


>  `<div id = "buton_intoarcere_lobbies">`  
`<a href="/lobbies">lobbies</a>`  
`</div>`  
creeaza un buton de intoarcere stilizat in css pentru asta



---

Pentru a crea un buton de întoarcere stilizat în CSS, poți folosi următorul cod:

HTML:

html

 Copy code

```
<div id="buton_intoarcere_lobbies"> <a  
href="/lobbies">lobbies</a> </div>
```

CSS:

css

 Copy code

```
#buton_intoarcere_lobbies { background-color:  
#007bff; border-radius: 5px; padding: 10px  
15px; display: inline-block; text-decoration:  
none; color: #ffffff; font-weight: bold; }  
#buton_intoarcere_lobbies:hover { background-  
color: #0056b3; } #buton_intoarcere_lobbies a  
{ color: #ffffff; text-decoration: none; }
```

În acest exemplu, am folosit un div cu un ID specific pentru a selecta butonul în CSS și să-i aplicăm stilurile corespunzătoare. Am utilizat proprietăți precum `background-color` pentru a seta culoarea de fundal, `border-radius` pentru a rotunji colțurile, `padding` pentru a adăuga spațiu în jurul textului, `display` pentru a se afișa ca un element de tip inline-block și `text-`

**decoration** și **color** pentru a seta stilul și culoarea textului.

Stilurile **:hover** se aplică când utilizatorul plasează cursorul mouse-ului pe buton, oferind un efect vizual de feedback. Am folosit de asemenea stilurile pentru a aplica culoarea și stilul textului în cadrul elementului a (link).

Dacă dorești să personalizezi butonul și mai mult, poți modifica proprietățile CSS după preferințe, cum ar fi mărimea fontului, marginea, umbra și altele.



cum pot sa il ridic putin mai sus fata de partea de jos



Pentru a ridica butonul puțin mai sus față de partea de jos, poți folosi proprietatea CSS **margin-bottom** pentru a adăuga un spațiu suplimentar între buton și partea de jos. Modifică codul CSS astfel:

css

Copy code

```
#buton_intoarcere_lobbies { /* Stilurile  
existente */ margin-bottom: 10px; /*  
Modificarea aici */ }
```

Prin adăugarea valorii **10px** la proprietatea **margin-bottom**, vei crea un spațiu de 10 pixeli între buton și partea de jos. Poți ajusta această valoare în funcție de cât de sus vrei să ridici butonul.

Asigură-te că actualizezi codul HTML și CSS în consecință pentru a vedea schimbările.



si cum schimb textul



Pentru a schimba textul din buton, trebuie să modifici conținutul elementului `<a>` din HTML. Înlocuiește textul existent cu noul text dorit. De exemplu, dacă dorești să schimbi textul din "lobbies" în "Înapoi", modifică codul HTML în felul următor:

html

Copy code

```
<div id="buton_intoarcere_lobbies"> <a  
href="/lobbies">Înapoi</a> </div>
```

După ce ai modificat textul în codul HTML, va trebui să actualizezi pagina pentru a vedea schimbarea în browser.



si dimenisunea lui?



Pentru a modifica dimensiunea butonului, poți utiliza proprietățile CSS `width` și `height` pentru a seta lățimea și înălțimea dorite. În codul CSS existent, poți adăuga sau modifica următoarele reguli:

css

Copy code

```
#buton_intoarcere_lobbies { /* Stilurile  
existente */ width: 120px; /* Modificarea
```

```
aici */ height: 40px; /* Modificarea aici */  
}
```

Prin setarea valorilor `width` și `height` la `120px` și respectiv `40px`, vei schimba dimensiunile butonului. Poți ajusta aceste valori conform preferințelor tale.

