

A Field Project Report on

TRIBUTE PAGE

Submitted

In partial fulfillment of the requirements for the award of the degree

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE and ENGINEERING

By

V.SRUJITH NANDAN	231FA04759
CH. NITHYA SREE	231FA04780
SK. GOUSE BAJI	231FA04783
P. SAI TARUN	231FA04787

Under the Guidance of

MR. T.Narasimha rao

Assistant Professor, CSE



VIGNAN'S

FOUNDATION FOR SCIENCE, TECHNOLOGY & RESEARCH

(Deemed to be University) - Estd. u/s 3 of UGC Act 1956

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SCHOOL OF COMPUTING AND INFORMATICS

VIGNAN'S FOUNDATION FOR SCIENCE, TECHNOLOGY & RESEARCH
(Deemed to be University)

Vadlamudi, Guntur -522213, INDIA.

April, 2025



VIGNAN'S


FOUNDATION FOR SCIENCE, TECHNOLOGY & RESEARCH

(Deemed to be University) - Estd. u/s 3 of UGC Act 1956

CERTIFICATE

This is to certify that the field project entitled "*TRIBUTE PAGE*" is being submitted by [V.SRUJITH NANDAN], [231FA04759], [CH.NITHYA SREE], [231FA04780], [SK.GOUSE BAJI], [231FA04783], and [P.SAI TARUN], [231FA04787] in partial fulfilment of the requirements for the degree of Bachelor of Technology (B.Tech.) in Computer Science and Engineering at Vignan's Foundation for Science, Technology and Research (Deemed to be University), Vadlamudi, Guntur District, Andhra Pradesh, India.


This is a bonafide work carried out by the aforementioned students under my guidance and supervision.



Guide



Project Review Committee



HoD, CSE

HoD
Dept. of Computer Science & Engineering
VFSTR Deemed to be University
VADLAMUDI - 522 213
Guntur Dist., A.P., India.



VIGNAN'S

FOUNDATION FOR SCIENCE, TECHNOLOGY & RESEARCH

(Deemed to be University) - Estd. u/s 3 of UGC Act 1956

DECLARATION

Date:09-04-2025

We hereby declare that the work presented in the field project titled "TRIBUTE PAGE" is the result of our own efforts and investigations.

This project is being submitted under the supervision of **Mr. T.Narasimha rao, Assistant Professor, CSE** in partial fulfillment of the requirements for the Bachelor of Technology (B.Tech.) degree in Computer Science and Engineering at Vignan's Foundation for Science, Technology and Research (Deemed to be University), Vadlamudi, Guntur, Andhra Pradesh, India.

V.SRUJITH NANDAN

(231FA04759)

CH.NITHYA SREE

(231FA04780)

SK.GOUSE BAJI

(231FA04783)

P. SAI TARUN

(231FA04787)

V. Srujith

Ch. Nithya Sree

Gouse Baji Shaik

P. Sai Tarun

CONTENTS

Chapter No.		Description	Page number
1		Introduction	4
	1.1	Problem definitions	5
	1.2	Existing system	5
	1.3	Proposed system	5-6
	1.4	Literature review	6
2		System Requirements	7
	2.1	Hardware & Software requirements	7
	2.2	Software requirements Specification (SRS)	8
3		System Design	8
	3.1	Modules of system	9
	3.2	UML Diagrams	9-10
4		Implementation	11
	4.1	Sample code	11-67
5		Results	68
	5.1	Output Screens	68-71
6		Conclusion	72
		References	73

1.INTRODUCTION

The Interactive Tribute Page is designed to provide an engaging platform for users to learn about historical figures through a multimedia experience. This project aims to enhance user interaction by integrating search functionalities, image galleries, and video content, thereby creating a comprehensive tribute to notable individuals. The primary objective is to facilitate educational exploration while ensuring an intuitive user interface.

1.1 Problem Definition:

A tribute page is a simple yet meaningful web page dedicated to honoring a person, group, or event. It serves as a digital space to showcase key achievements, contributions, and personal stories, often featuring images, quotes, and a timeline of important milestones. The page should be well-structured, visually appealing, and informative, ensuring visitors can quickly grasp the significance of the subject. The goal is to create an engaging and respectful tribute that effectively conveys admiration and appreciation while maintaining a clean and responsive design for an optimal user experience.

1.2 Existing System

Current tribute pages often lack interactivity and fail to engage users effectively. Many existing systems present static information, limiting user engagement and exploration. Furthermore, they often do not incorporate multimedia elements, which are essential for a richer learning experience. This project addresses these limitations by offering a dynamic platform that combines text, images, and videos.

1.3 Proposed Software

The proposed system introduces an interactive tribute page that allows users to search for historical figures and view detailed information, including summaries, childhood experiences, career highlights, and achievements. Key features include:

Search Functionality: Users can search for individuals using a text input field.**Multimedia Integration:** The system displays images and videos related to the searched individual.

User History: A search history feature allows users to revisit previous searches, enhancing usability.

1.4 Literature Review

1. Introduction

A tribute page is a dedicated web page designed to honor and commemorate a person, group, or event. It serves as a digital space where historical context, achievements, and contributions

are displayed in a structured manner. The emergence of tribute pages has been facilitated by the evolution of web technologies and digital storytelling techniques, allowing users to engage with content dynamically. Tribute pages have become an essential part of web-based memorialization, blending visual elements, narratives, and interactive components to create a compelling tribute.

2. Historical Perspective on Digital Memorialization

The concept of digital memorials can be traced back to early web development when static HTML pages were created to honor deceased individuals. With the advent of content management systems (CMS) and social media, online tributes evolved to include multimedia elements such as videos, timelines, and interactive features. Online memorialization has been studied extensively in human-computer interaction (HCI), with researchers exploring how digital spaces can facilitate remembrance and emotional connections. According to Jones et al. (2010), digital memorials provide a unique avenue for preserving legacies while ensuring accessibility to a global audience.

3. Design Principles for Tribute Pages

The effectiveness of a tribute page relies on its design, structure, and usability. Studies in web design suggest that an optimal tribute page should adhere to the following principles:

- **Clarity and Readability:** The page should present information in a clear and concise manner, using appropriate typography and color schemes to enhance readability.
- **Visual Appeal:** The inclusion of images, videos, and animations can improve user engagement and emotional connection with the subject.

Non-Functional Requirements

The system should load search results within 2 seconds.

The user interface must be responsive and accessible on various devices.

Security measures must be in place to protect user data

2.SYSTEM REQUIREMENTS

- **Hardware Requirements:**
 - A modern computer or server with at least 8GB RAM and a multi-core processor for hosting dynamic content.
 - Sufficient storage (SSD preferred) for multimedia elements such as images and videos.
 - High-speed internet connection for seamless access and interactivity.
- **Software Requirements:**
 - A robust CMS like WordPress, Drupal, or a custom-built solution using HTML, CSS, and JavaScript.
 - Backend frameworks such as Node.js, Django, or Laravel for managing data and user interactions.
 - Database management using MySQL, PostgreSQL, or Firebase for storing and retrieving information.

2.1 Hardware Requirements

Minimum Requirements:

Processor: Dual-core 2.0 GHz

RAM: 4 GB

Storage: 500 MB available space

Recommended Requirements:

Processor: Quad-core 2.5 GHz

RAM: 8 GB

Storage: 1 GB available space

Software Requirements

Operating System: Windows 10, macOS Mojave or later, Linux (Ubuntu 18.04 or later)

Web Browser: Google Chrome, Mozilla Firefox, Safari

Libraries/Frameworks: jQuery, Bootstrap, and the Wikipedia API for data retrieval.

2.2 Software Requirements Specification

Functional Requirements

Users must be able to search for individuals by name.

The system must display relevant information, including images and videos.

Users should be able to view their search history and clear it if desired.

3. SYSTEM DESIGN

The architecture of the Interactive Tribute Page follows a client-server model. The client-side is developed using HTML, CSS, and JavaScript, while the server-side utilizes the Wikipedia API for data retrieval. The user interface is designed to be intuitive, with a focus on usability and accessibility.

3.1 Modules of System

Search Module

This module handles user input and queries the Wikipedia API for relevant data.

Display Module

Responsible for rendering the search results, including text, images, and videos.

History Module

Manages the storage and retrieval of user search history, allowing users to revisit previous searches.

3.2 UML Diagram

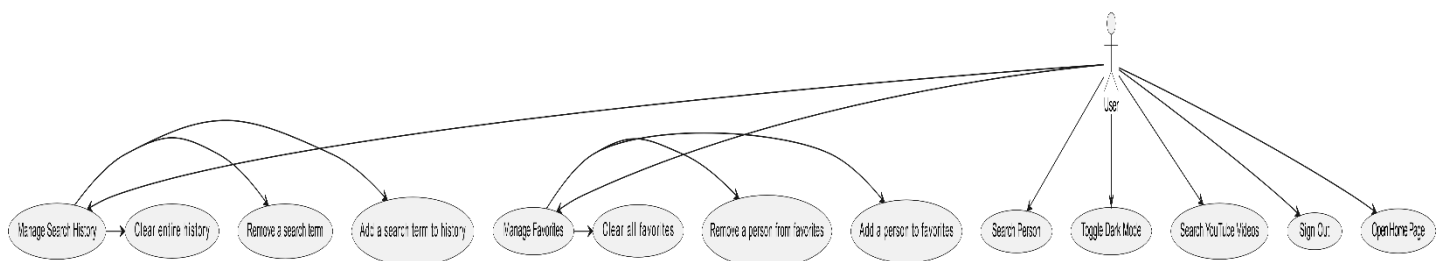
Use Case Diagram

Description:

The **User** interacts with different features of the system, such as searching for a person, managing search history, and organizing favorites. The system maintains a history and allows users to toggle settings like dark mode. Additionally, YouTube search and authentication functionalities are provided.

