Rapport mini-projet initiation au web.

CTF SHX 

Introduction:

The purpose of this report is to provide an overview and description of a CTF (Capture the Flag) platform designed to facilitate cybersecurity education. The platform offers a wide range of challenges and courses covering various domains, including general skills, cryptography, forensics, web exploitation, binary exploitation, and reverse engineering. This report aims to present the project, highlight its significance, outline the identified problem, and describe the objectives pursued through the development of this website.

Presentation of mini-projet:

The SHX CTF platform is an online learning environment that serves as a hub for aspiring cybersecurity enthusiasts and professionals seeking to enhance their knowledge and skills. It incorporates interactive challenges, practical exercises, and informative courses across multiple cybersecurity domains, providing users with hands-on experiences and theoretical understanding.

Problematic:

In the field of cybersecurity education, individuals often struggle to find a comprehensive platform that offers a diverse range of challenges and courses across multiple domains. Existing resources may be scattered, lacking proper organization, or fail to provide a cohesive learning experience. This fragmented landscape can hinder learners' progress and make it difficult for them to acquire a holistic understanding of cybersecurity concepts and techniques.

Objectives:

The main objectives of this project are as follows:

a. Comprehensive Learning: To create a platform that covers a wide range of cybersecurity domains, offering a comprehensive learning experience for users. By providing challenges and courses in general skills, cryptography, forensics, web exploitation, binary exploitation, and reverse engineering, the platform aims to equip learners with a diverse skill set.

b. Progressive Difficulty: To structure the challenges and courses in a progressive manner, allowing users to start with foundational concepts and gradually advance to more complex scenarios. This approach ensures a smooth learning curve, enabling learners to build upon their knowledge and skills over time.

c. Hands-on Experience: To provide practical exercises and interactive challenges that encourage users to apply their theoretical knowledge in real-world scenarios. This hands-on approach fosters active learning and enhances problem-solving abilities.

d. Accessibility: To design the platform with a user-friendly interface and intuitive navigation, making it accessible to users with varying levels of technical expertise. This inclusivity allows beginners to comfortably start their cybersecurity journey and enables experienced individuals to delve deeper into specific areas of interest.

e. Community Engagement: To foster a vibrant and supportive community where users can collaborate, exchange ideas, and seek guidance from peers and mentors. The platform will offer forums, discussion boards, and other interactive features to facilitate knowledge sharing and networking opportunities.

THE UI :
the main page :

SHX CTF

HELP ABOUT CONTACT SIGN UP

CYBER SECURITY

READ ABOUT CYBER ATTACKES FOR A GOOD START

EXPLORE MORE

YOU WANT TO CHECK OUT OUR CHALLENGES ?

START NOW

PHISHING

Phishing is when attackers send malicious emails designed to trick people into falling for a scam. Typically, the intent is to get users to reveal financial information, system credentials or other sensitive data.

MALWARE

MALWARE

Malware is any software intentionally designed to cause disruption to a computer, server, client, or computer network, leak private information, gain unauthorized access to information or systems, deprive access to information.

HACKING

Hacking refers to the misuse of devices like computers, smartphones, tablets, and networks to cause damage to or corrupt systems, gather information on users, steal data and documents, or disrupt data-related activity.

HACKING

Hacking refers to the misuse of devices like computers, smartphones, tablets, and networks to cause damage to or corrupt systems, gather information on users, steal data and documents, or disrupt data-related activity.

CTF

is an exercise in which "flags" are secretly hidden in purposefully-vulnerable programs or websites. It can either be

CTF

is an exercise in which "flags" are secretly hidden in purposefully-vulnerable programs or websites. It can either be for competitive or educational purposes. Competitors steal flags either from other competitors (attack/defense-style CTFs) or from the organizers (jeopardy-style challenges)

