



République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université des Sciences et de la Technologie Houari Boumediene

Faculté d'Informatique

Description Of the Muse hub Website

represented by :

Yahia Aissani

Groupe 1

The muse hub Website Documentation

Introduction

The "Art and Creativity" website is designed as a platform for artists to showcase their work, share creative ideas, learn new techniques, and connect with a community of art enthusiasts. This document outlines the features, structure, and implementation details of the website.

Presentation of the Mini-Project

This mini-project aims to create a static website with both admin and user access. The website provides a variety of art tutorials, a gallery for showcasing artworks, and user profile management functionalities. The website also allows users to post new art pieces and store them locally.

Problematic

Creating a user-friendly platform that allows artists to manage their profiles, upload artworks, and access valuable tutorials while ensuring smooth navigation and accessibility.

Objectives

- To provide a platform for artists and art enthusiasts to connect.
- To allow users to manage their profiles.
- To enable users to upload and display their artworks.
- To provide comprehensive art tutorials.

Input Interfaces

Login and Signup Page

- **Username or Email Input:** Allows users to enter their username or email.
- **Password Input:** Allows users to enter their password.
- **Role Selection:** Allows users to select their role (User or Artist).

Art Upload Form

- **Description Input:** Allows users to provide a description for their art.
- **File Upload:** Allows users to upload an image file of their artwork.
- **Post Button:** Submits the artwork for display.

Interface Admin

Admin access allows full control over the website content, including managing user accounts and moderating posted artworks.

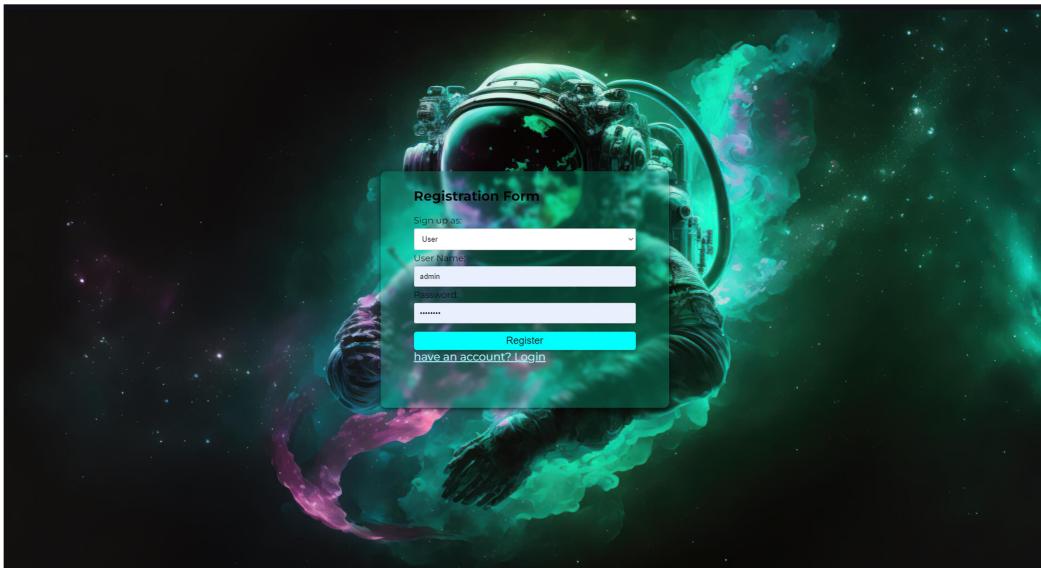
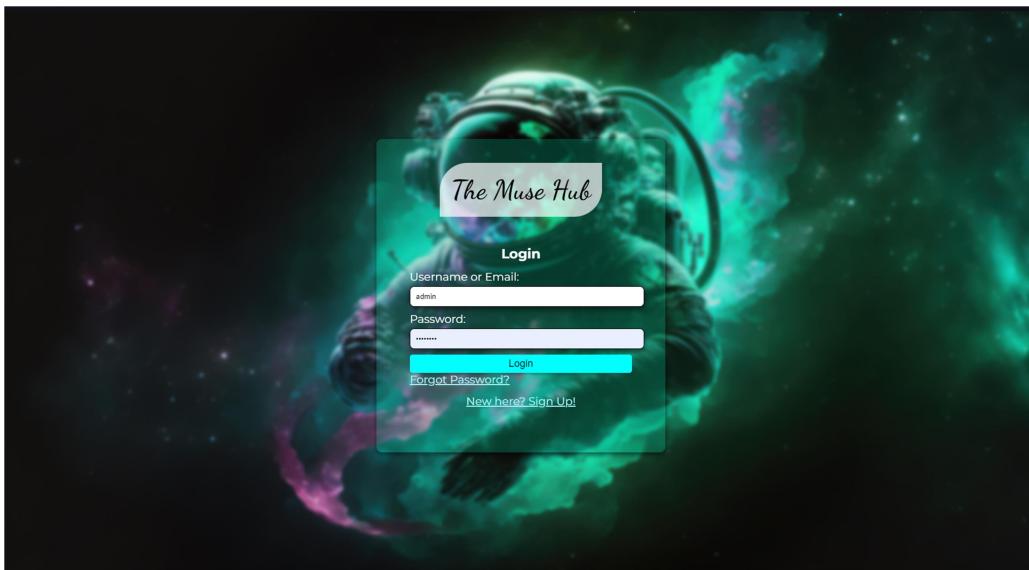
Interface User

