

Documentació Gestió d'Usuaris

Aquest subsistema es dedica a la gestió d'usuaris, incloent-hi la creació, edició i eliminació, accedint amb un login. Els usuaris poden tenir diferents tipus de permisos, per exemple, alguns poden tenir permís per a modificar i crear nous usuaris, mentre que altres no.

Continuarem amb l'explicació de cada pàgina d'aquest subsistema.

index.html

Aquest codi HTML crea una pàgina d'inici de sessió per a la gestió d'usuaris, amb validació en els camps de correu electrònic i contrasenya. A continuació, es detalla cada part del codi.

1. Capçalera (<head>):

- **<!DOCTYPE html>**: Especifica que el document és HTML5.
- **<html lang="en">**: Indica que el contingut està en anglès.
- **<meta charset="UTF-8">**: Defineix la codificació de caràcters com UTF-8 per a admetre la majoria de caràcters i símbols.
- **<meta name="viewport" content="width=device-width, initial-scale=1.0">**: Configura la pàgina perquè s'adapte a dispositius mòbils.
- **<title>**: Assigna el títol "Gestió Usuaris" a la pàgina.
- **<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384...">**: Importa Bootstrap per a un disseny visual i un estil responsive. Inclou un hash d'integritat i l'atribut **crossorigin** per a seguretat.
- **<script src="index.js" defer></script>**: Enllaça un arxiu JavaScript anomenat **index.js** per a la gestió de la lògica de la pàgina. L'atribut **defer** assegura que el script s'execute després que es carregue l'HTML.

2. Contenedor Principal (<div class="container">):

- **<h1>Gestió Usuaris</h1>**: Títol de la pàgina.

3. Formulari d'Inici de Sessió (<form>):

- Crea un formulari que conté els camps d'entrada i un botó per a iniciar sessió.

4. Camp Correu Electrònic (<input type="email">):

- **Etiqueta (<label for="usuari" class="col-sm-2 col-form-label">):** Etiqueta "Correu:" per al camp de correu electrònic.
- **Input (<input type="email">):** Un camp d'entrada de tipus **email** amb un **id="correu"**.
- **Atributs de Validació:**
 - **pattern="^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}\$"**: Expressió regular que valida el format de correu electrònic.
 - **required**: Fa que aquest camp siga obligatori.

5. Camp Contrasenya (<input type="password">):

- **Etiqueta** (`<label for="pw" class="col-sm-2 col-form-label">`): Etiqueta "Contrasenya:" per al camp de contrasenya.
- **Input** (`<input type="password">`): Un camp d'entrada de tipus `password` amb `id="pw"`.
- **Atributs de Validació:**
 - **pattern** (`="^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[!@#$%^&*]) [A-Za-z\d!@#$%^&*]{8,}$"`): Valida que la contrasenya continga almenys una lletra majúscula, una minúscula, un número i un caràcter especial, amb una longitud mínima de 8 caràcters.
 - **required**: Fa que el camp de contrasenya siga obligatori.

6. Missatge d'Error (`<h5 id="missatgeError" class="text-danger">`):

- Un missatge d'error opcional, identificat amb el `id="missatgeError"` i estilitzat en roig (`text-danger`). Aquest element pot ser utilitzat per `index.js` per a mostrar missatges d'error a l'usuari si les dades introduïdes són incorrectes.

7. Botó d'Inici de Sessió (`<button id="btnLogin" class="btn btn-primary">Iniciar sessió</button>`):

- Un botó amb el text "Iniciar sessió" i estils de Bootstrap (`btn btn-primary`) que envia el formulari.

index.js

Aquest codi JavaScript implementa la funcionalitat de validació i gestió d'inici de sessió per a una pàgina d'usuari. S'utilitzen diverses funcions per validar els camps de correu electrònic i contrasenya, gestionar errors i verificar l'usuari en el sistema.

1. `window.onload = init;`

- Assigna la funció `init` per a ser executada quan la pàgina carregue completament.