Features

Privacy

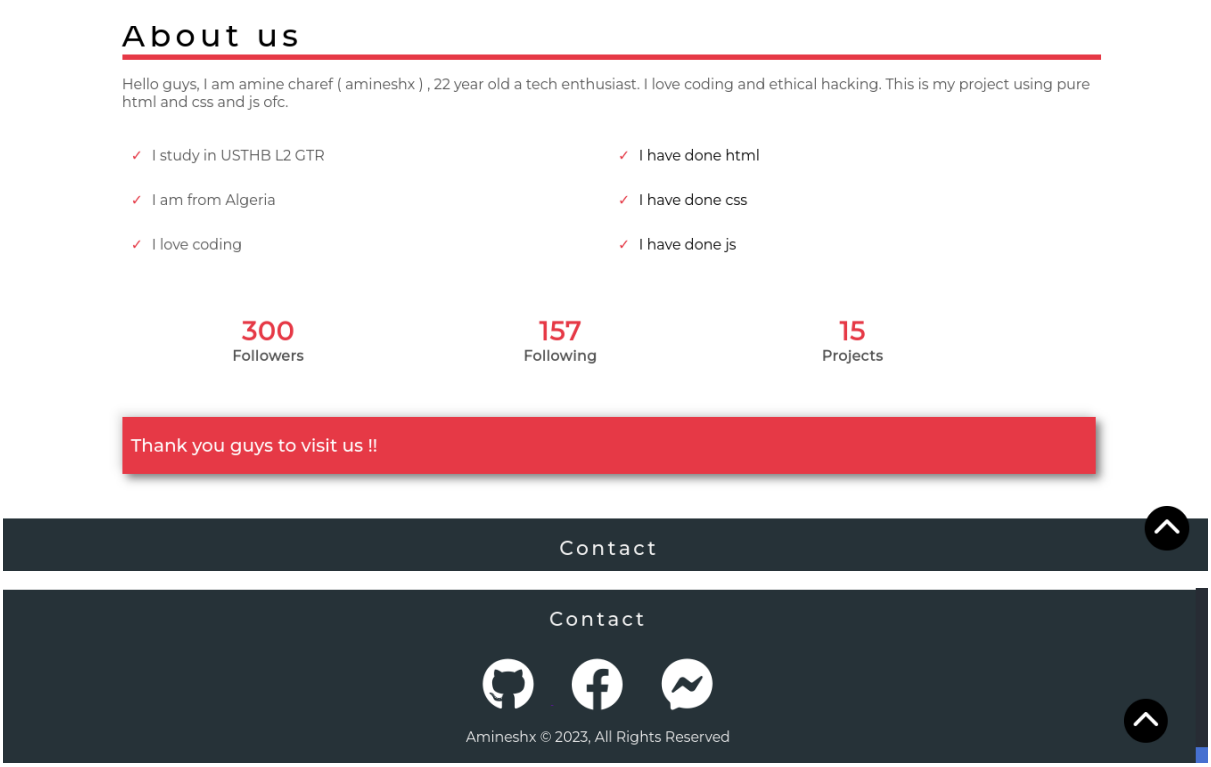
The shx security platform ensures thier user that privacy of users will be kept private. There will nkt be any leakage of data.

Less price

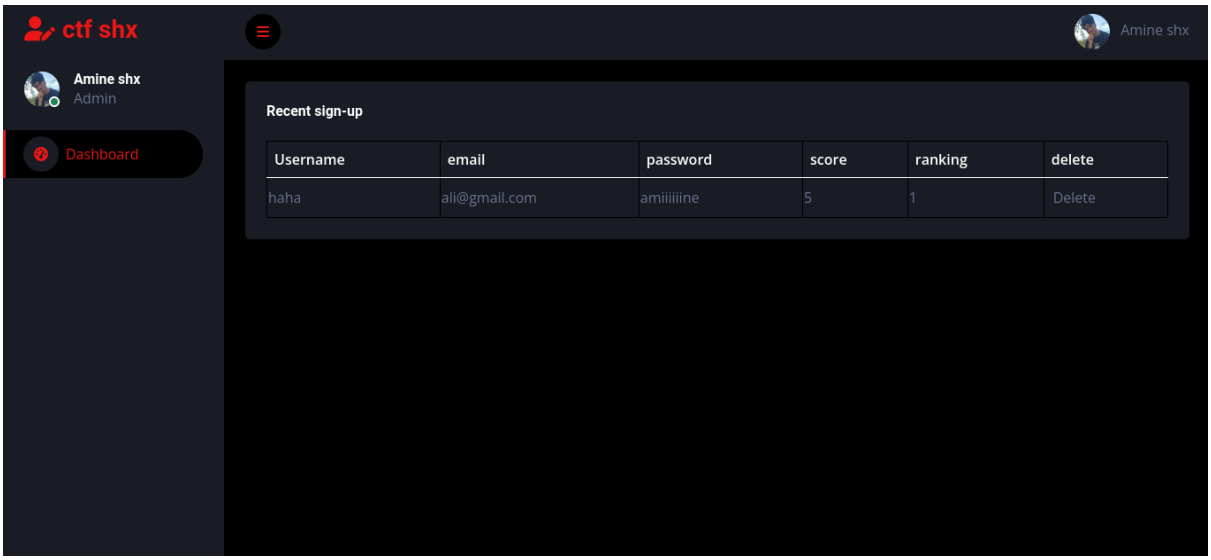
My company provide cheapest service in the world of internet for security solutions. To use primium services You have to pay only $50

Experience

we have experienced ethical hacker, developer, programmer and penetration tester, who will provide you best services.



