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 "Muse hube" 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;

Presentation of the Mini-Project

This mini-project aims to create a static website with both artists 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 Artists to post new art pieces and store them locally.

Objectives

- To provide a platform for artists and art enthusiasts to connect.
- To allow users to manage their profiles.
- To enable artists 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.

Interface Artist

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

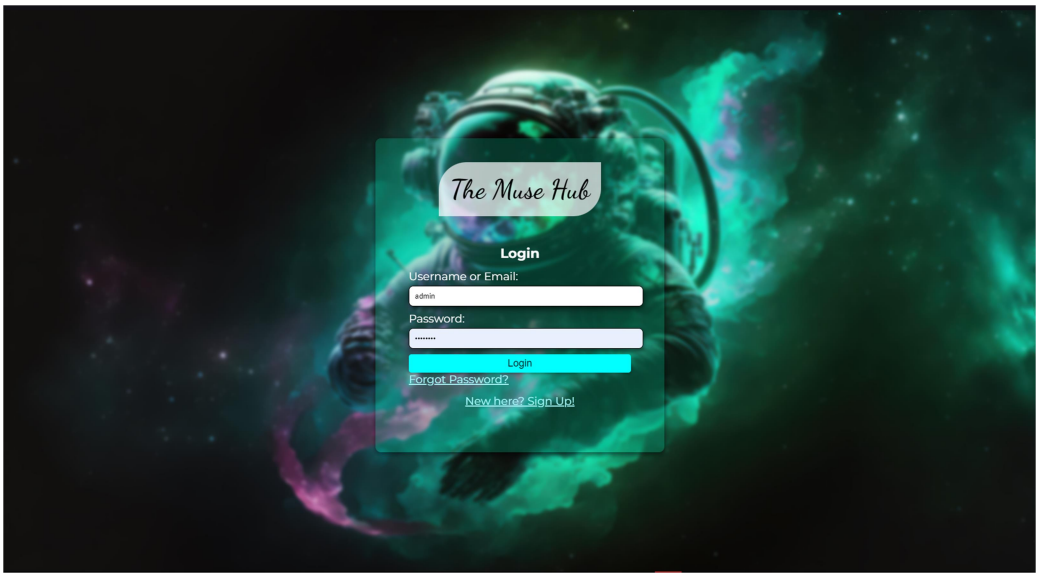
Interface User

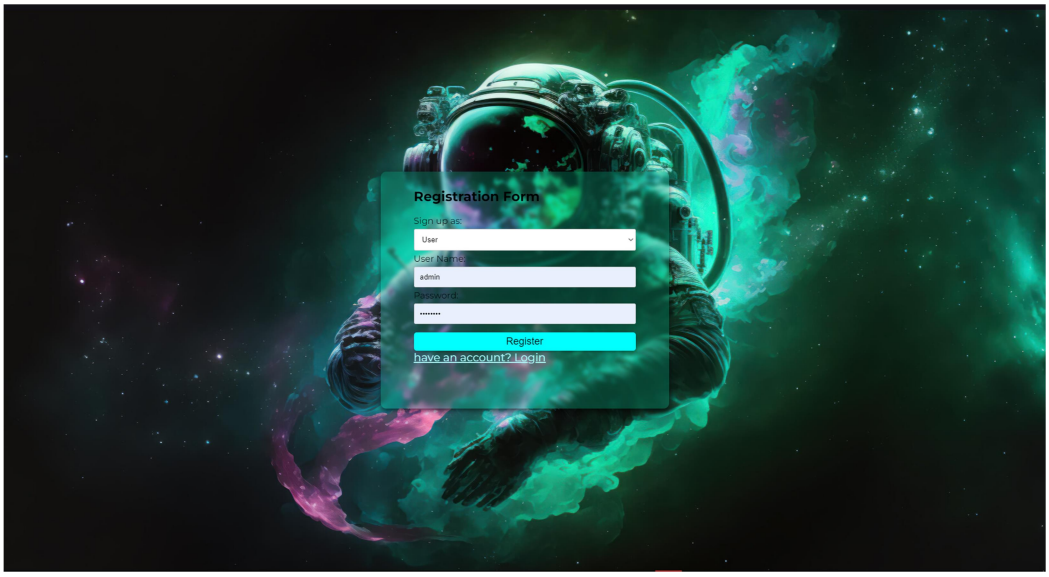
User access is limited to managing their profileand accessing tutorials.