Actors:

- **User** – The primary actor interacting with the system.



Use Cases:

1. **Search Person** – User searches for a person.
2. **Manage Search History**
 - Add a search term to history
 - Remove a search term
 - Clear entire history

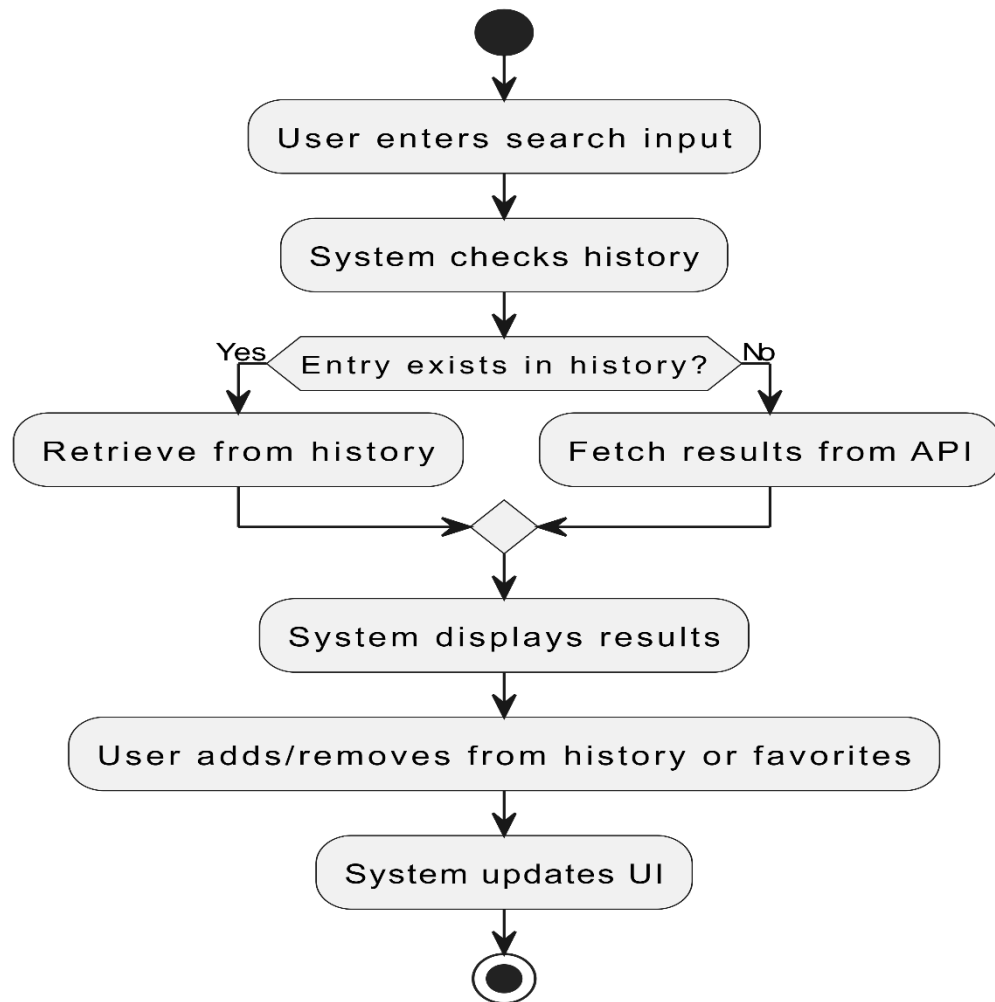
3. **Manage Favorites**
 - Add a person to favorites
 - Remove a person from favorites
 - Clear all favorites
4. **Toggle Dark Mode** – Enable or disable dark mode.
5. **Search YouTube Videos** – Fetch relevant YouTube videos.
6. **Sign Out** – User logs out of the system.
7. **Open Home Page** – Navigate back to the homepage.

Class diagram:

- SearchSystem manages user searches.
 - SearchHistory stores and retrieves past searches.
 - Favorites keeps track of users' favorite entries.
-
- **Textual Representation:**
 - Each class has methods to support specific functionalities.
 - SearchSystem is linked to SearchHistory and Favorites, ensuring searches can be saved or retrieved.
 - The UIHandler ensures smooth user experience and interaction.

Activity Diagram Description:

- The process starts when the user enters a search input.
- The system first checks if the entry exists in history.
- If found, it retrieves the result; otherwise, it fetches new data.
- The system then displays results, and the user can manage history or favorites.



4.IMPLEMENTATION

The development environment includes Visual Studio Code for coding, Git for version control, and the Wikipedia API for data access. The code is organized into separate files for HTML, CSS, and JavaScript, ensuring modularity and maintainability.

4.1 Code:

login page:

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

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

  <title>Login Page</title>

  <style>

    /* General Styles */

    * {

      margin: 0;

      padding: 0;

      box-sizing: border-box;

      font-family: Arial, sans-serif;

    }

    body {

      height: 100vh;

      background: linear-gradient(135deg, #e5d3f8, #e5e23f);

      display: flex;

      justify-content: center;

      align-items: center;

      color: #333;

    }

    /* Container */
```

```
.container {  
    background: white;  
    padding: 30px;  
    border-radius: 12px;  
    box-shadow: 0 8px 20px rgba(0, 0, 0, 0.3);  
    width: 350px;  
    text-align: center;  
    transition: transform 0.3s;  
}
```

```
.container:hover {  
    transform: scale(1.05);  
}
```

```
/* Heading */
```

```
h2 {  
    font-size: 28px;  
    margin-bottom: 20px;  
    color: #444;  
}
```

```
/* Form Fields */
```

```
form label {  
    display: block;  
    text-align: left;  
    margin-bottom: 8px;  
    font-weight: bold;  
}
```

```
form input {
```

```
width: 100%;
padding: 12px;
margin-bottom: 18px;
border: 1px solid #ccc;
border-radius: 5px;
transition: 0.3s;
}

form input:focus {
border-color: #6a11cb;
outline: none;
box-shadow: 0 0 8px #6a11cb;
}

/* Forgot Password */
.forgot {
text-align: right;
margin-bottom: 20px;
}

.forgot a {
color: #2575fc;
text-decoration: none;
font-size: 14px;
}

.forgot a:hover {
text-decoration: underline;
}
```

```

/* Buttons */

.btn {
    display: block;
    text-align: center;
    width: 100%;
    padding: 12px;
    margin: 10px 0;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    font-size: 16px;
    color: white;
    background: #4CAF50;
    transition: 0.3s;
    text-decoration: none;
}

.btn:hover {
    background: #45a049;
}

/* Responsive Design */
@media (max-width: 400px) {
    .container {
        width: 90%;
    }
}

/* Sign Out Button Styles */
.sign-out-btn {

```

```
padding: 0.5rem 1rem;
background: #7209b7; /* Secondary color */
color: white;
border: none;
border-radius: 25px;
cursor: pointer;
font-size: 1rem;
transition: 0.3s;
box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
position: absolute; /* Position it in the top right corner */
top: 1rem;
right: 1rem;
}
```

```
.sign-out-btn:hover {
  background: #f72585; /* Accent color */
  transform: translateY(-2px);
  box-shadow: 0 0 15px rgba(247, 37, 133, 0.5);
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div class="container">
```

```
<h2>Tribute Page Login</h2>
```

```
<form action="#" method="POST">
```

```
<label for="username">Email:</label>
```

```
<input type="text" id="username" name="username" placeholder="Enter Email" required>
```

```
<label for="password">Password:</label>
```

```
<input type="password" id="password" name="password" placeholder="Enter Password"
required>
```

```
<div class="forgot">
```

```
<a href="temp2.html">Forgot Password?</a>
```

```
</div>
```

```
<!-- Login Button -->
```

```
<a href="1page.html" class="btn">Login</a>
```

```
</form>
```

```
</div>
```

```
<!-- <button class="sign-out-btn" onclick="signOut()">Sign Out</button> -->
```

```
<!-- <script>
```

```
function signOut() {
```

```
    window.location.href = 'temp4.html'; // Redirect to temp3.html
```

```
}
```

```
</script> -->
```

```
</body>
```

```
</html>
```

Forget page:

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="UTF-8">
```