Asigură-te că actualizezi codul CSS și verifici rezultatul

în browser pentru a vedea cum se modifică

---

dimensiunile butonului.



```
<!DOCTYPE html>  
<html lang="en">  
<link rel="stylesheet" href="/style/login-  
template.css">  
<head>  
  
  <meta charset='utf-8'>  
  <meta http-equiv='X-UA-Compatible'  
content='IE=edge'>  
  <title>Login Form</title>  
  <meta name='viewport' content='width=device-  
width, initial-scale=1'>  
</head>  
<body>  
  <h1>Login Form</h1>  
  <form method="post" action="/login"  
enctype="multipart/form-data">  
    <label for="username">Username:</label> <br>  
    <input type="text" id="username"  
name="username"> <br>  
    <label for="password">Password:</label> <br>  
    <input type="password" id="password"  
name="password"> <br> <br>  
    <input type="submit" value="Submit">  
  </form>  
</body>  
</html> adauga unca un buton care sa te duca pe  
pagina de register
```



Pentru a adăuga un buton care să te ducă pe pagina de înregistrare (register), poți modifica codul HTML astfel:

html

Copy code

```
<!DOCTYPE html> <html lang="en"> <link
rel="stylesheet" href="./style/login-
template.css"> <head> <meta charset='utf-8'>
<meta http-equiv='X-UA-Compatible'
content='IE=edge'> <title>Login Form</title>
<meta name='viewport' content='width=device-
width, initial-scale=1'> </head> <body>
<h1>Login Form</h1> <form method="post"
action="/login" enctype="multipart/form-data">
<label for="username">Username:</label><br>
<input type="text" id="username"
name="username"><br> <label
for="password">Password:</label><br> <input
type="password" id="password" name="password">
<br><br> <input type="submit" value="Submit">
</form> <a href="/register">Register</a> <!--
Adăugarea butonului de înregistrare -->
</body> </html>
```

Am adăugat un element de ancoră (<a>) cu atributul href="/register". Acesta va crea un link către pagina de înregistrare. Textul "Register" va fi afișat ca un buton de navigare către pagina respectivă.

Asigură-te că actualizezi codul HTML și verifici rezultatul în browser pentru a vedea butonul de înregistrare adăugat.





```
body {  
  font-family: Arial, sans-serif;  
  background: linear-gradient( to left, #ffcccc,  
#66ccff);  
}  
h1 {  
  text-align: center;  
}  
  
form {  
  max-width: 300px;  
  margin: 0 auto;  
}  
  
label {  
  display: block;  
  margin-bottom: 5px;  
}  
  
input[type="text"],  
input[type="password"] {  
  width: 100%;  
  padding: 8px;  
  margin-bottom: 10px;  
  border: 1px solid #ccc;  
  border-radius: 4px;  
}  
  
input[type="submit"] {  
  width: 100%;  
  padding: 10px;  
  background-color: #007bff;  
  color: #fff;  
  border: none;  
  border-radius: 4px;  
  cursor: pointer;  
}
```

```
input[type="submit"]:hover {  
  background-color: #0056b3;
```

```
}
```

acum adauga si butonul de trimitere la pagina de  
register stilizat



---

Pentru a adăuga și stiliza butonul de trimitere către  
pagina de înregistrare, poți utiliza următorul cod CSS  
adăugat la stilurile existente:

css

Copy code

```
input[type="submit"].register { background-  
color: #28a745; }  
input[type="submit"].register:hover {  
background-color: #218838; }
```

După aceea, modifică codul HTML pentru butonul de  
înregistrare astfel:

html

Copy code

```
<input type="submit" value="Submit"  
class="register">
```

În acest mod, clasa `register` este adăugată butonului  
de trimitere. Aceasta va permite aplicarea stilurilor  
specifice butonului de înregistrare.

Asigură-te că actualizezi codul HTML și CSS, apoi  
verifică rezultatul în browser. Butonul de trimitere către  
pagina de înregistrare ar trebui să fie acum stilizat în  
culorile specificate.





```
<div id = register>  
  <a href="/register">Register</a>  
</div>
```

 fa o stilizare css

Pentru a stiliza elementul `<div>` și link-ul din cadrul acestuia pentru butonul de înregistrare, poți utiliza următorul cod CSS:

css

Copy code

```
#register { text-align: center; margin-top:  
20px; } #register a { display: inline-block;  
padding: 10px 20px; background-color: #28a745;  
color: #fff; text-decoration: none; border-  
radius: 4px; } #register a:hover { background-  
color: #218838; }
```

Asigură-te că adaugi atributul `id="register"` în elementul `<div>` din codul HTML:

html

Copy code

```
<div id="register"> <a  
href="/register">Register</a> </div>
```

Aceste stiluri vor alinia butonul în centru (`text-align: center`), vor adăuga un spațiu de 20px în partea de sus a butonului (`margin-top: 20px`), vor aplica stiluri pentru link (`<a>`) pentru a crea aspectul de buton (culoare de fundal, culoare text, stilul de hover, etc.).

