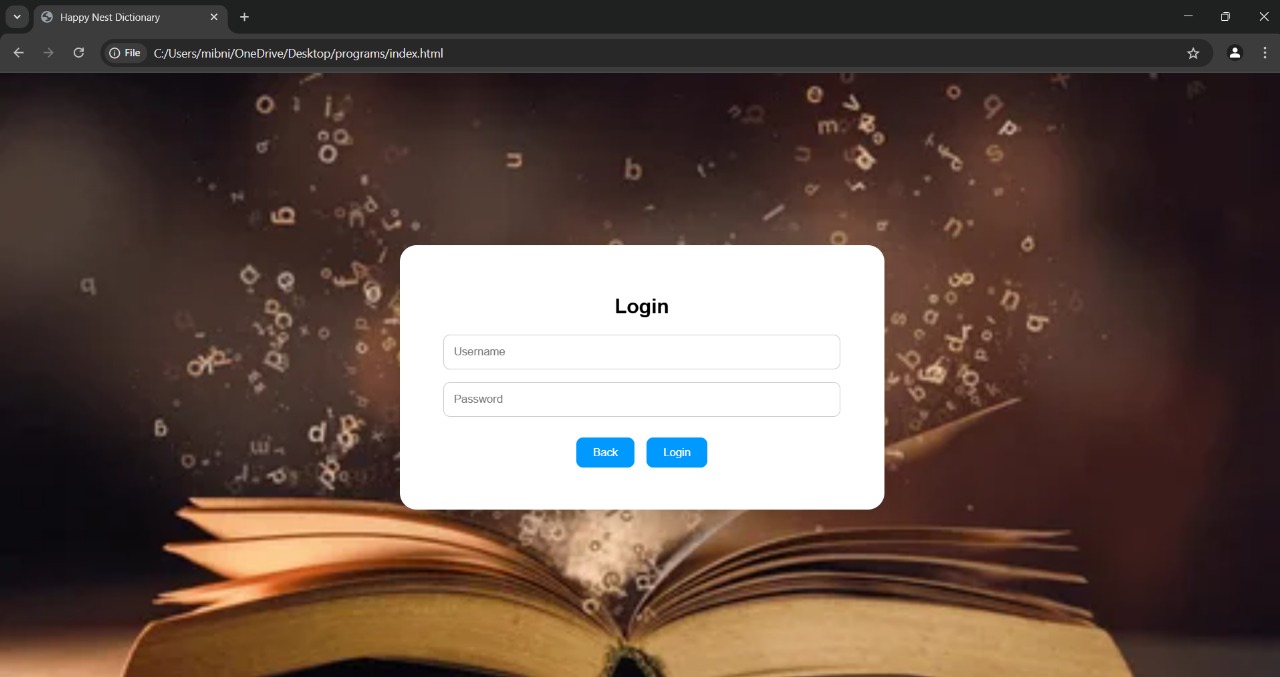
}

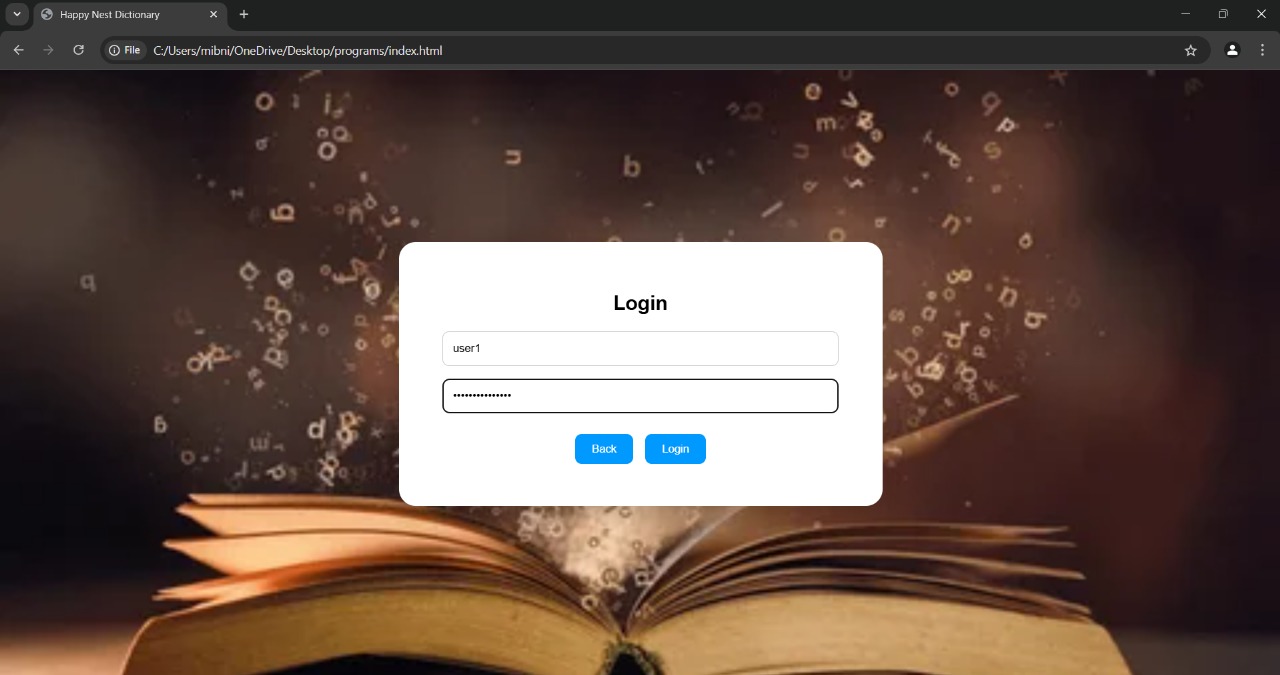
**7.Result/Output Screenshots**

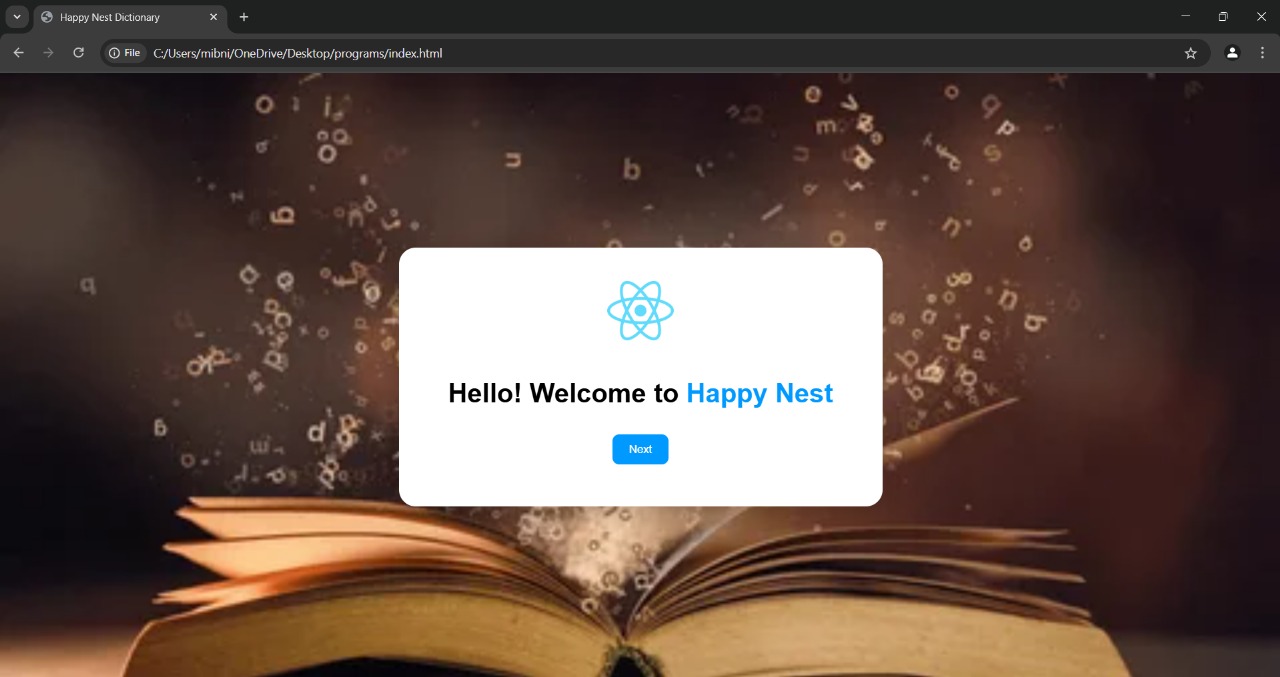
****

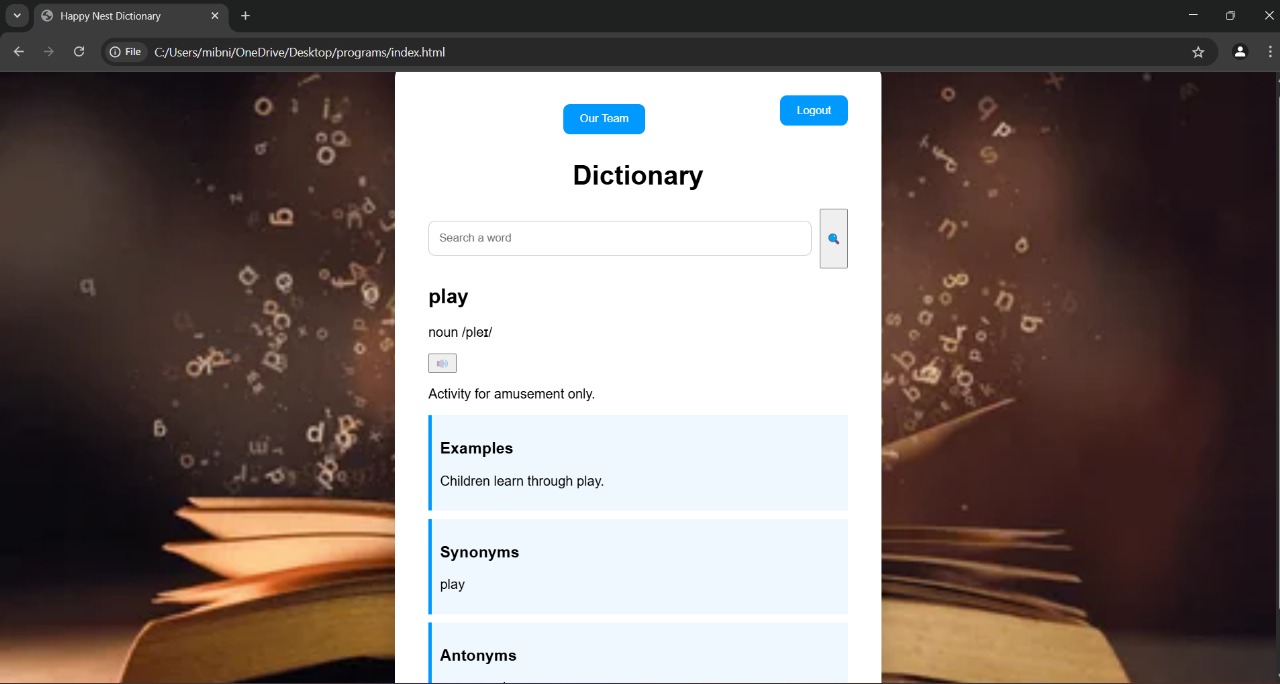
****

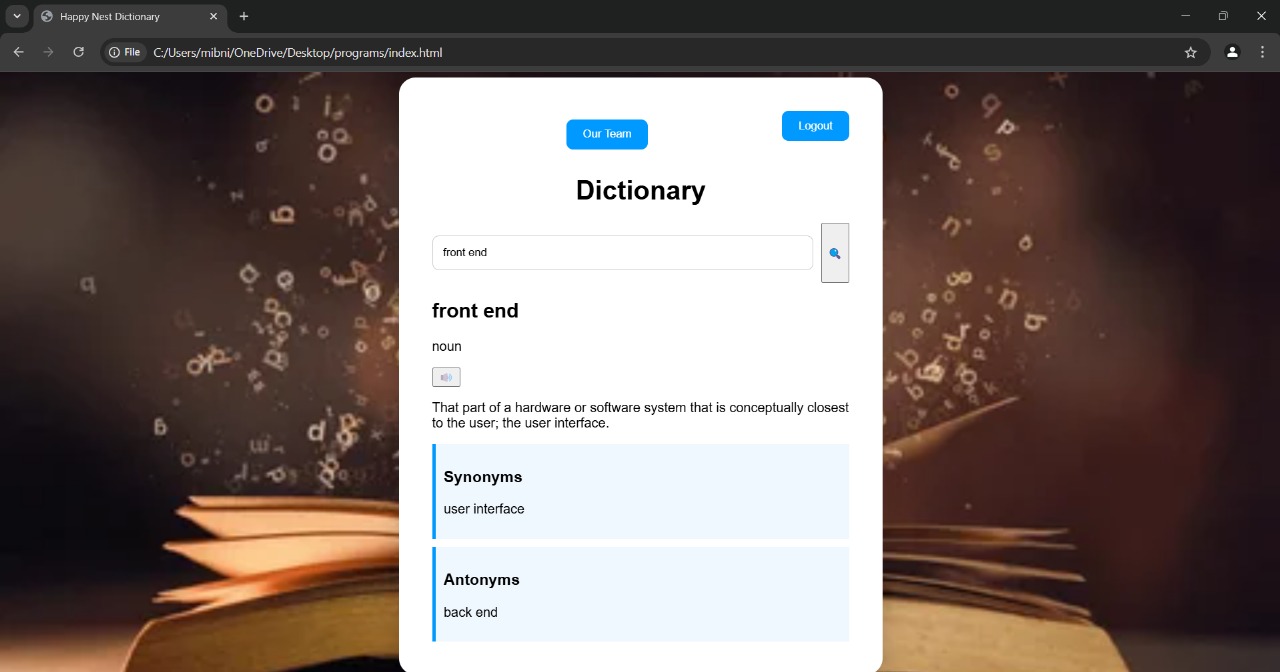
****

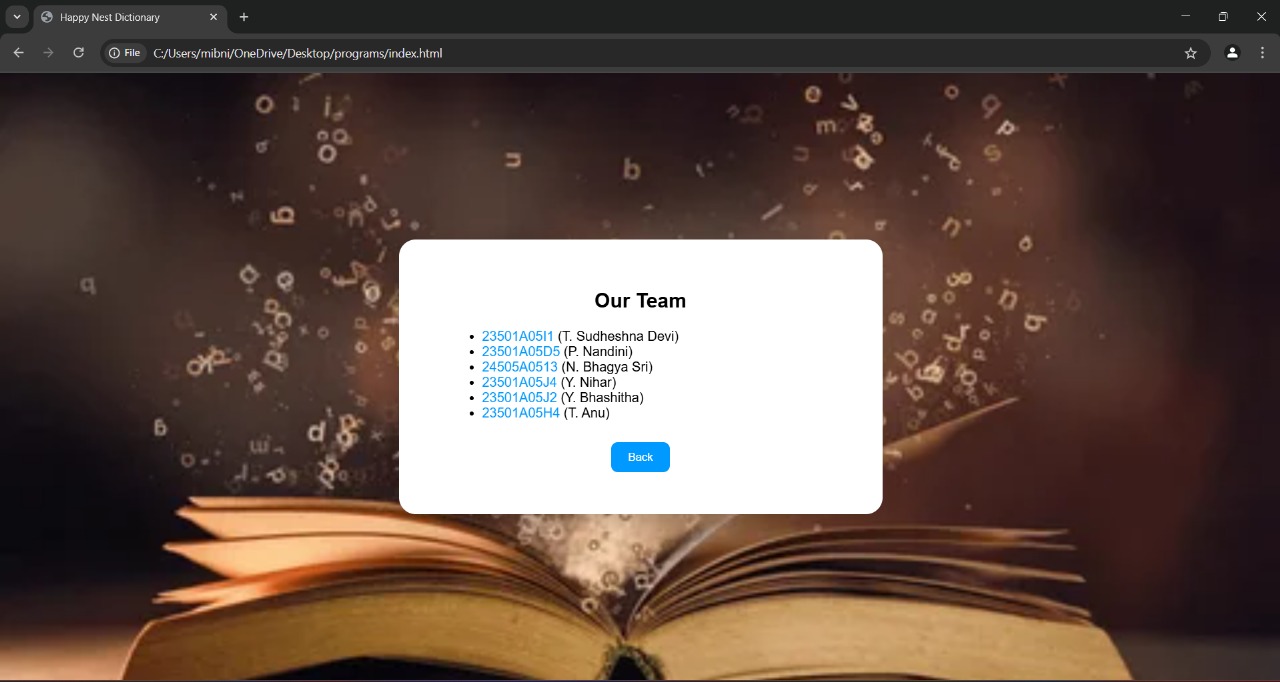
****

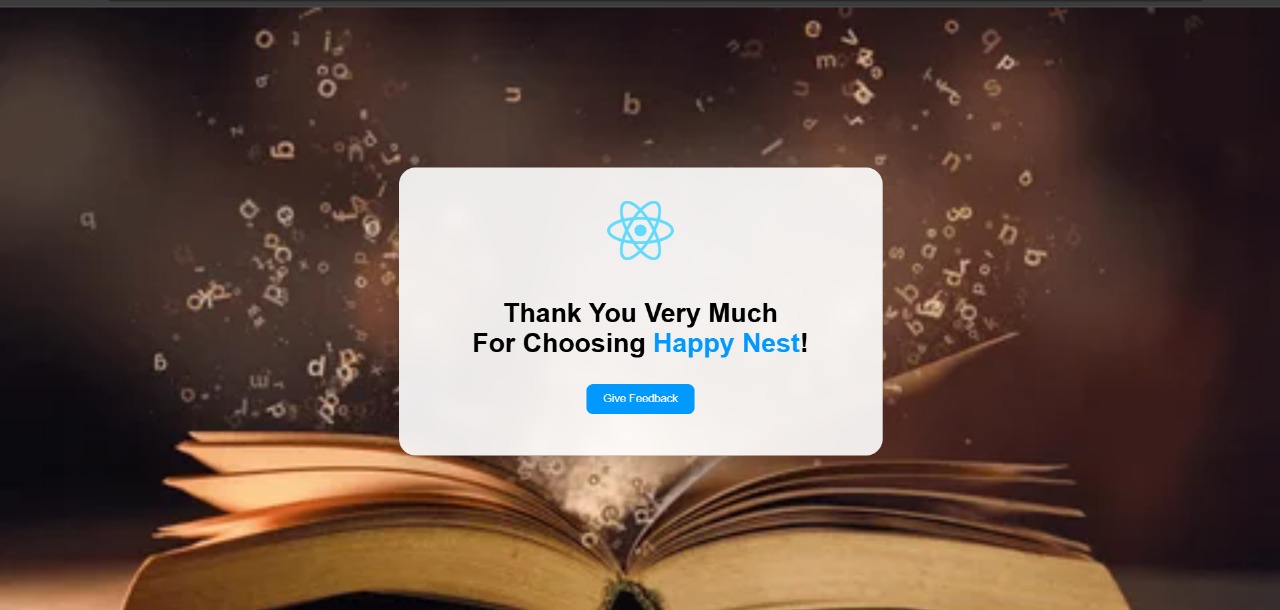