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

```
<title>Forgot Password</title>
```

```
<style>
```

```
/* General Styling */
```



```
* {  
    box-sizing: border-box;  
    margin: 0;  
    padding: 0;  
    font-family: Arial, sans-serif;  
}  
  
body {  
    height: 100vh;  
    background: linear-gradient(to right, #f8f6f5, #f4d1d2);  
    display: flex;  
    justify-content: center;  
    align-items: center;  
    color: #333;  
}  
  
.middle-container {  
    background: white;  
    border-radius: 15px;  
    box-shadow: 0 0 15px rgba(0, 0, 0, 0.2);  
    padding: 30px;  
    width: 50%;  
    max-width: 500px;  
    text-align: center;  
    transition: 0.3s;  
}  
  
.middle-container:hover {  
    transform: scale(1.05);  
}
```

```
h2 {  
    font-size: 28px;  
    margin-bottom: 20px;  
    color: #444;  
}
```

```
.login-container {  
    background: white;  
    padding: 20px;  
    border-radius: 10px;  
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);  
    margin-bottom: 20px;  
}
```

```
input {  
    width: 100%;  
    padding: 12px;  
    margin: 10px 0;  
    border: 1px solid #ccc;  
    border-radius: 5px;  
    transition: 0.3s;  
}
```

```
input:focus {  
    border-color: #007BFF;  
    box-shadow: 0 0 8px #007BFF;  
}
```

```
button {
```

```
width: 100%;  
padding: 12px;  
border: none;  
border-radius: 5px;  
color: white;  
font-size: 16px;  
cursor: pointer;  
transition: 0.3s;  
}
```

```
.login-btn {  
  background-color: #28a745;  
}
```

```
.login-btn:hover {  
  background-color: #218838;  
}
```

```
.signin-btn {  
  background-color: #007bff;  
}
```

```
.signin-btn:hover {  
  background-color: #0056b3;  
}
```

```
.disabled {  
  background-color: gray !important;  
  cursor: not-allowed;  
}
```

```
.links {
    margin-top: 20px;
}

.links a {
    display: block;
    text-decoration: none;
    color: #007bff;
    margin: 5px 0;
}

.links a:hover {
    text-decoration: underline;
}

#resend-otp {
    margin-top: 10px;
    font-size: 14px;
    color: red;
}

@media (max-width: 600px) {
    .middle-container {
        width: 90%;
    }
}

</style>
</head>
<body>
```

```

<div class="middle-container">
  <h2>Forgot Password</h2>

  <!-- Email Section -->
  <div class="login-container" id="email-section">
    <input type="email" id="email" placeholder="Enter your email" required>
    <button class="login-btn" onclick="sendOTP()" id="send-otp-btn">Send OTP</button>
  </div>

  <!-- OTP Section -->
  <div class="login-container" id="otp-section" style="display: none;">
    <input type="text" id="otp" placeholder="Enter OTP" required>
    <button class="login-btn" onclick="verifyOTP()">Verify OTP</button>
    <p id="resend-otp">Resend OTP in <span id="timer">30</span> seconds</p>
    <button class="signin-btn disabled" id="resend-btn" onclick="resendOTP()"
disabled>Resend OTP</button>
  </div>

  <!-- Reset Password Section -->
  <div class="login-container" id="reset-section" style="display: none;">
    <input type="password" id="new-password" placeholder="Enter new password" required>
    <button class="signin-btn" onclick="resetPassword()">Reset Password</button>
  </div>

  <div class="links">
    <a href="temp4.html">Back to Login</a>
  </div>
</div>

<script>

```

```

let generatedOTP = "";
let countdown;

// Function to send OTP
function sendOTP() {
    const email = document.getElementById("email").value;
    if (!email) {
        alert("Please enter your email.");
        return;
    }

    generatedOTP = Math.floor(100000 + Math.random() * 900000); // Generate 6-digit OTP
    alert("OTP Sent: " + generatedOTP); // Simulate OTP sending

    document.getElementById("email-section").style.display = "none";
    document.getElementById("otp-section").style.display = "block";

    startResendTimer();
}

// Function to verify OTP
function verifyOTP() {
    const enteredOTP = document.getElementById("otp").value;
    if (enteredOTP == generatedOTP) {
        alert("OTP Verified Successfully!");
        document.getElementById("otp-section").style.display = "none";
        document.getElementById("reset-section").style.display = "block";
    } else {
        alert("Invalid OTP! Try Again.");
    }
}

```

```

}

// Function to reset password
function resetPassword() {
    const newPassword = document.getElementById("new-password").value;
    if (!newPassword) {
        alert("Enter a new password.");
        return;
    }

    alert("Password Reset Successful! You can now log in.");
    window.location.href = "temp4.html"; // Redirect to login
}

// Function to start the resend OTP timer
function startResendTimer() {
    let timeLeft = 30; // 30 seconds
    document.getElementById("resend-btn").disabled = true;
    document.getElementById("resend-btn").classList.add("disabled");

    countdown = setInterval(() => {
        timeLeft--;
        document.getElementById("timer").innerText = timeLeft;

        if (timeLeft <= 0) {
            clearInterval(countdown);
            document.getElementById("resend-btn").disabled = false;
            document.getElementById("resend-btn").classList.remove("disabled");
            document.getElementById("resend-otp").innerText = "Didn't receive OTP? Click below  
to resend.";
        }
    }, 1000);
}

```

```

    }, 1000);
}

// Function to resend OTP
function resendOTP() {
    generatedOTP = Math.floor(100000 + Math.random() * 900000); // Generate new OTP
    alert("New OTP Sent: " + generatedOTP); // Simulate OTP sending

    document.getElementById("resend-otp").innerHTML = "Resend OTP in <span id='timer'>30</span> seconds";
    startResendTimer();
}
</script>

</body>
</html>

```

Project:

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Interactive Tribute Page</title>
    <style>
        /* Modern CSS Reset */
        * {
            margin: 0;
            padding: 0;
            box-sizing: border-box;
        }
    </style>

```



```

/* Custom Properties */

:root {
  --primary-color: #4361ee;
  --primary-color-rgb: 67, 97, 238; /* Added RGB values for primary color */
  --secondary-color: #7209b7;
  --accent-color: #f72585;
  --background-color: #f8f9fa;
  --text-color: #222;
  --box-shadow: 0 8px 30px rgba(0, 0, 0, 0.12);
  --transition: all 0.3s cubic-bezier(0.25, 0.8, 0.25, 1);
  --gradient-start: #4361ee;
  --gradient-mid: #7209b7;
  --gradient-end: #f72585;
}

body.dark-mode {
  --background-color: #121212;
  --text-color: #f8f9fa;
  --primary-color: #4cc9f0;
  --primary-color-rgb: 76, 201, 240; /* Added RGB values for primary color */
  --secondary-color: #7209b7;
  --accent-color: #f72585;
  --gradient-start: #4cc9f0;
  --gradient-mid: #7209b7;
  --gradient-end: #f72585;
}

/* Fix for text color in dark mode */
body.dark-mode section {
  background: rgba(40, 40, 40, 0.9);

```

```

}

body.dark-mode .video-description,
body.dark-mode p,
body.dark-mode h2,
body.dark-mode .search-term,
body.dark-mode .video-title {
    color: var(--text-color);
}

/* Background gradient */
body {
    background: linear-gradient(135deg, var(--background-color) 0%, var(--background-color) 100%);
    background-attachment: fixed;
}

/* Animations */
@keyframes fadeIn {
    from { opacity: 0; transform: translateY(20px); }
    to { opacity: 1; transform: translateY(0); }
}

@keyframes pulse {
    0% { transform: scale(1); }
    50% { transform: scale(1.05); }
    100% { transform: scale(1); }
}

@keyframes gradientBackground {
    0% { background-position: 0% 50%; }

```

```
50% { background-position: 100% 50%; }  
100% { background-position: 0% 50%; }  
}
```

```
@keyframes floatEffect {  
  0% { transform: translateY(0px); }  
  50% { transform: translateY(-10px); }  
  100% { transform: translateY(0px); }  
}
```

```
@keyframes slideInRight {  
  from { transform: translateX(100%); opacity: 0; }  
  to { transform: translateX(0); opacity: 1; }  
}
```

```
@keyframes slideOutRight {  
  from { transform: translateX(0); opacity: 1; }  
  to { transform: translateX(100%); opacity: 0; }  
}
```

```
@keyframes slideInLeft {  
  from { transform: translateX(-100%); opacity: 0; }  
  to { transform: translateX(0); opacity: 1; }  
}
```

```
@keyframes slideOutLeft {  
  from { transform: translateX(0); opacity: 1; }  
  to { transform: translateX(-100%); opacity: 0; }  
}
```

```

/* Spinner for loading */

@keyframes spin {
  0% { transform: rotate(0deg); }
  100% { transform: rotate(360deg); }
}

.spinner {
  width: 50px;
  height: 50px;
  border: 5px solid var(--background-color);
  border-top: 5px solid var(--primary-color);
  border-radius: 50%;
  animation: spin 1s linear infinite;
  margin: 20px auto;
}

/* Base Styles */

body {
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
  background-color: var(--background-color);
  color: var(--text-color);
  line-height: 1.6;
  overflow-x: hidden;
  transition: var(--transition);
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 2rem;

```

```
    position: relative;
    min-height: 100vh;
}

button {
    padding: 0.75rem 1.5rem;
    background: var(--primary-color);
    color: white;
    border: none;
    border-radius: 25px;
    cursor: pointer;
    font-size: 1rem;
    transition: var(--transition);
    box-shadow: var(--box-shadow);
}

button:hover {
    background: var(--secondary-color);
    transform: translateY(-2px);
    box-shadow: 0 0 15px var(--accent-color);
}

input[type="text"] {
    padding: 0.75rem 1rem;
    width: 70%;
    border: none;
    border-radius: 25px;
    box-shadow: var(--box-shadow);
    font-size: 1rem;
    transition: var(--transition);
}
```