the Admin side :



for the admin side , admin can see the users object {username, email. pass, score, ranking} and can delete users from local storage
its easy to display that table , u just need to get items from the local storage

and store it on variable with a typeof array of objects by using the JSON.parse method

```
var users = JSON.parse(localStorage.getItem("users")) || [];
```

and now u have the array of users , so now start creating the table and of course u need to have a table tag on your html file

```
<table
  class="table text-start align-middle table-hover mb-0"
>
  <thead>
    <tr class="text-white">
      <th scope="col">Username</th>
      <th scope="col">email</th>
      <th scope="col">password</th>
      <th scope="col">score</th>
      <th scope="col">ranking</th>
      <th scope="col">delete</th>
    </tr>
  </thead>

</table>
```

so now get that table by using `querySelector` `or any method that u want`

```
var table = document.querySelector(".table");
```

sort the that array of users for the ranking before displaying the table by using sort method

```
// Sort users
users.sort(function (a, b) {
    return b.score - a.score;
});
```

and now proceed the table displaying , and by using `forEach` method u can iterate through the array intel the array ends and putting every on a variable `row`

```
// Display users in the table
users.forEach(function (user, index) {

    var row = document.createElement("tr");


    var usernameCell = document.createElement("td");
    usernameCell.textContent = user.username;

    var emailCell = document.createElement("td");
    emailCell.textContent = user.email;

    var passwordCell = document.createElement("td");
    passwordCell.textContent = user.password;

    var scoreCell = document.createElement("td");
    scoreCell.textContent = user.score;

    var rankingCell = document.createElement("td");
    rankingCell.textContent = index + 1;
```

and for the final column of the table there is the delete user button

```
    // Create delete button
    var deleteCell = document.createElement("td");
    var deleteButton = document.createElement("button");
    deleteButton.textContent = "Delete";
    deleteButton.classList.add("btn");
    deleteButton.addEventListener("click", function () {
        deleteUser(index);
    });
    deleteCell.appendChild(deleteButton);
```

and now appending cells to the row

```
        // Append cells to the row
        row.appendChild(usernameCell);
        row.appendChild(emailCell);
        row.appendChild(passwordCell);
        row.appendChild(scoreCell);
        row.appendChild(rankingCell);
        row.appendChild(deleteCell);

        // Append the row to the table
        table.appendChild(row);
    }); // the end of forEach function
```

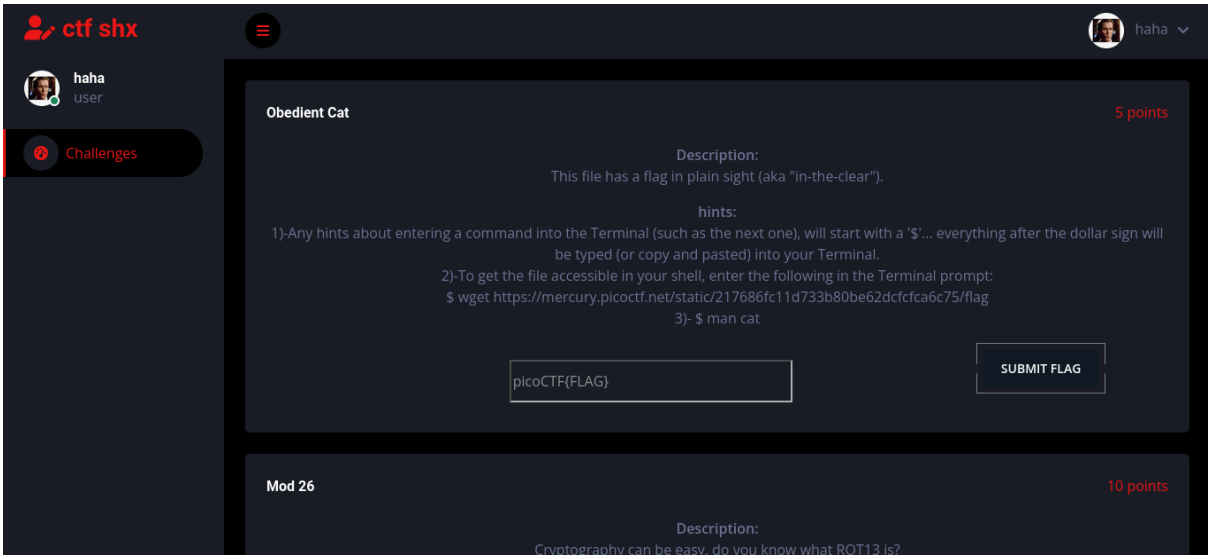
and now creating the deleting of users function , the main part is knowing the location of the user that u want to delete on the array simply getting the index and then just delete it

```
// Function to delete a user
function deleteUser(index) {
    // Remove the user from the users array
    users.splice(index, 1);

    // Update the users in the local storage
    localStorage.setItem("users", JSON.stringify(users));

    // Refresh the page to reflect the changes
    location.reload();
}
```

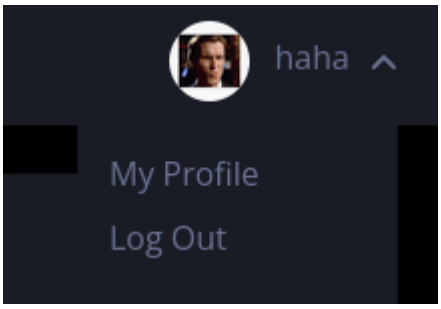
and for the login of the admin :
the username is : amineshx
and the password is : 212131040802

the user side :