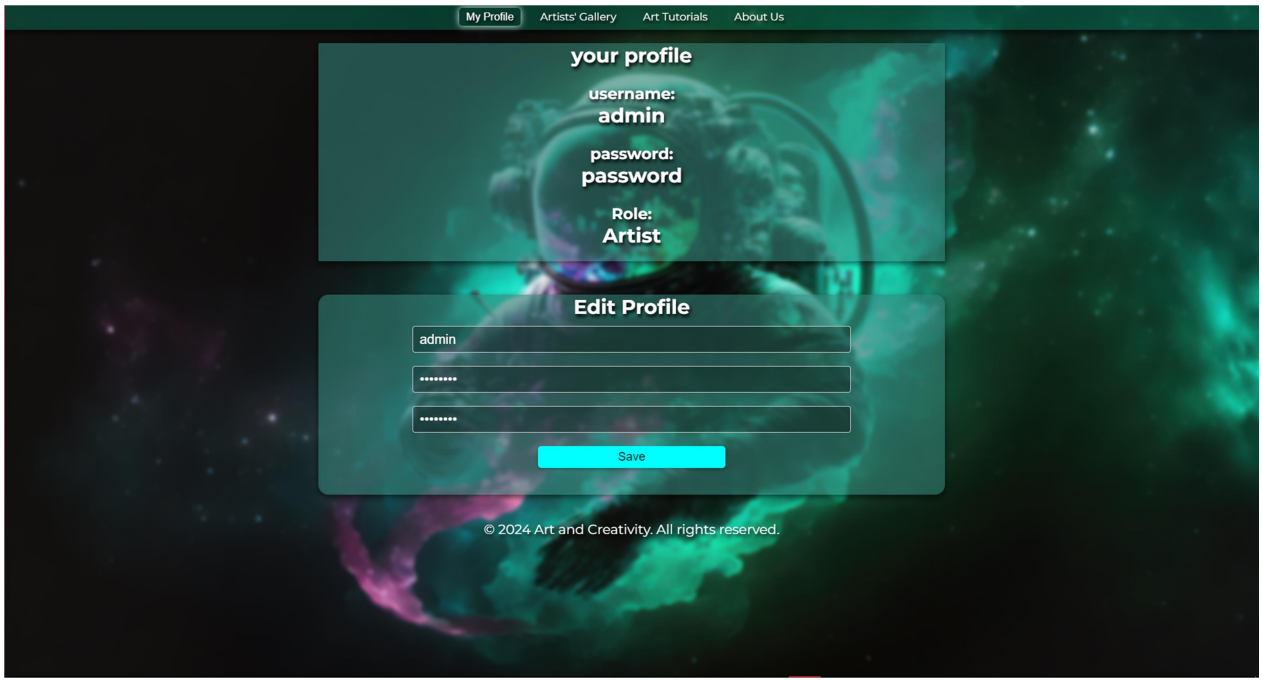
Website Features

Feature 01: User Authentication

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