```

    }

input[type="text"]:focus {
    outline: none;
    box-shadow: 0 0 0 3px var(--accent-color);
    transform: translateY(-2px);
}

/* Header Styles */
header {
    text-align: center;
    margin-bottom: 3rem;
    position: relative;
    z-index: 1;
}

.title {
    font-size: 3.5rem;
    margin-bottom: 1.5rem;
    background: linear-gradient(90deg, var(--gradient-start), var(--gradient-mid), var(--gradient-end));
    background-size: 200% 200%;
    -webkit-background-clip: text;
    -webkit-text-fill-color: transparent;
    animation: gradientBackground 5s ease infinite;
    text-shadow: 2px 5px rgba(0, 0, 0, 0.1);
}x

.search-container {
    display: flex;
    justify-content: center;

```

```

    gap: 0.5rem;
    max-width: 500px;
    margin: 0 auto;
}

/* Search History */
.search-history-section {
    margin: 2rem auto;
    max-width: 800px;
    background: white;
    border-radius: 15px;
    padding: 2rem;
    box-shadow: var(--box-shadow);
    transition: var(--transition);
}

body.dark-mode .search-history-section {
    background: rgba(40, 40, 40, 0.9);
}

.search-history-header {
    display: flex;
    justify-content: space-between;
    align-items: flex-start;
    margin-bottom: 1rem;
}

/* Search History Section Title */
.search-history-header h2 {
    margin-bottom: 0;

```

```

    position: relative;
    display: inline-block;
}

.search-history-header h2::after {
    content: "";
    position: absolute;
    left: 0;
    bottom: -5px;
    width: 40px;
    height: 3px;
    background: var(--accent-color);
    transition: var(--transition);
}

.search-history-header h2:hover::after {
    width: 100%;
}

/* Recent Searches Label */
.recent-searches-label {
    font-size: 0.9rem;
    color: var(--secondary-color);
    margin-top: 0.5rem;
    margin-bottom: 0.5rem;
    display: block;
}

body.dark-mode .recent-searches-label {
    color: var(--accent-color);
}

```



```
}

.search-history-list {
  list-style: none;
  margin-top: 1rem;
}

.search-history-item {
  display: flex;
  justify-content: space-between;
  align-items: center;
  padding: 0.75rem 1rem;
  background: var(--background-color);
  margin-bottom: 0.5rem;
  border-radius: 8px;
  transition: var(--transition);
}

.search-history-item:hover {
  transform: translateX(5px);
  background: rgba(var(--primary-color-rgb), 0.1);
}

.search-term {
  cursor: pointer;
  font-weight: 500;
}

.search-term:hover {
  color: var(--primary-color);
}
```

```
}
```

```
.remove-btn {  
  background: none;  
  color: var(--accent-color);  
  border: none;  
  font-size: 1rem;  
  cursor: pointer;  
  padding: 0.25rem 0.5rem;  
  border-radius: 4px;  
  transition: var(--transition);  
  box-shadow: none;  
}
```

```
.remove-btn:hover {  
  color: white;  
  background: var(--accent-color);  
}
```

```
.clear-all-btn {  
  background: var(--accent-color);  
  color: white;  
  box-shadow: none;  
  padding: 0.5rem 1rem;  
  font-size: 0.9rem;  
}
```

```
.history-empty {  
  text-align: center;  
  padding: 1rem;
```

```

        color: var(--secondary-color);
        font-style: italic;
    }

    /* Section styling */
    section {
        margin-bottom: 2rem;
        animation: fadeIn 0.5s ease-out;
        background: white;
        border-radius: 15px;
        padding: 2rem;
        box-shadow: var(--box-shadow);
        transition: var(--transition);
    }

    section:hover {
        box-shadow: 0 14px 28px rgba(0, 0, 0, 0.25);
        transform: translateY(-5px);
    }

    h2 {
        color: var(--primary-color);
        margin-bottom: 1rem;
        position: relative;
        display: inline-block;
    }

    h2::after {
        content: "";
        position: absolute;

```

```
    left: 0;
    bottom: -5px;
    width: 50%;
    height: 3px;
    background: var(--accent-color);
    transition: var(--transition);
}
```

```
h2: hover::after {
    width: 100%;
}
```

```
/* Home Page */
#home {
    display: block;
}
```

```
/* Tribute Page */
#tribute {
    display: none;
    animation: fadeIn 0.5s ease-out;
}
```

```
.profile-header {
    display: grid;
    grid-template-columns: 1fr 2fr;
    gap: 2rem;
    margin-bottom: 2rem;
}
```

```

@media (max-width: 768px) {
  .profile-header {
    grid-template-columns: 1fr;
  }
}

#image {
  width: 100%;
  max-width: 400px;
  border-radius: 10px;
  box-shadow: var(--box-shadow);
  transition: var(--transition);
  animation: floatEffect 6s ease-in-out infinite;
}

#caption {
  font-style: italic;
  color: var(--secondary-color);
  margin-bottom: 1rem;
}

/* Tab navigation */
.tabs {
  display: flex;
  justify-content: center;
  margin-bottom: 1rem;
}

.tab-button {
  padding: 0.5rem 1rem;

```

```
background: none;
border: none;
color: var(--secondary-color);
cursor: pointer;
transition: var(--transition);
border-bottom: 2px solid transparent;
box-shadow: none;
}

.tab-button.active {
color: var(--primary-color);
border-bottom: 2px solid var(--primary-color);
font-weight: bold;
}

.tab-button:hover {
background: none;
color: var(--primary-color);
transform: none;
box-shadow: none;
}

.tab-content {
display: none;
}

.tab-content.active {
display: block;
animation: fadeIn 0.5s ease-out;
}
```

```
/* Photo Gallery */
#photo-gallery {
  display: grid;
  grid-template-columns: repeat(auto-fill, minmax(200px, 1fr));
  gap: 1rem;
  margin-top: 1rem;
}

#photo-gallery img {
  width: 100%;
  border-radius: 8px;
  box-shadow: var(--box-shadow);
  transition: var(--transition);
}

#photo-gallery img:hover {
  transform: scale(1.05);
}

/* Video Gallery */
#video-gallery {
  display: grid;
  grid-template-columns: repeat(auto-fill, minmax(300px, 1fr));
  gap: 1.5rem;
  margin-top: 1rem;
}

.video-card {
  border-radius: 10px;
```

```

    overflow: hidden;

    box-shadow: var(--box-shadow);

    transition: var(--transition);

    background: rgba(255, 255, 255, 0.9);
}

body.dark-mode .video-card {
    background: rgba(40, 40, 40, 0.9);
}

.video-card:hover {
    transform: translateY(-5px);

    box-shadow: 0 14px 28px rgba(0, 0, 0, 0.25);
}

.video-thumbnail {
    position: relative;
    width: 100%;
    padding-top: 56.25%; /* 16:9 Aspect Ratio */
}

.video-thumbnail iframe {
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    border: none;
}

```



```

.video-info {
  padding: 1rem;
}

.video-title {
  font-weight: bold;
  margin-bottom: 0.5rem;
  color: var(--primary-color);
}

.video-description {
  font-size: 0.9rem;
  color: var(--text-color);
  opacity: 0.8;
}

/* Footer */
footer {
  text-align: center;
  padding: 1.5rem;
  color: var(--secondary-color);
  font-size: 0.9rem;
  margin-top: 3rem;
}

/* Dark mode toggle */
.dark-mode-toggle {
  position: fixed;
  top: 1rem;
  right: 1rem;

```

```

background: var(--primary-color);
color: white;
width: 40px;
height: 40px;
border-radius: 50%;
display: flex;
align-items: center;
justify-content: center;
cursor: pointer;
transition: var(--transition);
z-index: 1000;
}

.dark-mode-toggle:hover {
transform: rotate(45deg);
background: var(--secondary-color);
}

/* Achievements Styles */
.achievements-title {
margin-top: 2rem;
color: var(--primary-color);
font-size: 1.5rem;
position: relative;
display: inline-block;
}

.achievements-title::after {
content: "";
position: absolute;

```

```

    left: 0;
    bottom: -5px;
    width: 50%;
    height: 2px;
    background: var(--accent-color);
    transition: var(--transition);
}

.achievements-title:hover::after {
    width: 100%;
}

.achievements-list {
    margin-top: 1rem;
    list-style-type: none;
}

.achievements-list li {
    padding: 0.8rem 0;
    position: relative;
    padding-left: 2rem;
    line-height: 1.6;
    border-bottom: 1px solid rgba(0,0,0,0.05);
    transition: var(--transition);
}

.achievements-list li:last-child {
    border-bottom: none;
}

```

```

.achievements-list li:before {
    content: '◆';
    color: var(--accent-color);
    position: absolute;
    left: 0;
    font-size: 1.2rem;
    transition: var(--transition);
}

.achievements-list li:hover {
    transform: translateX(5px);
    color: var(--primary-color);
    font-weight: 500;
}

.achievements-list li:hover:before {
    color: var(--primary-color);
    transform: scale(1.2);
}

/* Mouse Trail Effect */
.cursor-trail {
    pointer-events: none;
    position: absolute;
    width: 10px;
    height: 10px;
    border-radius: 50%;
    background: var(--accent-color);
    opacity: 0.5;
    z-index: 9999;
}

```

```
    transform: translate(-50%, -50%);  
    transition: width 0.3s, height 0.3s, opacity 0.3s;  
}
```