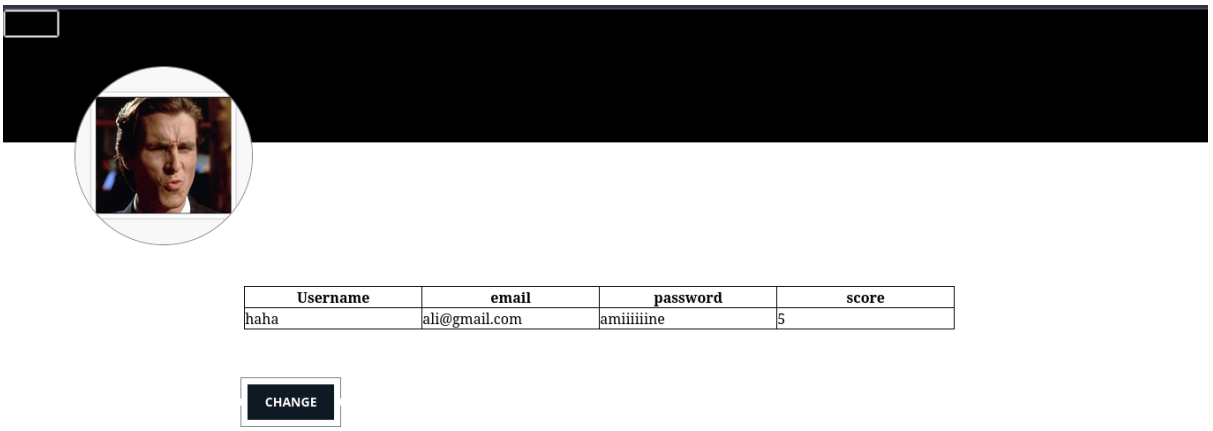
users can learn and play the challenges and they have a profile they can change it as they like
first ofc they can sign up and login after that
this is an example for a user with a **haha** username (sorry for the name lol)



and on the top right he can access his profile or log out his account



and this is the profile which is he can change his profile picture and some themes



to change the profile pic just click on the photo and brows for a picture on your computer and reload the page and the code behind that is

```
15      <div class="profile-pic-div">
16        <img src="" id="photo" />
17        <input type="file" id="file" />
18        <label for="file" id="uploadBtn">Choose Photo</label>
19      </div>
```

first get those elements or use my lovely `querySelector` and stor um in a variable

```
24
25    const imgDiv = document.querySelector('.profile-pic-div');
26    const img = document.querySelector('#photo');
27    const file = document.querySelector('#file');
28    const uploadBtn = document.querySelector('#uploadBtn');
```

and second addEventListener to the variable of the img div when the mouse cursor enter and leave the div , to display and block the label

```
<label for="file" id="uploadBtn">Choose Photo</label>
```

```
imgDiv.addEventListener('mouseenter', function(){
   uploadBtn.style.display = "block";
});

//if we hover out from img div

imgDiv.addEventListener('mouseleave', function(){
   uploadBtn.style.display = "none";
});
```

and now lets get the user object from the local storage and after that

```
//lets work for image showing functionality when we choose an image to
upload

//when we choose a foto to upload

file.addEventListener('change', function(){
   //this refers to file
   const choosedFile = this.files[0];

   if (choosedFile) {

       const reader = new FileReader();
       reader.addEventListener('load', function(){
           foundUser.img = reader.result ;
           users[j] = foundUser;
           localStorage.setItem('users', JSON.stringify(users));

       });

       reader.readAsDataURL(choosedFile);

   }
});
```

```
    document.addEventListener("DOMContentLoaded", ()=>{
        const recentImgeDartaUrl = foundUser.img ;
        if(recentImgeDartaUrl){

document.querySelector("#photo").setAttribute("src",recentImgeDartaUrl)
;
        }
    });
```

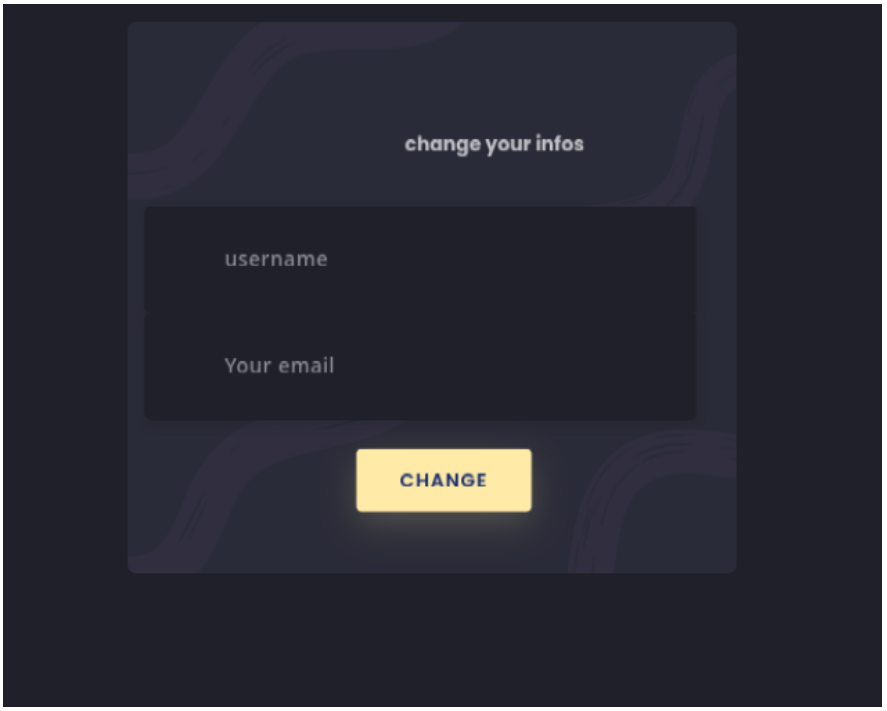
and to store the picture on the local storage u need to have on the user object a data img to store that image