Asigură-te că actualizezi codul HTML și CSS, apoi verifică rezultatul în browser. Elementul `<div>` și link-ul



"Register" ar trebui să fie acum stilizate conform  
regulilor CSS specifice.

---



```
<!DOCTYPE html>
<html lang="en">
<link rel="stylesheet" href="/style/register-
template.css">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible"
content="IE=edge">
  <meta name="viewport" content="width=device-
width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <h1> Regiser Form </h1>
  <form id="my-form" action="/register"
method="post" enctype="multipart/form-data">
    <label for="username">Username:</label>
    <input type="text" id="username"
name="username">
    <br>
    <label for="email">Email:</label>
    <input type="email" id="email" name="email">
    <br>
    <label for="password">Password:</label>
    <input type="password" id="password"
name="password">
    <br>
    <label for="password">Confirm Password:
</label>
    <input type="password" id="confirm-password"
name="confirm-password">
    <br>
    <input type="submit" value="Register">
  </form>
</body>
</html> creeaza o stilizare css pentru asta
```



Pentru a crea o stilizare CSS pentru formularul de înregistrare, poți utiliza următorul cod CSS:

css

 Copy code

```
body { font-family: Arial, sans-serif;
background: linear-gradient(to left, #ffcccc,
#66ccff); } h1 { text-align: center; margin-
top: 20px; } form { max-width: 300px; margin:
0 auto; padding: 20px; background-color: #fff;
border-radius: 4px; } label { display: block;
margin-bottom: 5px; } input[type="text"],
input[type="email"], input[type="password"] {
width: 100%; padding: 8px; margin-bottom:
10px; border: 1px solid #ccc; border-radius:
4px; } input[type="submit"] { width: 100%;
padding: 10px; background-color: #007bff;
color: #fff; border: none; border-radius: 4px;
cursor: pointer; } input[type="submit"]:hover
{ background-color: #0056b3; }
```

Asigură-te că adaugi link-ul către fișierul CSS în secțiunea <head> a documentului HTML:

html

 Copy code

```
<link rel="stylesheet" href="./style/register-
template.css">
```

Aceste stiluri vor aplica următoarele modificări:

Fundalul paginii va fi un gradient liniar între culorile #ffcccc și #66ccff.

Titlul <h1> va fi centrat și va avea un spațiu de 20px în partea de sus.

Formularul <form> va avea o lățime maximă de 300px, va fi centrat, va avea un fundal alb, margini rotunjite și un spațiu de 20px în interior.

Etichetele `<label>` vor fi afișate ca blocuri separate și vor avea un spațiu de 5px în partea de jos.

Câmpurile de introducere `<input>` pentru text, email și parolă vor avea lățime de 100%, un spațiu de 8px deasupra și dedesubt, o bordură subțire de culoare #ccc și margini rotunjite.

Butonul de trimitere `<input type="submit">` va avea lățime de 100%, un spațiu de 10px în interior, fundal de culoare #007bff, text alb, nicio bordură și margini rotunjite. La hover, culoarea fundalului se va schimba în #0056b3.

Asigură-te că actualizezi codul HTML și CSS, apoi verifică rezultatul în browser. Formularul de înregistrare va fi stilizat conform regulilor CSS specificate.



```
<div id="login_alredy">  
  <a href="/login">Login</a>  
</div>
```

 și un buton pentru persoanele care au deja un cont

Pentru a adăuga un buton pentru persoanele care au deja un cont, poți utiliza următorul cod HTML:

html

Copy code

```
<div id="login_already"> <a  
href="/login">Login</a> <button  
type="button">Am deja un cont</button> </div>
```

În acest exemplu, am adăugat un element `<button>` în cadrul div-ului cu id-ul "login\_already". Textul "Am deja un cont" este afișat pe buton.