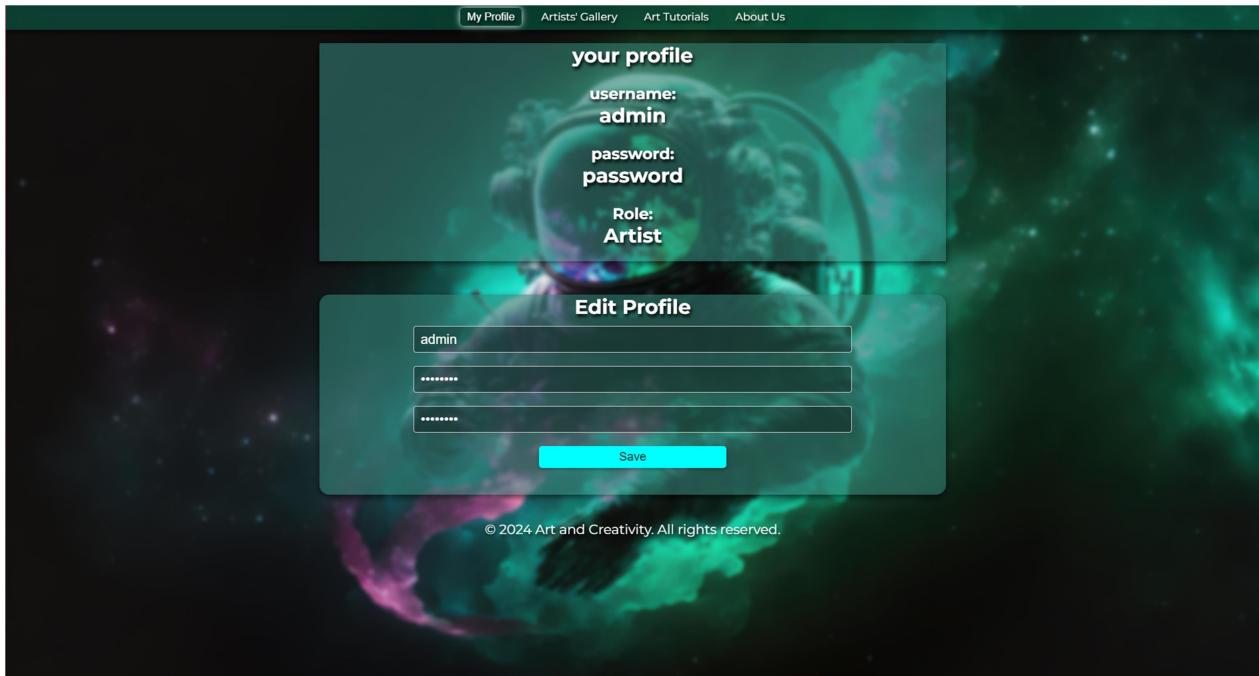
User access is limited to managing their profile, posting new artworks, and accessing tutorials.