```
/* Favorite Button Styles */
```

```
.tribute-header-controls {  
    display: flex;  
    justify-content: space-between;  
    align-items: center;  
    margin-bottom: 1rem;  
    width: 100%;  
}
```

```
.favorite-btn {  
    background-color: transparent;  
    border: 2px solid var(--accent-color);  
    color: var(--accent-color);  
    padding: 0.5rem 1rem;  
    border-radius: 20px;  
    cursor: pointer;  
    transition: var(--transition);  
}
```

```
.favorite-btn:hover {  
    background-color: var(--accent-color);  
    color: white;  
}
```

```
.favorite-btn.active {  
    background-color: var(--accent-color);
```

```

        color: white;
    }

    /* Favorites list styles */
    .favorites-list {
        list-style-type: none;
    }

    .favorite-item {
        display: flex;
        justify-content: space-between;
        align-items: center;
        padding: 0.75rem 1rem;
        background: var(--background-color);
        margin-bottom: 0.5rem;
        border-radius: 8px;
        transition: var(--transition);
    }

    .favorite-item:hover {
        transform: translateX(5px);
        background: rgba(var(--primary-color-rgb), 0.1);
    }

    .sign-out-btn {
padding: 0.5rem 1rem;
background: var(--secondary-color);
color: white;
border: none;
border-radius: 25px;
cursor: pointer;

```

```

    font-size: 1rem;

    transition: var(--transition);

    box-shadow: var(--box-shadow);

    position: absolute; /* Position it in the top right corner */
    top: 1rem;
    right: 1rem;
}

.sign-out-btn:hover {
    background: var(--accent-color);
    transform: translateY(-2px);
    box-shadow: 0 0 15px var(--accent-color);
}

</style>
</head>
<body>

    <div class="dark-mode-toggle" onclick="toggleDarkMode()">🌙</div>

    <div class="container">
        <!-- Home Page -->
        <div id="home">
            <header>
                <h1 class="title">Tribute Page</h1>
                <div class="search-container">
                    <input type="text" id="search-input" placeholder="Search for a person..."
onkeypress="checkEnter(event)">
                    <button onclick="searchPerson()">Search</button>
                </div>
                <button class="sign-out-btn" onclick="signOut()">Sign Out</button>
            </header>

```

```
<div id="search-result"></div>
```

```
<div class="search-history-section">
```

```
  <div class="search-history-header">
```

```
    <div>
```

```
      <h2>Search History</h2>
```

```
      <span class="recent-searches-label">Your recent searches will appear here</span>
```

```
    </div>
```

```
    <button class="clear-all-btn" onclick="clearAllHistory()">Clear All</button>
```

```
  </div>
```

```
  <ul id="search-history-list" class="search-history-list">
```

```
    <li class="history-empty">No search history yet. Search for someone!</li>
```

```
  </ul>
```

```
</div>
```

```
<!-- Favorites Section -->
```

```
<div class="search-history-section">
```

```
  <div class="search-history-header">
```

```
    <div>
```

```
      <h2>Favorites</h2>
```

```
      <span class="recent-searches-label">Your favorite people will appear here</span>
```

```
    </div>
```

```
    <button class="clear-all-btn" onclick="clearAllFavorites()">Clear All</button>
```

```
  </div>
```

```
  <ul id="favorites-list" class="search-history-list">
```

```
    <li class="history-empty">No favorites yet. Add someone to favorites!</li>
```

```
  </ul>
```

```
</div>
```

```
</div>
```



```

<!-- Tribute Page -->

<div id="tribute">

  <header>

    <div class="tribute-header-controls">

      <button onclick="showHome()">← Back</button>

      <button id="favorite-btn" class="favorite-btn" onclick="toggleFavorite()">♥ Add
to Favorites</button>

    </div>

    <h1 id="title" class="title">Person Name</h1>

    <p id="caption">Caption</p>

  </header>

  <section class="profile-header">

    <div class="image-container">

      <img id="image" src="" alt="Person Image">

    </div>

    <div class="profile-info">

      <div class="tabs">

        <button class="tab-button active" onclick="openTab(event,
'summary')">Summary</button>

        <button class="tab-button" onclick="openTab(event,
'childhood')">Childhood</button>

        <button class="tab-button" onclick="openTab(event, 'career')">Career</button>

        <button class="tab-button" onclick="openTab(event,
'achievements')">Achievements</button>

      </div>

      <div id="summary" class="tab-content active">

        <p id="summary-text"></p>

      </div>

```

```
<div id="childhood" class="tab-content">
  <p id="childhood-text"></p>
</div>

<div id="career" class="tab-content">
  <p id="career-text"></p>
</div>

<div id="achievements" class="tab-content">
  <p id="achievements-text"></p>
</div>
</div>
</section>

<section>
  <h2>Photo Gallery</h2>
  <div id="photo-gallery"></div>
</section>

<section>
  <h2>Videos</h2>
  <div id="video-gallery"></div>
</section>
</div>

<footer>
  <p>&copyright 2025 | Interactive Tribute Page</p>
</footer>
</div>
```

```

<script>

  // Load page and search history on page load

  let searchHistory = JSON.parse(localStorage.getItem('searchHistory')) || [];
  let favorites = JSON.parse(localStorage.getItem('favorites')) || [];

  window.onload = function() {
    createMouseTrail();
    updateSearchHistoryDisplay();
    updateFavoritesDisplay();
    applyStoredDarkModePreference();
  };

  async function searchPerson() {
    const input = document.getElementById('search-input').value.trim();
    if (!input) {
      document.getElementById('search-result').innerHTML = '<p>Please enter a
name.</p>';
      return;
    }

    // Clear previous search results

    document.getElementById('search-result').innerHTML = '<h3>Search
Results</h3><p>Loading...</p><div class="spinner"></div>';

    try {
      const searchTerm = input.toLowerCase();

      const searchResponse = await
fetch(`https://en.wikipedia.org/w/api.php?action=query&origin=*&list=search&srsearch=${enc
odeURIComponent(searchTerm)}&format=json`);

      const searchData = await searchResponse.json();

      if (searchData.query.search.length === 0) {

```

```

        document.getElementById('search-result').innerHTML = '<p>No results found.</p>';
        return;
    }

    const pageTitle = searchData.query.search[0].title;

    const pageResponse = await
    fetch(`https://en.wikipedia.org/w/api.php?action=query&origin=*&prop=extracts|pageimages|d
    escription&explaintext&titles=${encodeURIComponent(pageTitle)}&format=json&redirects=1
    &piprop=original`);

    const pageData = await pageResponse.json();
    const pages = pageData.query.pages;
    const pageId = Object.keys(pages)[0];
    if (pageId === '-1') {
        document.getElementById('search-result').innerHTML = '<p>No results found.</p>';
        return;
    }
    const page = pages[pageId];

    // Add to search history
    addToSearchHistory(page.title);

    document.getElementById('home').style.display = 'none';
    document.getElementById('tribute').style.display = 'block';
    document.getElementById('title').innerText = page.title;
    document.getElementById('caption').innerText = page.description || '';

    // Update favorite button state
    updateFavoriteButtonState(page.title);

    if (page.original && page.original.source) {
        document.getElementById('image').src = page.original.source;
    }

```

```

        document.getElementById('image').style.display = 'block';
    } else {
        document.getElementById('image').style.display = 'none';
    }

    // Set summary, childhood, career, and achievements text
    const extract = page.extract.split('\n');

    document.getElementById('summary-text').innerText = extract[0] || "Summary not
    available."; // First paragraph as summary

    document.getElementById('childhood-text').innerText = extract[1] || "Childhood
    information not available."; // Placeholder text

    document.getElementById('career-text').innerText = extract[2] || "Career information
    not available."; // Placeholder text

    document.getElementById('achievements-text').innerText = extract[3] || "Achievements
    information not available."; // Placeholder text

    // Fetch images for photo gallery

    const imagesResponse = await
    fetch(`https://en.wikipedia.org/w/api.php?action=query&origin=*&prop=images&titles=${enco
    deURIComponent(page.title)}&format=json`);

    const imageData = await imagesResponse.json();
    const imagesPage = imageData.query.pages[pageId];

    const photoGallery = document.getElementById('photo-gallery');
    photoGallery.innerHTML = "";

    if (imagesPage.images) {
        const imageTitles = imagesPage.images.slice(0, 5).map(img => img.title);

        for (const title of imageTitles) {
            const imageInfoResponse = await
            fetch(`https://en.wikipedia.org/w/api.php?action=query&origin=*&titles=${encodeURICompo
            nent(title)}&prop=imageinfo&iiprop=url&format=json`);

```

```

const imageInfoData = await imageInfoResponse.json();
const imageId = Object.keys(imageInfoData.query.pages)[0];
const imageInfo = imageInfoData.query.pages[imageId];

if (imageInfo.imageinfo) {
  const imgUrl = imageInfo.imageinfo[0].url;
  const img = document.createElement('img');
  img.src = imgUrl;
  img.alt = title;
  photoGallery.appendChild(img);
}
}
}

// Fetch YouTube videos
await searchYouTubeVideos(page.title);

} catch (error) {
  console.error(error);

  document.getElementById('search-result').innerHTML = '<p>Error fetching data.
Please try again later.</p>';
}
}

// Search history functions
function addToSearchHistory(term) {
  // Remove duplicate if exists
  searchHistory = searchHistory.filter(item => item !== term);