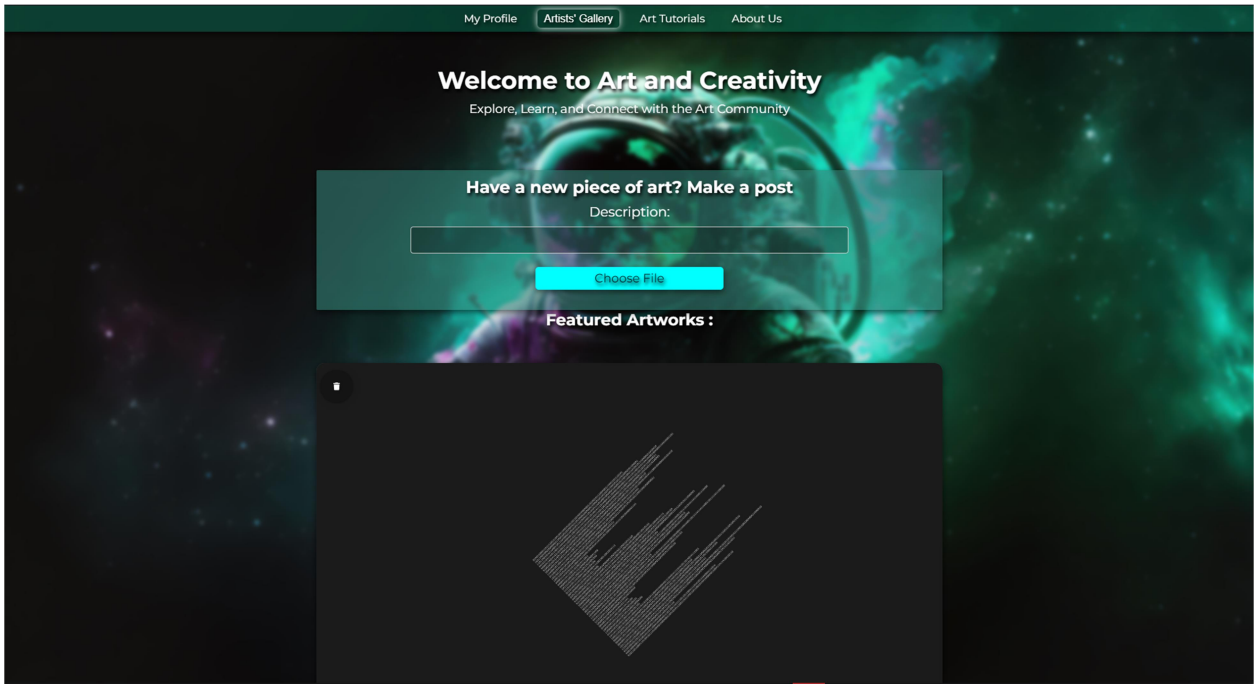




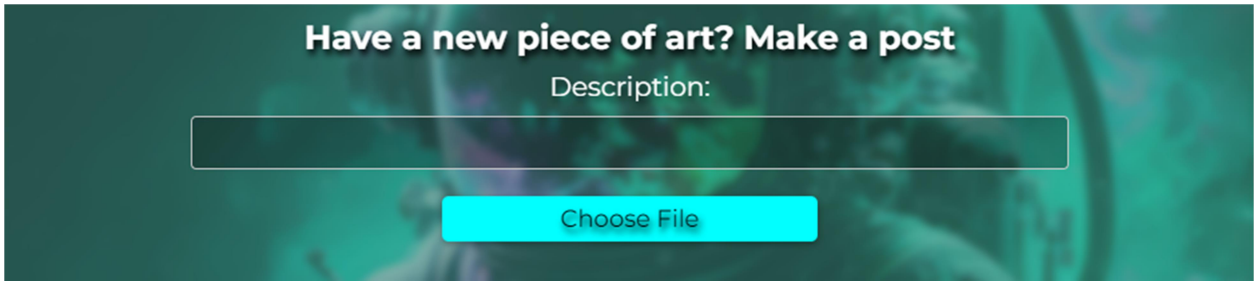
- **Objects Used:** Forms, Local Storage, Session Storage.