Website Features

Feature 01: User Authentication

- **Description:** Users can sign up, log in, and manage their profiles.
- **Screenshot:**

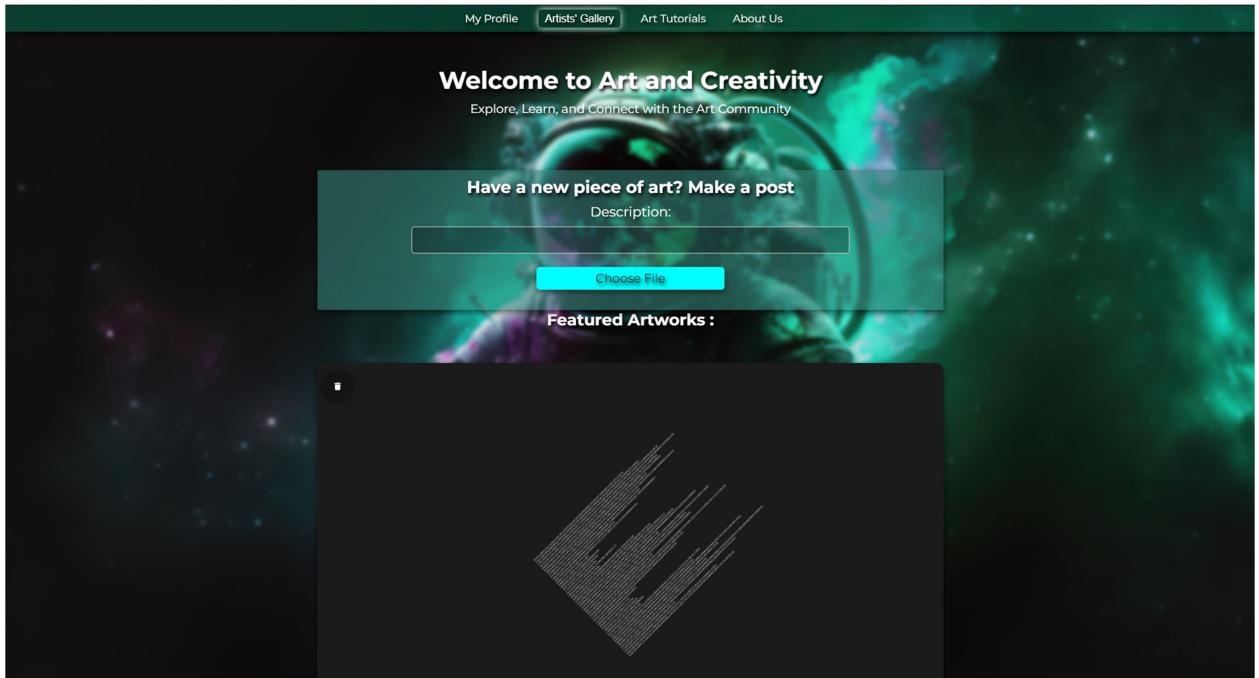




- **Objects Used:** Forms, Local Storage, Session Storage.

Feature 02: Art Upload and Display

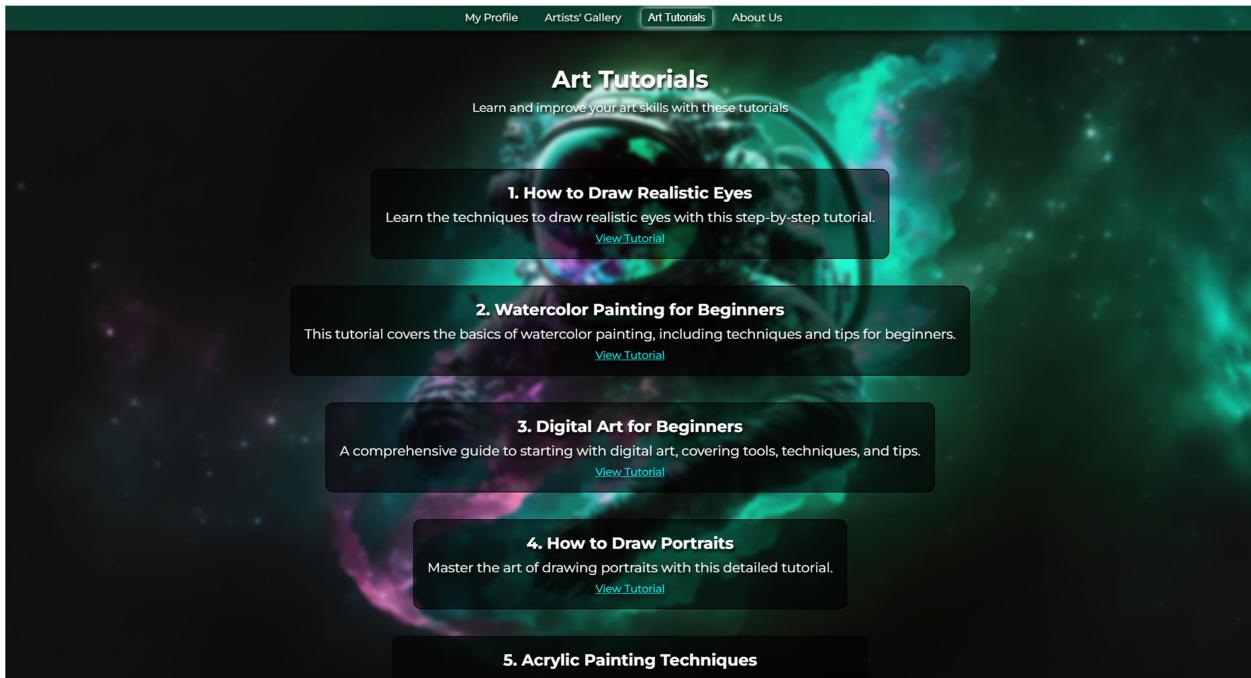
- **Description:** Users can upload new art pieces, which are displayed on the homepage.
- **Screenshot:**