Pentru a stiliza acest buton, poți folosi codul CSS următor:

css

 Copy code

```
#login_already button { padding: 10px 20px;
background-color: #dc3545; color: #fff;
border: none; border-radius: 4px; cursor:
pointer; } #login_already button:hover {
background-color: #c82333; }
```

În acest exemplu, am definit stilurile pentru butonul "Am deja un cont". Acesta va avea un fundal roșu (`background-color: #dc3545`), text alb (`color: #fff`), nicio bordură (`border: none`), și margini rotunjite (`border-radius: 4px`). La hover, culoarea fundalului se va schimba într-o nuanță mai închisă de roșu (`background-color: #c82333`).

Asigură-te că actualizezi codul HTML și CSS, apoi verifică rezultatul în browser. Div-ul "login\_already" va conține atât link-ul "Login", cât și butonul "Am deja un cont", ambele stilizate conform regulilor CSS specificate.



```
body {
  font-family: Arial, sans-serif;
  background: linear-gradient( to left, #ffcccc,
#66ccff);
}

h1 {
  text-align: center;
  margin-top: 20px;
}
```

```
form {
  max-width: 300px;
  margin: 0 auto;
  padding: 20px;
  background-color: #fff;
  border-radius: 4px;
}

label {
  display: block;
  margin-bottom: 5px;
}

input[type="text"],
input[type="email"],
input[type="password"] {
  width: 100%;
  padding: 8px;
  margin-bottom: 10px;
  border: 1px solid #ccc;
  border-radius: 4px;
}

input[type="submit"] {
  width: 100%;
  padding: 10px;
  background-color: #007bff;
  color: #fff;
  border: none;
  border-radius: 4px;
  cursor: pointer;
}

input[type="submit"]:hover {
  background-color: #0056b3;
}

#login_already button {
  padding: 10px 20px;
  background-color: #dc3545;
  color: #fff;
}
```

```
border: none;
border-radius: 4px;
cursor: pointer;
}

#login_already button:hover {
  background-color: #c82333;
}
```

vreau sa apara in aceeași caseta sub butonul submit



---

Pentru a face ca butonul "Am deja un cont" să apară în aceeași casetă sub butonul "Submit" în formularul de înregistrare, poți utiliza următorul cod CSS:

CSS

 Copy code

```
#my-form { position: relative; }
#login_already { text-align: center; margin-top: 20px; position: absolute; bottom: -40px; left: 0; right: 0; }
```

Acest cod CSS utilizează proprietatea **position**: **relative** pentru formularul `<form>` cu id-ul "my-form" și proprietatea **position**: **absolute** pentru div-ul cu id-ul "login\_already". Proprietatea **position**: **absolute** permite plasarea div-ului în raport cu formularul.

De asemenea, se utilizează proprietățile **bottom**: **-40px** și **left**: **0** pentru a poziționa div-ul în partea de jos a formularului, la o distanță de 40 de pixeli sub formular, și **right**: **0** pentru a-l alinia la dreapta formularului.

Asigură-te că actualizezi codul CSS, apoi verifică rezultatul în browser. Butonul "Am deja un cont" va apărea acum în aceeași casetă sub butonul "Submit" în formularul de înregistrare.



```
<div id="mare" style="display: flex">
  <div id="unu" style="" class="carte_de_joc">
    
  </div>
  <div id="container" style="">
    <div id="chat">
    </div>
    <div class="chat-input">
      <input placeholder="Message" type="text"
id="message">
      <button onclick="sendMessage()" class="send-
button">Send</button>
    </div>
  </div>
</div> in css cum fac sa fie mai mica cartea de joc
```



Pentru a face ca imaginea cartei de joc să fie mai mică, poți utiliza următorul cod CSS:

css

Copy code

```
#unu { width: 100px; /* sau orice valoare
dorită pentru lățime */ } #unu img { width:
100%; /* pentru a asigura redimensionarea
imaginii în funcție de lățimea div-ului */
height: auto; /* pentru a menține proporțiile
originale ale imaginii */ }
```

Aceste stiluri vor aplica o lățime specifică pentru div-ul cu id-ul "unu", care conține imaginea cartei de joc. Apoi, imaginii din interiorul div-ului se va aplica o lățime de 100% pentru a se adapta la lățimea div-ului



și înălțimea va fi ajustată în mod automat pentru a menține proporțiile originale ale imaginii.