  // Add new term at the beginning
  searchHistory.unshift(term);
}

```

```

// Keep only the last 10 searches
if (searchHistory.length > 10) {
    searchHistory.pop();
}

// Save to localStorage
localStorage.setItem('searchHistory', JSON.stringify(searchHistory));

// Update display
updateSearchHistoryDisplay();
}

function updateSearchHistoryDisplay() {
    const historyList = document.getElementById('search-history-list');
    const recentLabel = document.querySelector('.recent-searches-label');

    if (searchHistory.length === 0) {
        historyList.innerHTML = '<li class="history-empty">No search history yet. Search for
someone!</li>';
        recentLabel.textContent = 'Your recent searches will appear here';
        return;
    }

    historyList.innerHTML = "";
    recentLabel.textContent = `Your recent searches (${searchHistory.length})`;

    searchHistory.forEach((term, index) => {
        const li = document.createElement('li');
        li.className = 'search-history-item';

```

```

const termSpan = document.createElement('span');
termSpan.className = 'search-term';
termSpan.textContent = term;
termSpan.onclick = function() {
    document.getElementById('search-input').value = term;
    searchPerson();
};

const removeBtn = document.createElement('button');
removeBtn.className = 'remove-btn';
removeBtn.textContent = '×';
removeBtn.setAttribute('title', 'Remove from history');
removeBtn.onclick = function(e) {
    e.stopPropagation();
    removeFromHistory(index);
};

li.appendChild(termSpan);
li.appendChild(removeBtn);
historyList.appendChild(li);
});
}

function removeFromHistory(index) {
    searchHistory.splice(index, 1);
    localStorage.setItem('searchHistory', JSON.stringify(searchHistory));
    updateSearchHistoryDisplay();
}

function clearAllHistory() {

```



```

searchHistory = [];
localStorage.setItem('searchHistory', JSON.stringify(searchHistory));
updateSearchHistoryDisplay();
}

// Favorites functions
function updateFavoritesDisplay() {
  const favoritesList = document.getElementById('favorites-list');

  const favoritesLabel = document.querySelector('.search-history-section:nth-of-type(2)
.recent-searches-label');

  if (favorites.length === 0) {
    favoritesList.innerHTML = '<li class="history-empty">No favorites yet. Add someone
to favorites!</li>';

    favoritesLabel.textContent = 'Your favorite people will appear here';
    return;
  }

  favoritesList.innerHTML = "";
  favoritesLabel.textContent = `Your favorites (${favorites.length})`;

  favorites.forEach((favorite, index) => {
    const li = document.createElement('li');
    li.className = 'search-history-item';

    const favoriteSpan = document.createElement('span');
    favoriteSpan.className = 'search-term';
    favoriteSpan.textContent = favorite;
    favoriteSpan.onclick = function() {
      document.getElementById('search-input').value = favorite;
      searchPerson();
    }
  });
}

```

```

    };

    const removeBtn = document.createElement('button');
    removeBtn.className = 'remove-btn';
    removeBtn.textContent = '×';
    removeBtn.setAttribute('title', 'Remove from favorites');
    removeBtn.onclick = function(e) {
        e.stopPropagation();
        removeFromFavorites(favorite);
    };

    li.appendChild(favoriteSpan);
    li.appendChild(removeBtn);
    favoritesList.appendChild(li);
});
}

function addToFavorites(person) {
    if (!favorites.includes(person)) {
        favorites.push(person);
        localStorage.setItem('favorites', JSON.stringify(favorites));
        updateFavoritesDisplay();
    }
    updateFavoriteButtonState(person);
}

function removeFromFavorites(person) {
    favorites = favorites.filter(favorite => favorite !== person);
    localStorage.setItem('favorites', JSON.stringify(favorites));
    updateFavoritesDisplay();
}

```

```

        updateFavoriteButtonState(person);
    }

    function signOut() {
window.location.href = 'temp4.html'; // Redirect to temp3.html
    }

function clearAllFavorites() {
    favorites = [];
    localStorage.setItem('favorites', JSON.stringify(favorites));
    updateFavoritesDisplay();
    // Update button state if we're viewing a person page
    const currentPerson = document.getElementById('title').textContent;
    if (currentPerson) {
        updateFavoriteButtonState(currentPerson);
    }
}

function toggleFavorite() {
    const personName = document.getElementById('title').textContent;
    if (favorites.includes(personName)) {
        removeFromFavorites(personName);
        document.getElementById('favorite-btn').textContent = "♥ Add to Favorites";
        document.getElementById('favorite-btn').classList.remove('active');
    } else {
        addToFavorites(personName);
        document.getElementById('favorite-btn').textContent = "♥ Remove from Favorites";
        document.getElementById('favorite-btn').classList.add('active');
    }
}

```

```

function updateFavoriteButtonState(personName) {
  const favoriteBtn = document.getElementById('favorite-btn');
  if (favorites.includes(personName)) {
    favoriteBtn.textContent = "💔 Remove from Favorites";
    favoriteBtn.classList.add('active');
  } else {
    favoriteBtn.textContent = "💖 Add to Favorites";
    favoriteBtn.classList.remove('active');
  }
}

async function searchYouTubeVideos(query) {
  const videoGallery = document.getElementById('video-gallery');
  videoGallery.innerHTML = '<p>Loading videos...</p>';

  try {
    // Clean the query for YouTube
    const sanitizedQuery = encodeURIComponent(query);

    // Create direct search links rather than playlists which may not work consistently
    const videoTypes = [
      {
        title: 'Biography',
        searchQuery: `${sanitizedQuery}+biography`,
        description: 'Learn about the life and achievements of this historical figure.'
      },
      {
        title: 'Documentary',
        searchQuery: `${sanitizedQuery}+documentary`,
        description: 'Watch documentaries featuring detailed explorations and analysis.'
      }
    ]
  }
}

```

```

    },
    {
      title: 'History',
      searchQuery: `${sanitizedQuery}+history`,
      description: 'Explore the historical context and impact on the world.'
    }
  ];

  let videoHTML = "";

  for (const video of videoTypes) {
    // Create a direct link to YouTube search instead of embedding which can be unreliable
    videoHTML += `
    <div class="video-card">
      <div class="video-thumbnail">
        <iframe
          src="https://www.youtube.com/embed?listType=search&list=${video.search
hQuery}"
          allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope;
picture-in-picture"
          allowfullscreen>
        </iframe>
      </div>
      <div class="video-info">
        <div class="video-title">${query} ${video.title}</div>
        <div class="video-description">${video.description}</div>
      </div>
    </div>`;
  }

  videoGallery.innerHTML = videoHTML;

```

```

// Add a backup message and link in case embeds don't work
videoGallery.innerHTML += `
<div style="margin-top: 1rem; text-align: center;">
  <p>If videos don't load properly, you can search directly on YouTube:</p>
  <a href="https://www.youtube.com/results?search_query=${sanitizedQuery}"
    target="_blank"
    style="display: inline-block; margin-top: 0.5rem; color: var(--primary-color); text-
decoration: none;">
    Search "${query}" on YouTube →
  </a>
</div>`;
} catch (error) {
  console.error('Error loading videos:', error);
  videoGallery.innerHTML = `
  <p>Unable to load videos at this time. Please try searching directly on YouTube.</p>
  <a
href="https://www.youtube.com/results?search_query=${encodeURIComponent(query)}"
  target="_blank"
  style="display: inline-block; margin-top: 0.5rem; color: var(--primary-color); text-
decoration: none;">
    Search "${query}" on YouTube →
  </a>`;
}
}

function showHome() {
  document.getElementById('home').style.display = 'block';
  document.getElementById('tribute').style.display = 'none';

  // Clear the search results, input field, and remove any loading animations

```

```

document.getElementById('search-result').innerHTML = "";
document.getElementById('search-input').value = "";

// Remove any spinners or loading animations that might be present
const spinners = document.querySelectorAll('.spinner');
spinners.forEach(spinner => {
    spinner.remove();
});
}

function checkEnter(event) {
    if (event.key === 'Enter') {
        searchPerson();
    }
}

function toggleDarkMode() {
    document.body.classList.toggle('dark-mode');
    // Store user's preference in localStorage
    const isDarkMode = document.body.classList.contains('dark-mode');
    localStorage.setItem('darkMode', isDarkMode ? 'enabled' : 'disabled');
}

// Apply dark mode on page load if previously selected
function applyStoredDarkModePreference() {
    const storedDarkMode = localStorage.getItem('darkMode');
    if (storedDarkMode === 'enabled') {
        document.body.classList.add('dark-mode');
    }
}

```

```

function openTab(evt, tabName) {
    // Hide all tab content
    const tabContents = document.getElementsByClassName('tab-content');
    for (let i = 0; i < tabContents.length; i++) {
        tabContents[i].classList.remove('active');
    }

    // Remove active class from all tab buttons
    const tabButtons = document.getElementsByClassName('tab-button');
    for (let i = 0; i < tabButtons.length; i++) {
        tabButtons[i].classList.remove('active');
    }

    // Show the current tab and add active class to the button that opened it
    document.getElementById(tabName).classList.add('active');
    evt.currentTarget.classList.add('active');
}

function createMouseTrail() {
    const trailCount = 20;
    const trails = [];