and the users can change their username and email

```
function change() {
    function getCookie(name) {
        const cDecoded = decodeURIComponent(document.cookie);
        const cArray = cDecoded.split("; ");
        console.log(cArray);
        let Username = cArray[0].split("=");
        Username = Username[1];
        return Username
    }
    let usernameToFind = getCookie("loggedInUser");

    var users = JSON.parse(localStorage.getItem('users'));
    var foundUser = null;
    var j = -1; // userindex in the users array objects
    for (var i = 0; i < users.length; i++) {
        if (users[i].username === usernameToFind) {
            foundUser = users[i];
            j = i;
            break;
        }
    }
    let newusername = document.getElementById("newname").value;
    let newemail = document.getElementById("newemail").value;
    foundUser.username = newusername;
    foundUser.email = newemail;
```

```
   users[j] = foundUser;
   localStorage.setItem('users', JSON.stringify(users));
   alert("your data has changed ");
   document.cookie = "loggedInUser=; expires=Thu, 01 Jan 1970 00:00:00
UTC; path=/;";
   document.cookie = 'loggedInUser=' +
encodeURIComponent(foundUser.username) + '; SameSite=None; Secure';
   window.location.href= "profile.html";
}
```

the process is simple just get the cookie name for the user that want to change his profile informations and look up that name on the local storage and get the elements of the inputs of the new username and email and change um



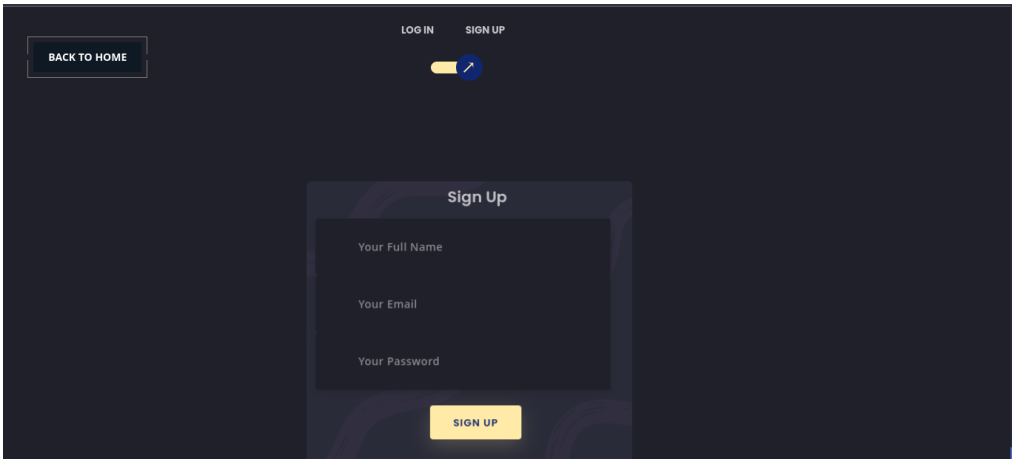

```
            <div class="section text-center">
              <h6 class="mb-4 pb-3">change your infos</h4>
              <div class="form-group">
                <input
                  type="text"
                  name="newname"
                  class="form-style"
                  placeholder="username"
                  id="newname"
                  autocomplete="off"
                />
                <i class="input-icon uil uil-at"></i>
```

```
            </div>
            <div class="form-group mt-2">
              <input
                type="email"
                name="newemail"
                class="form-style"
                placeholder="Your email"
                id="newemail"
                autocomplete="off"
              />
              <i class="input-icon uil uil-lock-alt"></i>
            </div>
            <a class="btn" onclick="change()">change</a>
          </div>
```

Features:

feature1 : (login users and sign up)

users can sign up ofc and any match of username or email of the new user , the logging procedure will be interpreted, every user must have a unique username and email



and it’s simple
first get the elements from the html by id

```
  var username = document.getElementById('signname').value;
  var email = document.getElementById('signmail').value;
  var password = document.getElementById('signpass').value;
```

and get the items or the users from the local storage and change the format to an array by using the `JSON.parse` and store it on variable users

```
var users = JSON.parse(localStorage.getItem('users')) || [];
```

and start checking if the username exist on the users variable therefore if the user exist on the local storage by using the `find` method .

```
var existingUser = users.find(function(user) {
  return user.username === username || user.email === email;
});
```