- Art Upload Form:
- Displayed Artwork:
- **Objects Used:** Forms, File Reader, Local Storage.

Feature 03: Art Tutorials

- **Description:** A collection of art tutorials accessible to all users.
- **Screenshot:**



- **Objects Used:** HTML Links, Sections.

Maintenance

Data Backup and Recovery

- User data is stored in local storage, ensuring persistence across sessions.
- Artworks and user profiles are saved locally, with an option to backup data manually.

Overall Conclusion

Objectives Achieved

- Created a user-friendly platform for artists.
- Implemented user authentication and profile management.

- Enabled users to upload and display artworks.
- Provided a variety of art tutorials.

Outlook

- Further improvements could include enhancing the design, adding more interactive tutorials, and implementing server-side storage for increased security.

CODE SCRIPT REVIEW

Loading posts from the local storage :

```

1 loadPostsFromLocalStorage()
2
3 function loadPostsFromLocalStorage() {
4     const posts = JSON.parse(localStorage.getItem('posts')) || [];
5     posts.forEach(post => {
6         displayPost(post.description, post.imageData);
7     });
8 }
9 function displayPost(description, imageData) {
10     const postsContainer = document.getElementById('postsContainer');
11     const postElement = document.createElement('div');
12     postElement.className = 'container'; // Add class to the div
13     postElement.innerHTML =
14
15     `![User Art](${imageData})`;
16     `

${description}

`;
17     ``;
18     postsContainer.appendChild(postElement);
19 }
```

Login script :

(If the login done correctly the user name , role , password saves to the session storage)

```
1  document.getElementById('loginForm').addEventListener('submit', function(event) {
2      event.preventDefault(); // Prevent form submission
3
4      // Get input values
5      var username = document.getElementById('username').value;
6      var password = document.getElementById('password').value;
7
8      // Check if username and password are valid (dummy validation for demonstration)
9      var userData = localStorage.getItem(username);
10     const userObject = JSON.parse(userData);
11     if (userData) {
12         userData = JSON.parse(userData); // Parse JSON string back to object
13         if (userData.password === password) {
14             alert('Login successful! Redirecting...');
15
16             // Save username and password to session storage
17             sessionStorage.setItem('username', username);
18             sessionStorage.setItem('password', password);
19             sessionStorage.setItem('role', userObject.role);
20
21             // Redirect based on user role
22             if (userData.role === 'Artist') {
23                 window.location.href = 'artist.html'; // Redirect to artist page
24             } else {
25                 window.location.href = 'homepage.html'; // Redirect to user page
26             }
27         } else {
28             alert('Invalid password. Please try again.');
29         }
30     } else {
31         alert('Invalid username. Please try again.');
32     }
33 });
34 }
```