    // Create mouse trail elements
    for (let i = 0; i < trailCount; i++) {
        const trail = document.createElement('div');
        trail.className = 'cursor-trail';
        document.body.appendChild(trail);
        trails.push({
            element: trail,

```



```

        x: 0,
        y: 0,
        size: 8 - (i * 0.3),
        speed: 0.2 + (i * 0.01)
    });
}

let mouseX = 0;
let mouseY = 0;

// Update mouse position
document.addEventListener('mousemove', function(e) {
    mouseX = e.clientX;
    mouseY = e.clientY;
});

// Animate mouse trail
function animateTrails() {
    trails.forEach((trail, index) => {
        // Calculate new position
        trail.x += (mouseX - trail.x) * trail.speed;
        trail.y += (mouseY - trail.y) * trail.speed;

        // Update trail element position
        trail.element.style.left = trail.x + 'px';
        trail.element.style.top = trail.y + 'px';
        trail.element.style.width = trail.size + 'px';
        trail.element.style.height = trail.size + 'px';

        // Set opacity based on distance from cursor

```

```

    const distance = Math.sqrt(
      Math.pow(mouseX - trail.x, 2) +
      Math.pow(mouseY - trail.y, 2)
    );
    const opacity = Math.max(0, 1 - (distance / 200));
    trail.element.style.opacity = opacity * 0.6;
  });

  requestAnimationFrame(animateTrails);
}

animateTrails();

// Add hover effect to elements
const interactiveElements = document.querySelectorAll('button, img, .tab-button');
interactiveElements.forEach(element => {
  element.addEventListener('mouseover', function() {
    trails.forEach(trail => {
      trail.element.style.background = 'var(--primary-color)';
      trail.size *= 1.5;
    });
  });

  element.addEventListener('mouseout', function() {
    trails.forEach(trail => {
      trail.element.style.background = 'var(--accent-color)';
      trail.size /= 1.5;
    });
  });
});

```