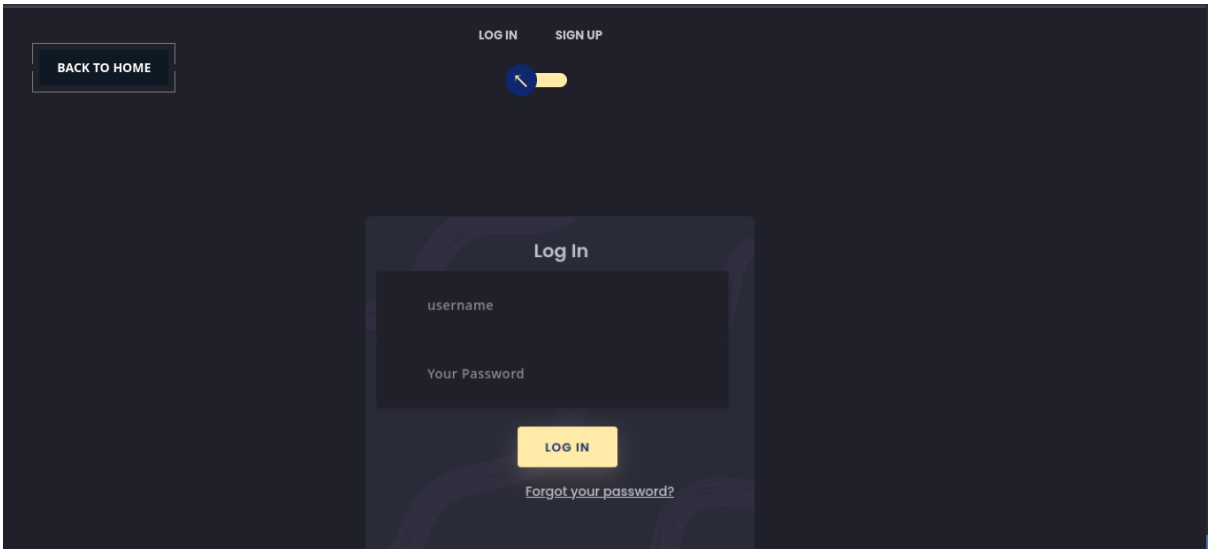
the parameter of the `find` method is a function that returns a boolean that checks if the user.username and user.email exist so now we have a variable existingUser which is an boolean type so

```
if (existingUser) {

  alert('User already exists!');

} else {
  var newUser = {
    username: username,
    email: email,
    password: password,
    score : 0 ,
    img : "umpty"
  };
  users.push(newUser);
  localStorage.setItem('users', JSON.stringify(users));
  alert('User created successfully!');
 }
}
```

as u can see if the existingUser == False we need to set up our new user object and push it the users array (the array that contains the users from the local storage)
and put back our array on the local storage .

for the login is more simple,
first get the elements from the html by id or as u like

```
var username = document.getElementById('logname').value;
var password = document.getElementById('logpass').value;
```



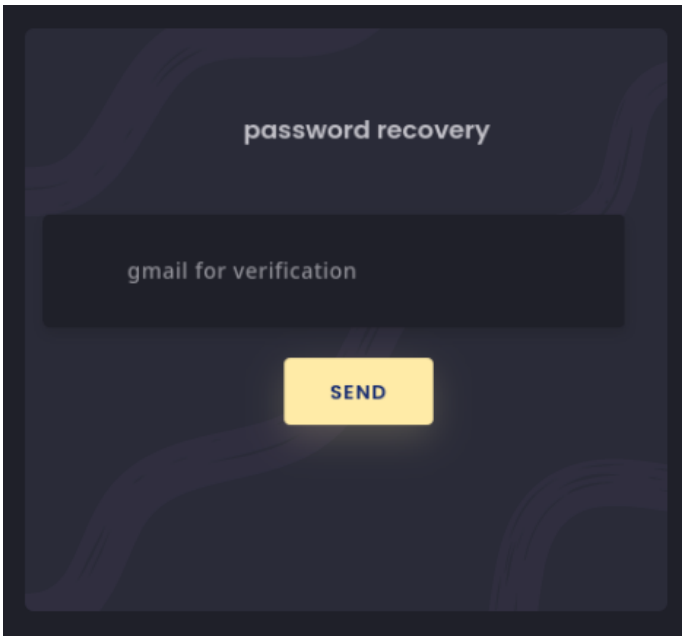

```
// Retrieve users from local storage
var users = JSON.parse(localStorage.getItem('users')) || [];
 // Find the user with matching username and password
var foundUser = users.find(function(user) {
  return user.username === username && user.password === password;
});
```