****

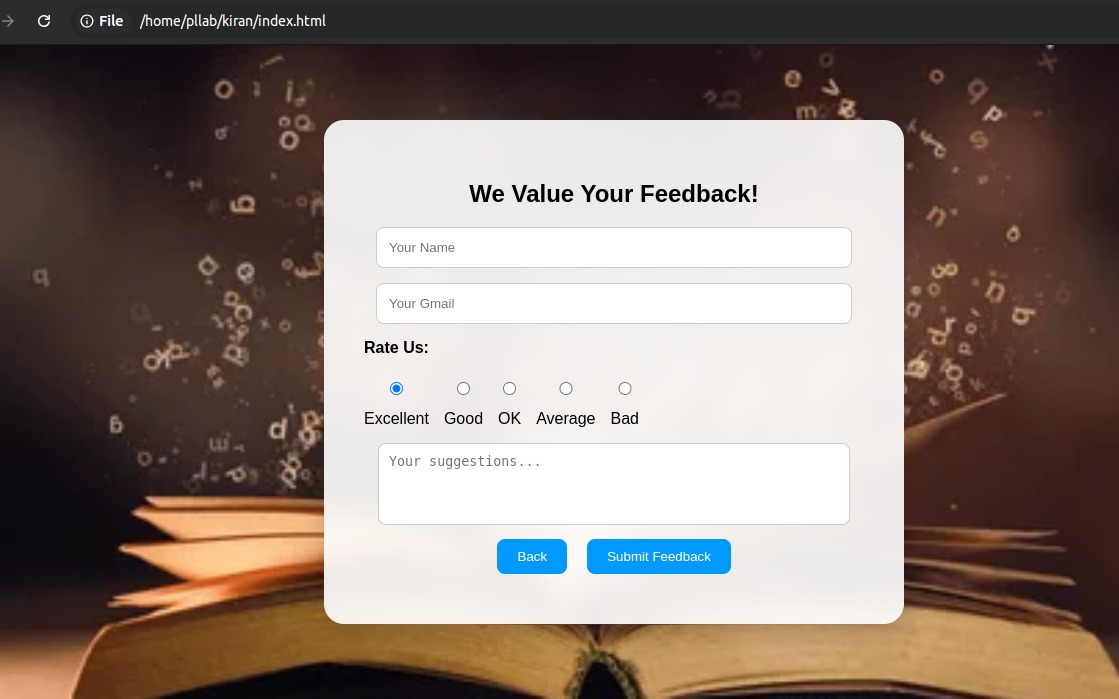
****

****

****

****

****

****

**8.Conclusion**

The development of Word Nest as a modern, web-based dictionary platform represents a step forward in making language learning more accessible, engaging, and user-centric. By combining essential dictionary functionalities with an intuitive user interface and responsive design, Word Nest offers a seamless experience for users across all age groups and devices. The inclusion of features such as real-time search, pronunciation support, usage examples, and customizable user settings enhances both the educational and practical value of the platform.

With its clean layout, thoughtful navigation, and performance optimization, Word Nest not only simplifies the process of exploring and understanding words but also encourages users to build and expand their vocabulary consistently. Furthermore, the integration of trending words and a "Word of the Day" feature helps keep users engaged and motivated to learn new terms regularly.

By leveraging modern web technologies and ensuring accessibility standards are met, the platform is designed to be inclusive, scalable, and adaptable for future enhancements. In conclusion, Word Nest successfully bridges the gap between traditional dictionaries and modern digital experiences, offering a smart, efficient, and enriching tool for language exploration in the digital age.

**9.References**

Github Repository link:

https://github.com/nihar5776/Dictionary-webpage

1.HTML Introduction

https://www.w3schools.com/html/html\_intro.asp

2.CSS Introduction

https://www.w3schools.com/css/css\_intro.asp

3.JAVA SCRIPT Introduction

https://www.w3schools.com/js/default.asp