```
    }  
  </script>  
</body>  
</html>\
```

Test Cases

Outline test cases to validate the functionality of the application:

Test Case 1: Verify that the search function returns results for valid input.

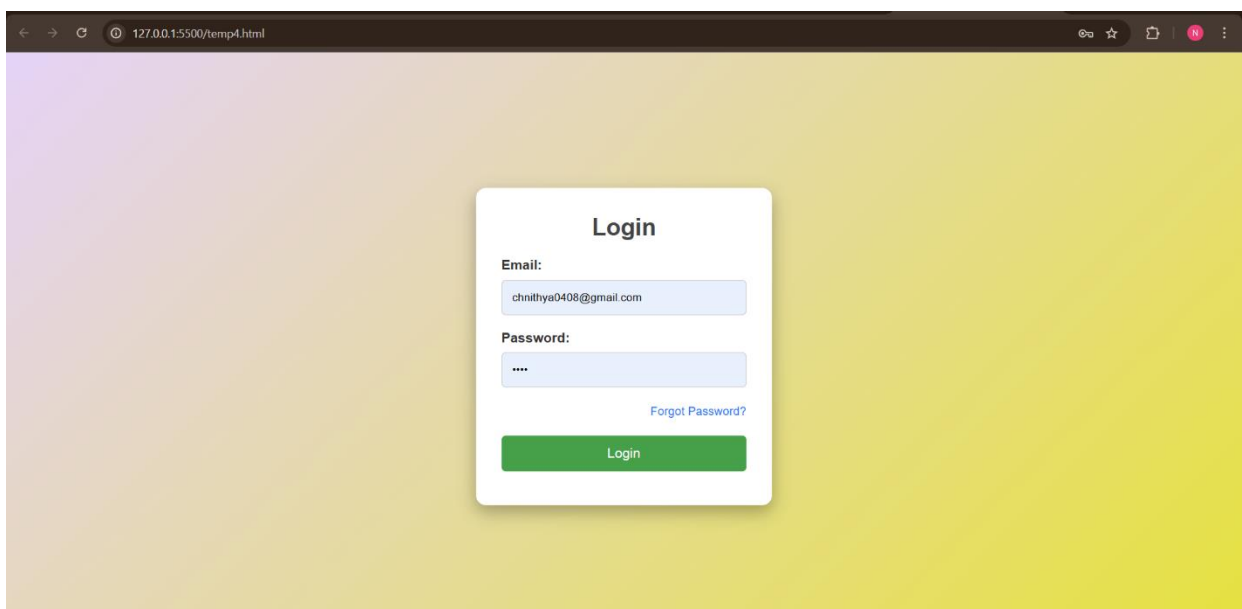
Test Case 2: Check that the application handles empty input gracefully.

Test Case 3: Ensure that the dark mode toggle works as expected.

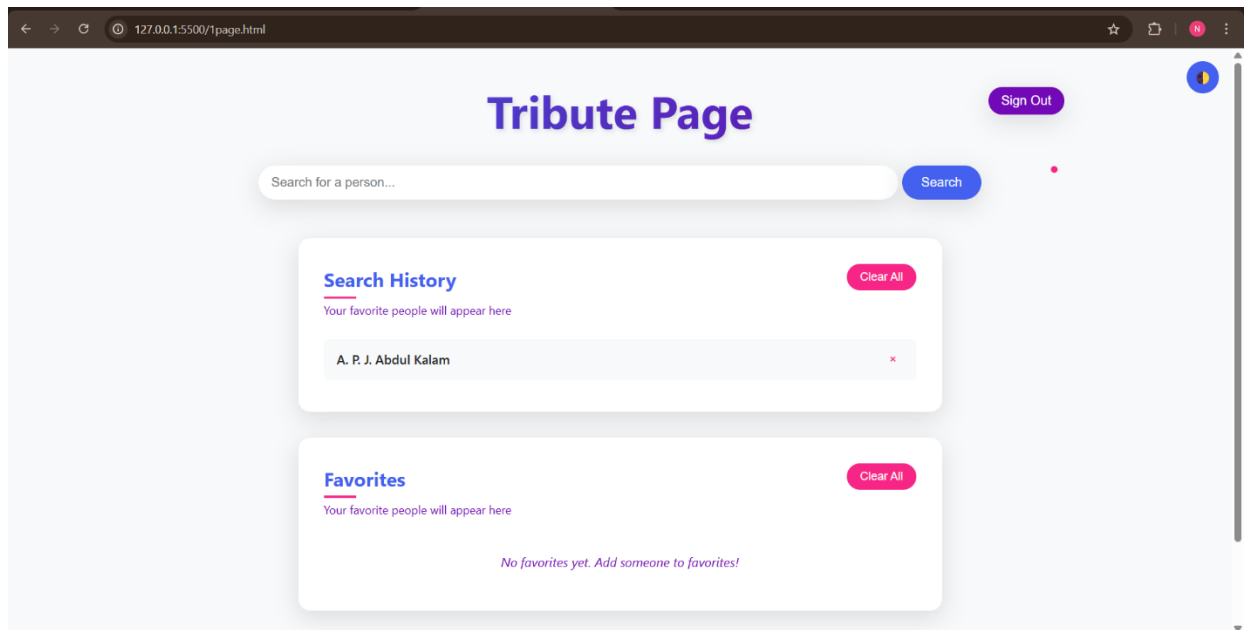
5.RESULTS

5.1 Output Screens:

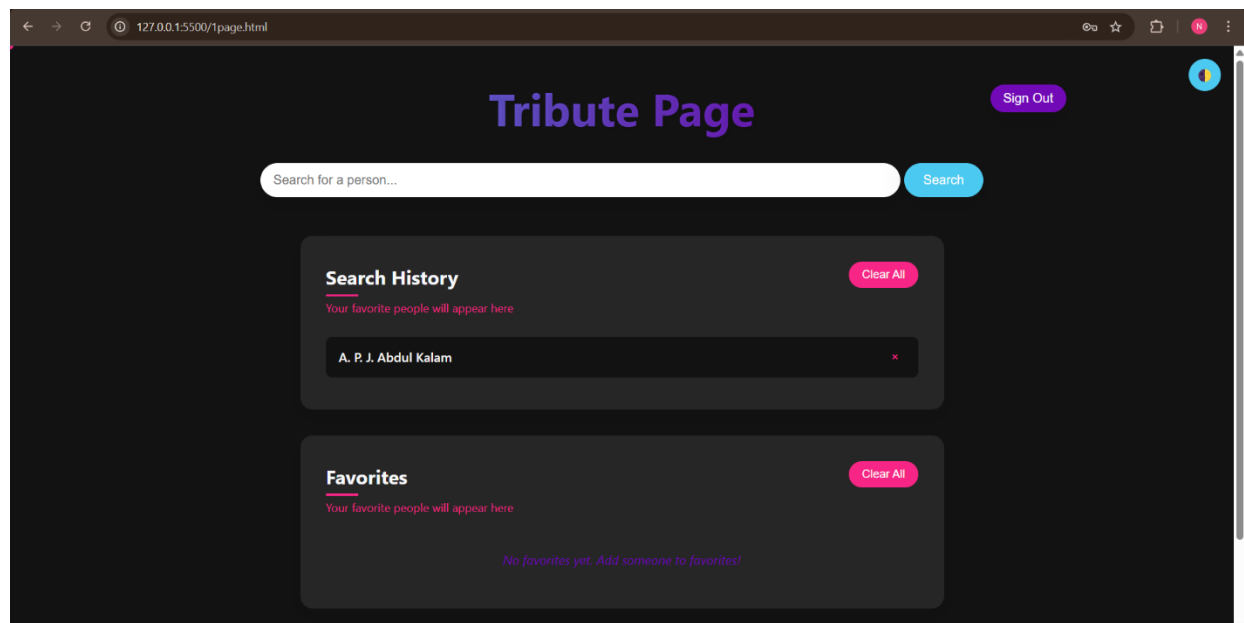
Login page:



Home page:



Dark Mode:



Search results:

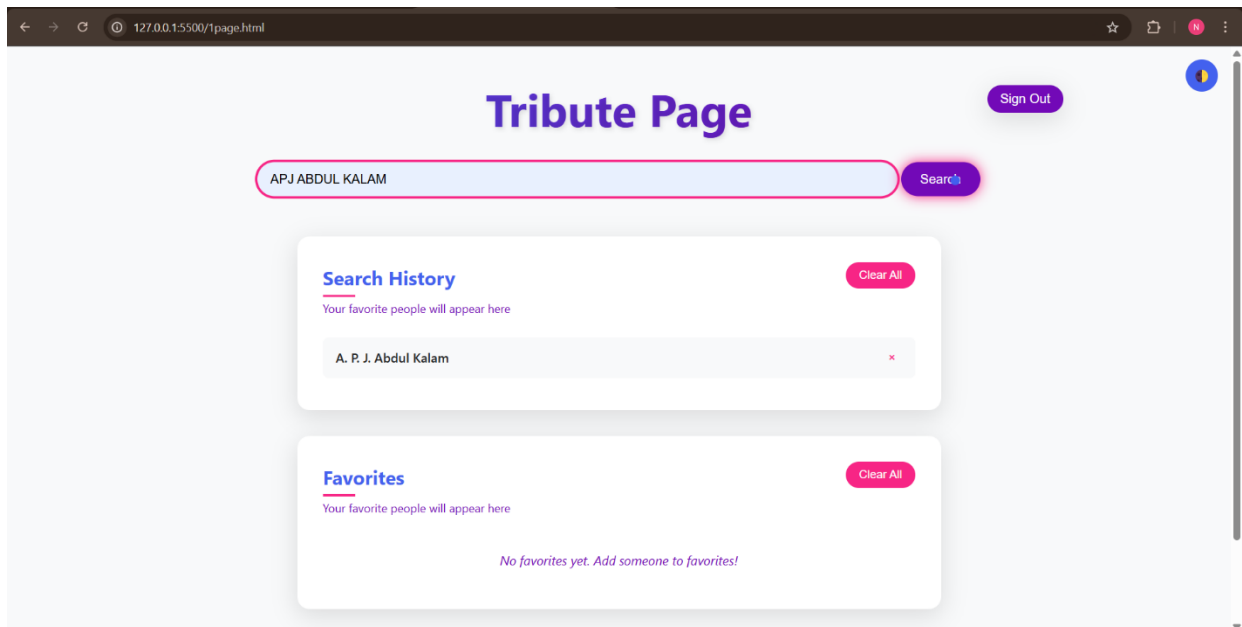
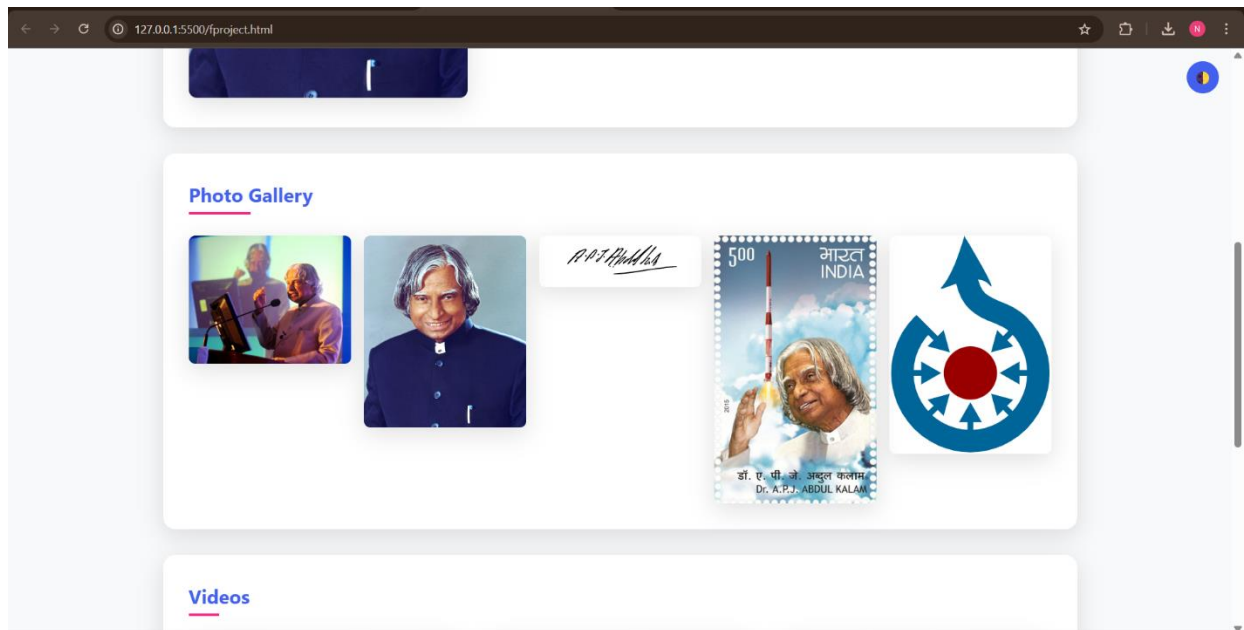
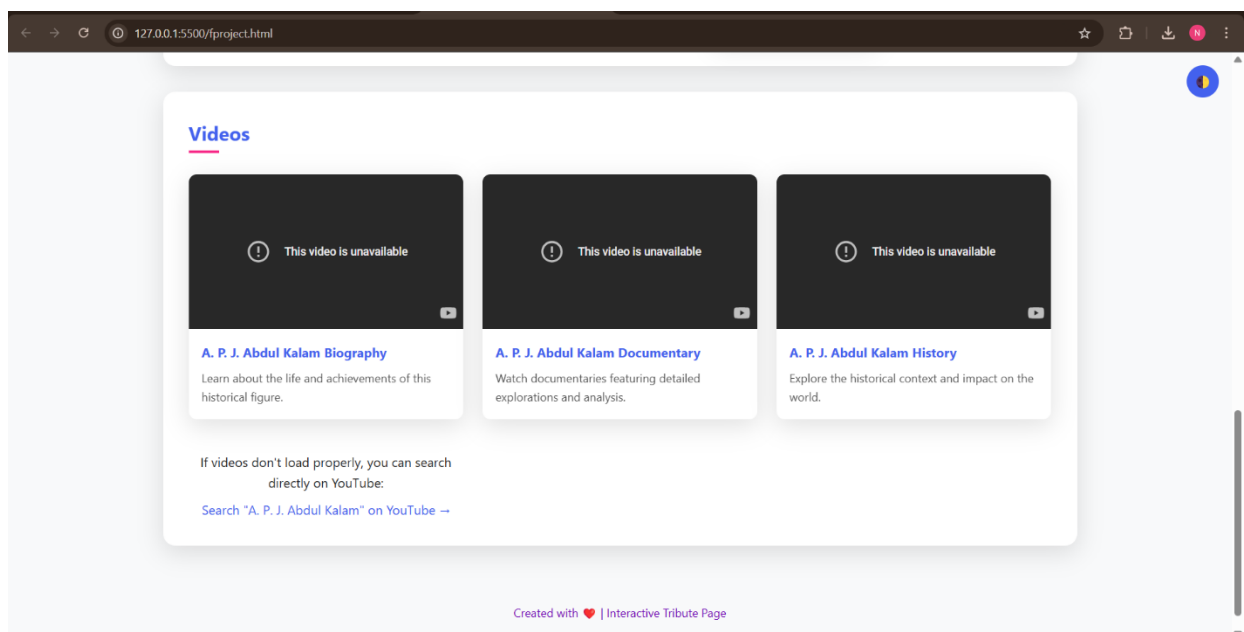


Photo gallery :



Videos:



6.CONCLUSION

The Interactive Tribute Page successfully integrates multimedia elements to enhance user engagement and learning. By addressing the limitations of existing tribute pages, this project provides a dynamic platform for exploring historical figures. Future work may include expanding the database of individuals and incorporating user-generated content.

6.1 References

□ **Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994).** *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.

- Used for structuring the class relationships and modularizing functionalities.

□ **Fowler, M. (2004).** *UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd ed.)*. Addison-Wesley.

- Referenced for UML diagram notations, including use case, class, and activity diagrams.

□ **Booch, G., Rumbaugh, J., & Jacobson, I. (2005).** *The Unified Modeling Language User Guide (2nd ed.)*. Addison-Wesley.

- Utilized for refining UML elements and interactions between system components.

□ **MDN Web Docs. (n.d.).** *JavaScript Guide*. Mozilla. Retrieved from <https://developer.mozilla.org/>

- Referenced for JavaScript-based UI handling and event-driven programming.

□ **YouTube Data API Documentation.** Google Developers. Retrieved from <https://developers.google.com/youtube/v3>

Project link:

⇒ <https://github.com/srujithnandan/Interactive-Tribute-Page.git>