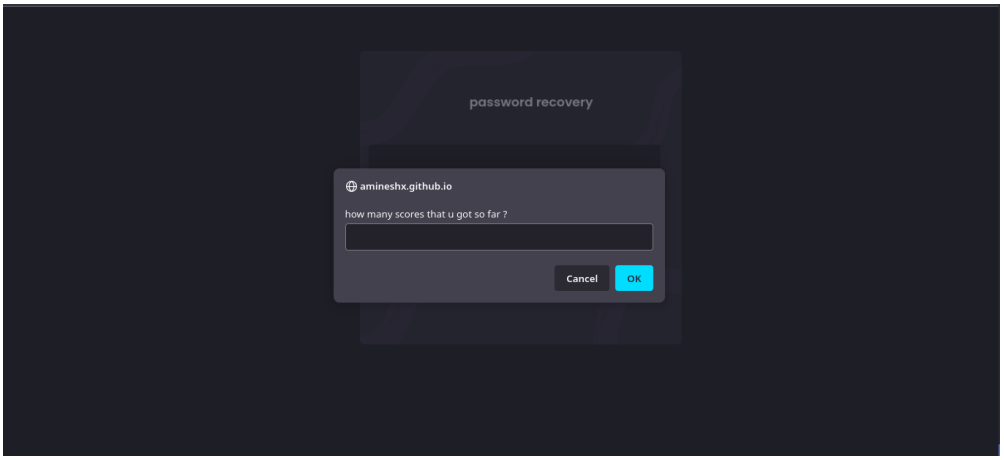
if the username == amineshx and password == 212131040802
that means it's an admin so href the dashboard for admins
else is found user == true the login is successful and create the cookie for the user and take him to the users page .

```
if(username == "amineshx" && password == "212131040802"){
    window.location.href= "/dashboard.html"
}else if (foundUser) {
  // User found, proceed with login logic
  alert('Login successful!');
   // Set a cookie for the session
  document.cookie = 'loggedInUser=' + encodeURIComponent(username) +
'; SameSite=None; Secure';
    window.location.href = "/user.html";
} else {
  alert('Invalid username or password!');
}
}
```

feature2 : (forgetting password while login)

when the user forget his password he may needs to remember his score





if the email exist on the local storage a prompt of submitting the score will show , and then submit your score and u will prompt again to submit your new password

else an admin may delete you .

the code for that is simple,

first retrieve the users from the local storage and then create a variable to store the user who forget his password which is foundUser but first initialize it with Null

```
const verfemail = document.getElementById("verfemail").value;
var users = JSON.parse(localStorage.getItem("users")) || [];
var foundUser = null;
   var j = -1; // userindex in the users array objects
   for (var i = 0; i < users.length; i++) {
```

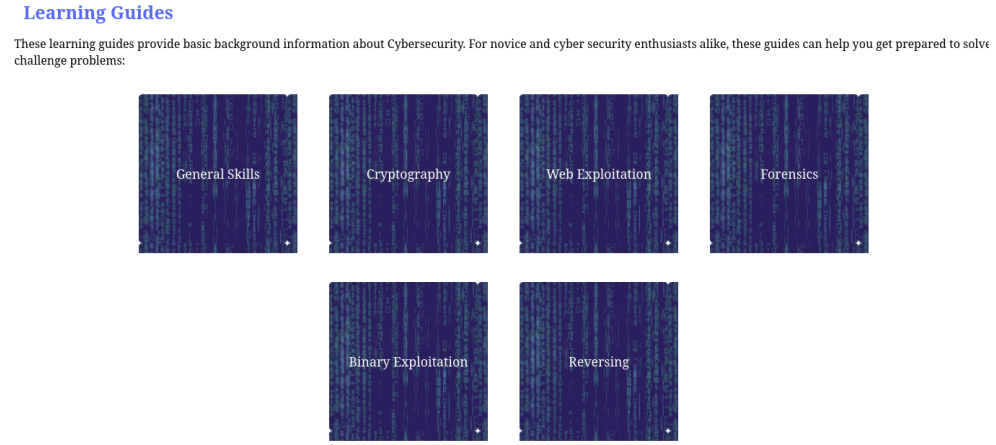
```
        if (users[i].email === verfemail) {
            foundUser = users[i];
            j = i;
            break;
        }
    }
if (foundUser==null) {
        alert("Email not found admin may deleted u");
    }else{
        const verfusername = prompt("how many scores that u got so far
?");
        if (foundUser.score == verfusername){
            const newpass = prompt("its not that professional and secure
website but put your new password here ");
            foundUser.password = newpass ;
            users[j] = foundUser;
            localStorage.setItem('users', JSON.stringify(users));
            alert("your password has changed ");
            window.location.href= "../sign-up.html";
        }
    }

}
```

if the user is found then and verified his score and submitted a new password then : `foundUser.password = newpass ;`

and update the users array and then update the local storage back again .