Feature 02: Art Upload and Display

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



- Art Upload Form:

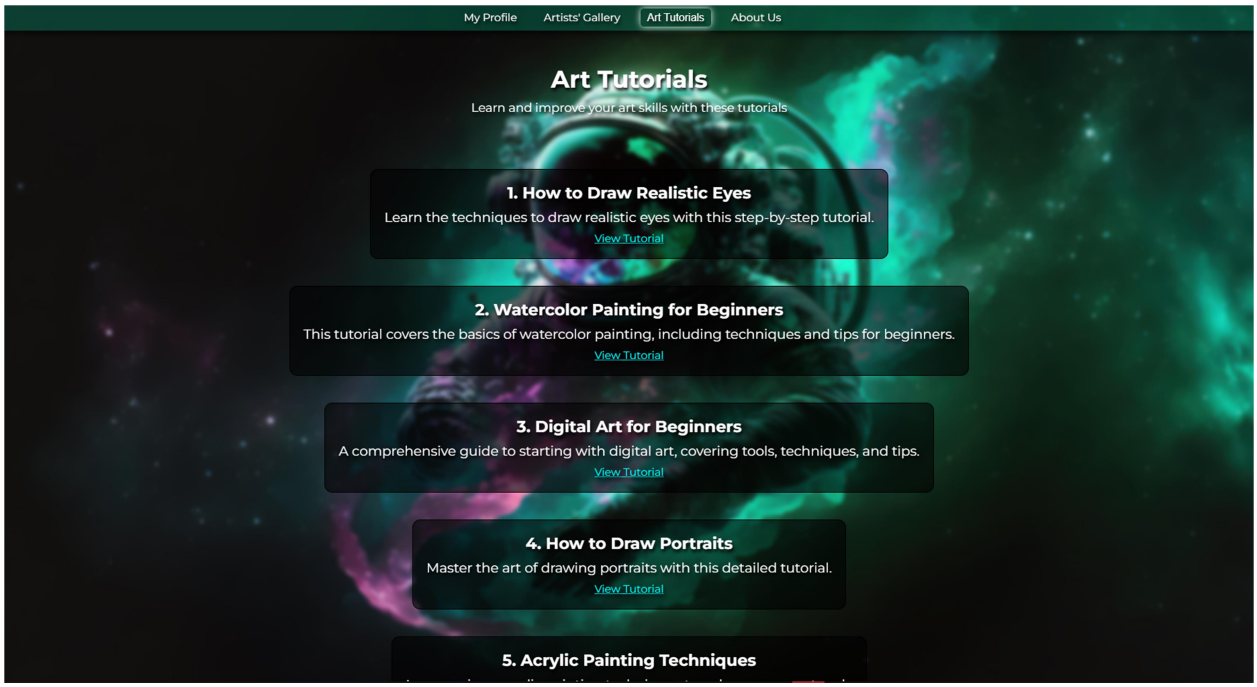


- Displayed Artwork:

- **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,divs.

Overall Conclusion

Objectives Achieved

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

```
loadPostsFromLocalStorage()

function loadPostsFromLocalStorage() {
    const posts = JSON.parse(localStorage.getItem('posts')) || [];
    posts.forEach(post => {
        displayPost(post.description, post.imageData);
    });
}
function displayPost(description, imageData) {
    const postsContainer = document.getElementById('postsContainer');
    const postElement = document.createElement('div');
    postElement.className = 'container';  // Add class to the div
    postElement.innerHTML = `

    <img src="${imageData}" alt="User Art" style="max-width: 100%; height: auto;">
    <p>${description}</p>
    `;
    postsContainer.appendChild(postElement);
}
```

Login script :