The script for saving posts to the local storage and deleting them :

```
1  document.addEventListener('DOMContentLoaded', function() {
2      // Load posts from local storage on page load
3      loadPostsFromLocalStorage();
4
5      const imageUpload = document.getElementById('imageUpload');
6      const descriptionInput = document.getElementById('description');
7
8
9      imageUpload.addEventListener('change', function() {
10         if (descriptionInput.value.trim() === '') {
11             alert('Please enter a description before uploading an image.');
12             return; // Exit the function if description is empty
13         }
14
15         if (imageUpload.files && imageUpload.files[0]) {
16             const reader = new FileReader();
17             reader.onload = function(e) {
18                 // Get image data and description
19                 const imageData = e.target.result;
20                 const description = descriptionInput.value;
21
22                 // Save the post to Local storage
23                 savePostToLocalStorage(description, imageData);
24
25                 // Display the post
26                 displayPost(description, imageData);
27
28                 // Clear the input fields
29                 descriptionInput.value = '';
30                 imageUpload.value = '';
31             }
32             reader.readAsDataURL(imageUpload.files[0]);
33         });
34     });
35 });
36
37 function savePostToLocalStorage(description, imageData) {
38     const posts = JSON.parse(localStorage.getItem('posts')) || [];
39     posts.push({ description, imageData });
40     localStorage.setItem('posts', JSON.stringify(posts));
41 }
42
43 function loadPostsFromLocalStorage() {
44     const posts = JSON.parse(localStorage.getItem('posts')) || [];
45     posts.forEach(post => {
46         displayPost(post.description, post.imageData);
47     });
48 }
49
50 function displayPost(description, imageData) {
51     const postsContainer = document.getElementById('postsContainer');
52     const postElement = document.createElement('div');
53     postElement.className = 'container'; // Add class to the div
54
55     postElement.innerHTML = `
56         
57         <p>${description}</p>
58         <button class="delete-button">
59             <svg viewBox="0 0 448 512" class="svgIcon">
60                 <path d="M135.2 17.7L128 32H32C14.3 32 0 46
61                     .3 0 64S14.3 96 32 96H416c17.7 0 32-14.3 32
62                     -32s-14.3-32-32H320l-7.2-14.3C307.4 6.8 2
63                     06.3 0 284.2 0H163.8c-12.1 0-23.2 6.8-28.6 17.
64                     7zM416 128H32L53.2 467c1.6 25.3 22.6 45 47.9 45H
65                     346.9c25.3 0 46.3-19.7 47.9-45L416 128z"></path>
66         </svg>
67     </button>
68     `;
69
70     const deleteButton = postElement.querySelector('.delete-button');
71     deleteButton.addEventListener('click', function() {
72         // Remove the post element from the DOM
73         postsContainer.removeChild(postElement);
74
75         // Remove the post from local storage
76         removePostFromLocalStorage(description, imageData);
77     });
78
79     postsContainer.appendChild(postElement);
80 }
81
82 function removePostFromLocalStorage(description, imageData) {
83     let posts = JSON.parse(localStorage.getItem('posts')) || [];
84     posts = posts.filter(post => post.description !== description || post.imageData !== imageData);
85     localStorage.setItem('posts', JSON.stringify(posts));
86 }
```

And finaly the script for saving new users to the local storage :

```
1 // Function to handle user sign up
2 function signupUser(username, password, role) {
3     // Dummy validation (for demonstration)
4     if (username && password && role) {
5         // Check if the username already exists in LocalStorage
6         if (localStorage.getItem(username)) {
7             alert('Username already exists. Please choose a different username.');
8         } else {
9             // Store new user data in LocalStorage
10            localStorage.setItem(username, JSON.stringify({ password: password, role: role }));
11            alert('Sign up successful! Please login with your new credentials.');
12        }
13    } else {
14        alert('Please fill in all fields to sign up.');
15    }
16 }
17
18 // Event Listener for signup form submission
19 document.getElementById('signupForm').addEventListener('submit', function(event) {
20     event.preventDefault(); // Prevent form submission
21
22     // Get input values
23     var newUsername = document.getElementById('newUsername').value;
24     var newPassword = document.getElementById('newPassword').value;
25     var selectedRole = document.getElementById('userRole').value; // Get selected role from <select> element
26
27     // Call signupUser function with input values
28     signupUser(newUsername, newPassword, selectedRole);
29
30     // Reset the form after signup
31     document.getElementById('signupForm').reset();
32 });
33
```