feature3 : (help)

users can learn before starting the hunt

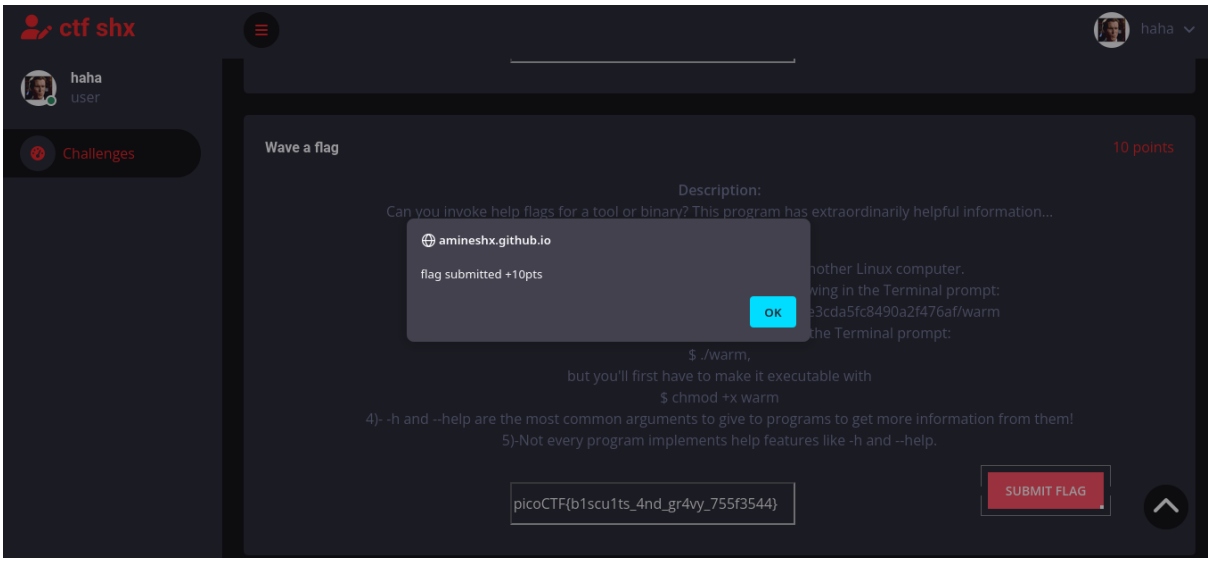


other Features:

i have mention other features on the part of describing the user side and admin side , like user can change their personal infos and changing their profile style , and when it comes to the challenges:
first when the user submitted his flag , i think u know what are the procedures now , getting the user name from the cookie value and retrieve the user from the users array, and check if the flag match the flag given ,

```
    var x = document.getElementById("Wave-a-flag").value;
    const flag4 = "picoCTF{b1scu1ts_4nd_gr4vy_755f3544}";
    if (x == flag4) {
        alert("flag submitted +10pts ");
        foundUser.score = foundUser.score + 10;
        users[j] = foundUser;
        localStorage.setItem('users', JSON.stringify(users));
        var button = document.getElementById("myButton4");
        button.disabled = true;
    } else {
        alert(" no lol  ");
    }
}
```

and if the flags matches add the challenge score to the user.score
and update the user on the users array and store back the infos to the local storage and disable the button (there will be a problem when u refresh the page)



conclusion:

Achieved Objectives:

1. Comprehensive Platform: We have successfully created a platform that offers a wide range of challenges and courses covering various cybersecurity domains. This comprehensive approach ensures that users can explore different areas of interest and develop a diverse skill set.
2. Progressive Learning: The challenges and courses on the platform have been structured in a progressive manner, allowing users to start with foundational concepts and gradually advance to more complex scenarios. This approach ensures a smooth learning curve and enables users to build upon their knowledge and skills over time.
3. Hands-on Experience: The platform provides practical exercises and interactive challenges that encourage users to apply their theoretical knowledge in real-world scenarios. This hands-on approach enhances problem-solving abilities and reinforces the learning process.
4. User-Friendly Interface: The platform has been designed with a user-friendly interface and intuitive navigation, making it accessible to users with varying levels of technical expertise. This inclusivity ensures that beginners can comfortably start their

Future Perspectives (Remaining Objectives):

While we have achieved several objectives in the development of the CTF platform, there are still some areas for future improvement and expansion. These remaining objectives include:

1. Expansion of Domains: Consider adding additional cybersecurity domains, such as network security, mobile security, and cloud security, to provide a more comprehensive learning experience.
2. Advanced Challenges and Courses: Develop more advanced challenges and courses to cater to experienced professionals and provide opportunities for continuous skill development.
3. Gamification Elements: Introduce gamification elements, such as leaderboards, badges, and rewards, to enhance user engagement and motivation.

4. Integration of Real-World Scenarios: Incorporate real-world scenarios and case studies to provide users with practical insights into cybersecurity challenges faced by organizations.
5. Continuous Updates and Maintenance: Regularly update and maintain the platform to ensure the inclusion of the latest cybersecurity trends, technologies, and techniques.
6. teachers: users can became teachers or source providers and thy get paid for any purchased course

By addressing these future perspectives, the CTF platform can further solidify its position as a leading educational resource in the field of cybersecurity, providing users with valuable learning opportunities and fostering their professional growth.