Asigură-te că actualizezi codul CSS, apoi verifică rezultatul în browser. Cartea de joc va fi redimensionată la dimensiunea specificată și se va potrivi în cadrul div-ului.



cum fac sa afisez containerul de joc cu cartile utilizatorului in partea de jos si cele intoarse pe lateral

```
#container_joc {
    display: flex;
    margin-left: 1vw;
    width: 100px; /* sau orice valoare dorită pentru
lățime */
}
#container_joc carti{
    display: flex;
    margin-left: 1vw;
    width: 100px; /* sau orice valoare dorită pentru
lățime */
}

#container_joc img {

    width: 100%; /* pentru a asigura redimensionarea
imaginii în funcție de lățimea div-ului */
    height: auto; /* pentru a menține proporțiile
originale ale imaginii */
}
#carti_intoarse img {

    width: 100%; /* pentru a asigura redimensionarea
imaginii în funcție de lățimea div-ului */
    height: auto; /* pentru a menține proporțiile
originale ale imaginii */
}
```

html ul

```

<div id="container_joc" style="display: flex">
  <div id = "carti_intoarse">
    
    
    
    
    
    
    
    
    
  </div>
  <div id="carti_jucator" style=""
class="carte_de_joc">
    
    
    
    
    
    
    

```



&lt;/div&gt;

Pentru a afișa containerul de joc cu cărțile utilizatorului în partea de jos și cărțile întoarse pe lateral, poți utiliza următorul cod CSS:

css

Copy code

```
#container_joc { display: flex; flex-direction: column; align-items: flex-start; margin-left: 1vw; width: fit-content; }
#carti_jucator, #carti_intoarse { display: flex; flex-wrap: nowrap; }
#carti_jucator { justify-content: flex-start; }
#carti_intoarse { justify-content: center; flex-direction: column; }
#carti_jucator img { width: 100px; height: auto; }
#carti_intoarse img { width: auto; height: 100px; margin-bottom: 5px; }
```

Aceste stiluri vor afișa div-ul principal "container\_joc" ca un container flexibil, cu direcția de coloană și alinierea elementelor către începutul containerului. Cărțile jucătorului ("carti\_jucator") vor fi aliniate către începutul containerului, iar cărțile întoarse ("carti\_intoarse") vor fi aliniate central și vor avea direcția de coloană.

De asemenea, stilurile vor redimensiona cărțile astfel încât să se afișeze în lățimea și înălțimea dorită.

Asigură-te că actualizezi codul CSS, apoi verifică rezultatul în browser. Cărțile jucătorului vor fi afișate în partea de jos a containerului, iar cărțile întoarse vor fi afișate pe lateral și vor fi aliniate vertical una sub alta.





poti sa faci sa se suprapuna cartile de joc intoarse? nu in totalitate, doar cat sa se vada mai frumos pe site



Da, poți folosi stiluri CSS pentru a face ca cărțile de joc întoarse să se suprapună parțial pentru a arăta mai frumos pe site. Iată un exemplu de cod CSS pe care îl poți adăuga:

css

Copy code

```
#carti_intoarse { position: relative; }  
#carti_intoarse img { position: absolute;  
transform: translateX(-50%); } #carti_intoarse  
img:not(:last-child) { margin-right: -20px; }
```

Aceste stiluri aplică o poziție relativă pe containerul "carti\_intoarse". Apoi, cărțile de joc întoarse sunt plasate într-o poziție absolută și se folosește transformarea CSS `translateX(-50%)` pentru a le centra pe orizontală în interiorul containerului.