2. `init()`:

- **Descripció**: Aquesta funció inicia l'aplicació, afegint un event listener al botó "Iniciar sessió" que crida a la funció `login` quan es fa clic.
- **Codi**:

```
function init() {  
    document.getElementById("btnLogin").addEventListener("click",  
    login);  
}
```

3. `validarCorreu()`:

- **Descripció**: Valida el camp de correu electrònic utilitzant una expressió regular i la funció `checkValidity` d'HTML. Mostra un missatge d'error si no és vàlid.
- **Paràmetres**: No té.
- **Retorn**: Retorna `true` si el correu és vàlid, `false` en cas contrari.

- **Codi:**

```
function validarCorreu() {
  const pattern = RegExp(
    /^[a-zA-Z0-9._%+\-]+\@[a-zA-Z0-9.\-]+\.[a-zA-Z]{2,}$/
  );
  const correu = document.getElementById("correu");

  if (!correu.checkValidity()) {
    if (correu.validity.valueMissing) error(correu, "Ompli el teu usuari!");
    if (correu.validity.patternMismatch)
      error(correu, "L'usuari no està be!");
  } else if (pattern.test(correu.value)) return true;

  return false;
}
```

4. **validarPw():**

- **Descripció:** Valida el camp de contrasenya amb una expressió regular que requereix almenys una lletra majúscula, una minúscula, un número, un caràcter especial i una longitud mínima de 8 caràcters.
- **Paràmetres:** No té.
- **Retorn:** Retorna **true** si la contrasenya és vàlida, **false** en cas contrari.
- **Codi:**

```
function validarPw() {
  const pattern = RegExp(
    /^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[!@#$%^&*])[A-Za-z\d!@#$%^&]*
    {8,}$/
  );
  const pw = document.getElementById("pw");

  if (!pw.checkValidity()) {
    if (pw.validity.valueMissing) error(pw, "Ompli el teu usuari!");
    if (pw.validity.patternMismatch) error(pw, "L'usuari no està be!");
  } else if (pattern.test(pw.value)) return true;

  return false;
}
```

5. **error(element, missatge):**

- **Descripció:** Mostra un missatge d'error en un element HTML i enfoca el camp amb error.
- **Paràmetres:**
- **element:** L'element HTML on es mostrarà l'error.

- **missatge**: El missatge d'error que es mostrarà.
- **Codi**:

```
function error(element, missatge) {  
  const textError = document.createTextNode(missatge);  
  const elementError = document.getElementById("missatgeError");  
  elementError.appendChild(textError);  
  element.classList.add("error");  
  element.focus();  
}
```

6. **esborrarError()**:

- **Descripció**: Elimina els missatges d'error i la classe **error** de tots els camps del formulari.
- **Codi**:

```
function esborrarError() {  
  let formulari = document.forms[0].elements;  
  for (let ele of formulari) {  
    ele.classList.remove("error");  
  }  
  document.getElementById("missatgeError").replaceChildren();  
}
```

7. **login(e)**:

- **Descripció**: Aquesta funció comprova les dades d'usuari introduïdes amb els usuaris guardats en **localStorage**. Si el correu i la contrasenya són correctes, guarda l'identificador de perfil d'usuari a **localStorage** i redirigeix a la pàgina "llistatUsuaris.html".
- **Paràmetres**:
- **e**: Event de clic del botó d'inici de sessió.
- **Codi**:

```
function login(e) {  
  const data = JSON.parse(localStorage.getItem("data")) || { users: [] };  
  const users = data.users;  
  const email = document.getElementById("correu").value;  
  const password = document.getElementById("pw").value;  
  
  esborrarError();  
  e.preventDefault();  
  const user = users.find((u) => u.email == email && u.password == password);  
  
  if (user) {  
    if (validarCorreu() && validarPw()) {
```

```
        localStorage.setItem("currentUserRole", user.user_profile_id);
        location.href = "llistatUsuaris.html";
    }
    return true;
} else {
    error(
        document.getElementById("correu"),
        "Login incorrecte, verifiqueu el correu i la contrasenya."
    );
    return false;
}
}
```