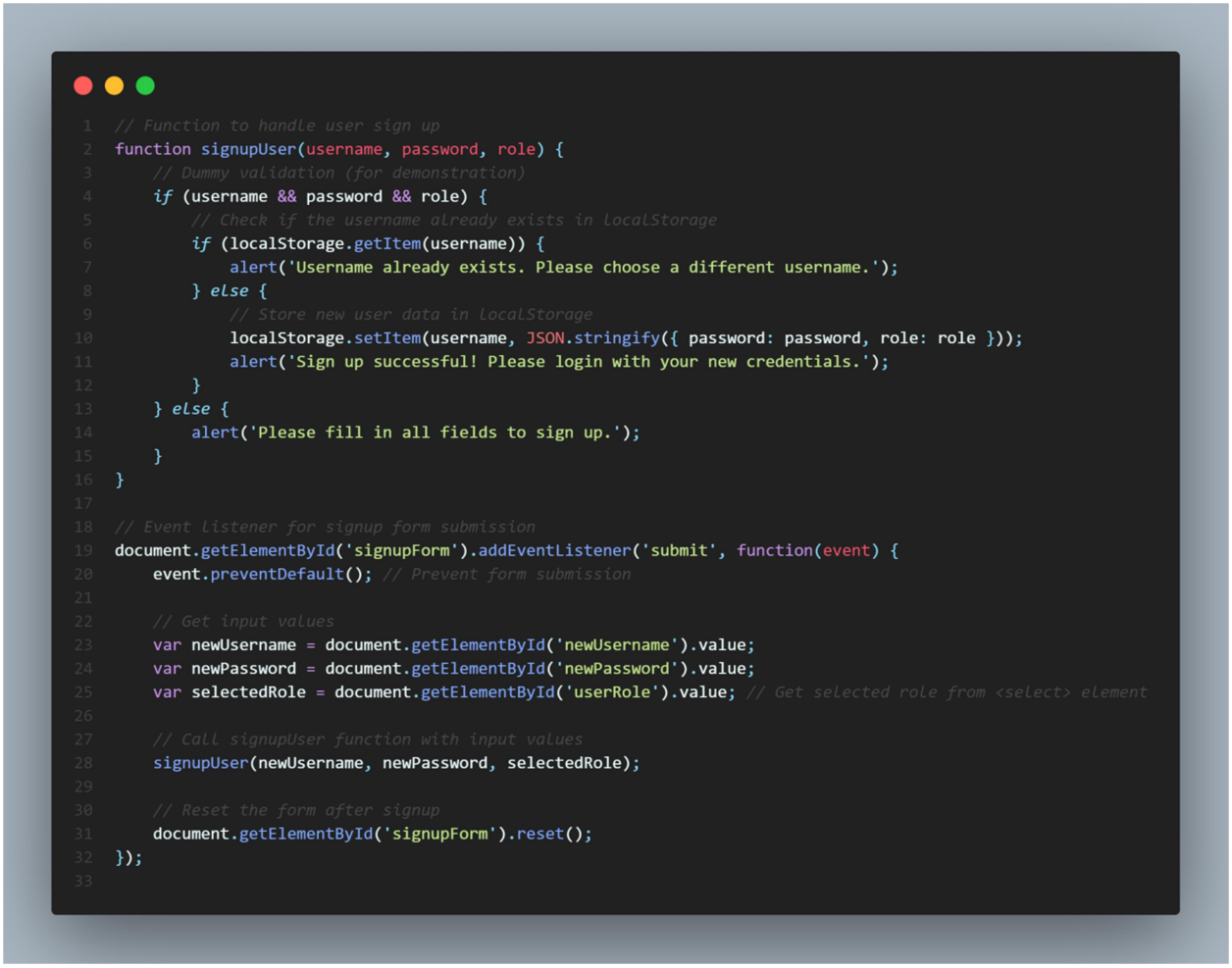
(If the login done correctly the user name , role , password saves to the session storage)

```
document.getElementById('loginForm').addEventListener('submit', function(event) {
    event.preventDefault(); // Prevent form submission

    // Get input values
    var username = document.getElementById('username').value;
    var password = document.getElementById('password').value;

    // Check if username and password are valid (dummy validation for demonstration)
    var userData = localStorage.getItem(username);
    const userObject = JSON.parse(userData);
    if (userData) {
        userData = JSON.parse(userData); // Parse JSON string back to object
        if (userData.password === password) {
            alert('Login successful! Redirecting...');

            // Save username and password to session storage
            sessionStorage.setItem('username', username);
            sessionStorage.setItem('password', password);
            sessionStorage.setItem('role', userObject.role);

            // Redirect based on user role
            if (userData.role === 'Artist') {
                window.location.href = 'artist.html'; // Redirect to artist page
            } else {
                window.location.href = 'homepage.html'; // Redirect to user page
            }
        } else {
            alert('Invalid password. Please try again.');
        }
    } else {
        alert('Invalid username. Please try again.');
    }
});

```

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

```
ocument.addEventListener('DOMContentLoaded', function() {
    // Load posts from local storage on page load
    loadPostsFromLocalStorage();

    const imageUpload = document.getElementById('imageUpload');
    const descriptionInput = document.getElementById('description');


    imageUpload.addEventListener('change', function() {
        if (descriptionInput.value.trim() === '') {
            alert('Please enter a description before uploading an image.');
            return; // Exit the function if description is empty
        }

        if (imageUpload.files && imageUpload.files[0]) {
            const reader = new FileReader();
            reader.onload = function(e) {
                // Get image data and description
                const imageData = e.target.result;
                const description = descriptionInput.value;

                // Save the post to local storage
                savePostToLocalStorage(description, imageData);

                // Display the post
                displayPost(description, imageData);

                // Clear the input fields
                descriptionInput.value = '';
                imageUpload.value = '';
            }
            reader.readAsDataURL(imageUpload.files[0]);
        }
    });
});

function savePostToLocalStorage(description, imageData) {
    const posts = JSON.parse(localStorage.getItem('posts')) || [];
    posts.push({ description, imageData });
    localStorage.setItem('posts', JSON.stringify(posts));
}

function loadPostsFromLocalStorage() {
    const posts = JSON.parse(localStorage.getItem('posts')) || [];
    posts.forEach(post => {
        displayPost(post.description, post.imageData);
    });
}

function displayPost(description, imageData) {
    const postsContainer = document.getElementById('postsContainer');
    const postElement = document.createElement('div');
    postElement.className = 'container';  // Add class to the div

    postElement.innerHTML = `
    <img src="${imageData}" alt="User Art" style="max-width: 100%; height: auto;">
    <p>${description}</p>
        <button class="delete-button">
  <svg viewBox="0 0 448 512" class="svgIcon">
  <path d="M135.2 17.7L128 32H32C14.3 32 0 46
  .3 0 64S14.3 96 32 96H416c17.7 0 32-14.3 32
  -32s-14.3-32-32-32H320l-7.2-14.3C307.4 6.8 2
  96.3 0 284.2 0H163.8c-12.1 0-23.2 6.8-28.6 17.
  7zM416 128H32L53.2 467c1.6 25.3 22.6 45 47.9 45H
  346.9c25.3 0 46.3-19.7 47.9-45L416 128z"></path>
  </svg>
</button>
    `;

    const deleteButton = postElement.querySelector('.delete-button');
    deleteButton.addEventListener('click', function() {
        // Remove the post element from the DOM
        postsContainer.removeChild(postElement);

        // Remove the post from local storage
        removePostFromLocalStorage(description, imageData);
    });

    postsContainer.appendChild(postElement);
}

function removePostFromLocalStorage(description, imageData) {
    let posts = JSON.parse(localStorage.getItem('posts')) || [];
    posts = posts.filter(post => post.description !== description || post.imageData !== imageData);
    localStorage.setItem('posts', JSON.stringify(posts));
}
```

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



```
// Function to handle user sign up
function signupUser(username, password, role) {
    // Dummy validation (for demonstration)
    if (username && password && role) {
        // Check if the username already exists in localStorage
        if (localStorage.getItem(username)) {
            alert('Username already exists. Please choose a different username.');
        } else {
            // Store new user data in localStorage
            localStorage.setItem(username, JSON.stringify({ password: password, role: role }));
            alert('Sign up successful! Please login with your new credentials.');
        }
    } else {
        alert('Please fill in all fields to sign up.');
    }
}

// Event listener for signup form submission
document.getElementById('signupForm').addEventListener('submit', function(event) {
    event.preventDefault(); // Prevent form submission

    // Get input values
    var newUsername = document.getElementById('newUsername').value;
    var newPassword = document.getElementById('newPassword').value;
    var selectedRole = document.getElementById('userRole').value; // Get selected role from <select> element

    // Call signupUser function with input values
    signupUser(newUsername, newPassword, selectedRole);

    // Reset the form after signup
    document.getElementById('signupForm').reset();
});
```