De asemenea, pentru a crea suprapunerea parțială, se aplică o margine negativă de -20px la toate cărțile între prima și penultima cărtică folosind selectorul `:not(:last-child)`.

Asigură-te că actualizezi codul CSS și verifică rezultatul în browser. Cărțile de joc întoarse se vor suprapune parțial, creând un aspect mai plăcut pe site.



si unde inlocuiesc



Poți înlocui codul CSS cu stilurile actualizate în fișierul tău CSS extern sau în secțiunea `<style>` a fișierului HTML. Dacă ai un fișier CSS separat, deschide acel fișier și înlocuiește stilurile pentru `#carti_intoarse` și `#carti_intoarse img` cu cele furnizate mai sus.

Dacă dorești să adaugi stilurile direct în fișierul HTML, poți utiliza un bloc `<style>` în secțiunea `<head>` așa cum este prezentat mai jos:

html

 Copy code

```
<head> <meta charset="UTF-8"> <meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Document</title> <style>
#carti_intoarse { position: relative; }
#carti_intoarse img { position: absolute;
transform: translateX(-50%); } #carti_intoarse
img:not(:last-child) { margin-right: -20px; }
</style> </head>
```

Asigură-te că actualizezi codul CSS în fișierul sau secțiunea corespunzătoare și verifică rezultatul în browser. Cărțile de joc întoarse se vor suprapune parțial conform stilurilor actualizate.



```
#container_joc {
display: flex;
flex-direction: column;
align-items: flex-start;
margin-left: 1vw;
width: fit-content;
}
```

```
#carti_jucator,  
#carti_intoarse {  
    display: flex;  
    flex-wrap: nowrap;  
}  
  
#carti_jucator {  
    justify-content: flex-start;  
}  
  
#carti_intoarse {  
    justify-content: center;  
    flex-direction: column;  
}  
  
#carti_jucator img {  
    width: 100px;  
    height: auto;  
}  
  
#carti_intoarse img {  
    width: auto;  
    height: 100px;  
    margin-bottom: 5px;  
}
```

si in css ul asta ce pot sa inlocuiesc



---

Blocul CSS pe care l-ai prezentat se referă la stilurile pentru elementele din interiorul containerului `#container_joc`. Dacă dorești să înlocuiești stilurile pentru un element `<u1>` specific, va trebui să adaugi un identificator sau o clasă corespunzătoare în codul HTML.

De exemplu, dacă ai un element `<u1>` în interiorul `#container_joc` căruia dorești să-i aplici stiluri specifice, poți adăuga un identificator sau o clasă la acel element

în codul HTML și să actualizezi stilurile CSS în consecință.

HTML:

html

 Copy code

```
<div id="container_joc" style="display: flex">
<ul id="lista_carti"> <li>Carte 1</li>
<li>Carte 2</li> <li>Carte 3</li> <!-- ... -->
</ul> </div>
```

CSS:

css

 Copy code

```
#lista_carti { /* Stilurile pentru <ul> */ /*
... */ } #lista_carti li { /* Stilurile pentru
fiecare <li> din interiorul <ul> */ /* ... */
}
```

Înlocuiește "lista\_carti" cu identificatorul sau clasa corespunzătoare pe care îl dorești și adaugă stilurile specifice în acele selecții. Asigură-te că adaugi aceste stiluri în fișierul CSS sau secțiunea `<style>` a fișierului HTML pentru a le aplica corespunzător.



```
#carti_jucator,
#carti_intoarse_stanga,
#carti_intoarse_fata,
#carti_intoarse_dreapta
{
    display: flex;
    flex-wrap: nowrap;
```

```
}
```

```
#carti_jucator {  
    justify-content: flex-start;  
}
```

```
#carti_jucator img {  
    width: 90px;  
    height: auto;  
}
```

```
/*stanga*/  
#carti_intoarse_stanga {  
    justify-content: center;  
    flex-direction: column;  
}
```

```
#carti_intoarse_stanga img {  
    width: auto;  
    height: 100px;  
    margin-bottom: 5px;  
}
```

```
/*fata*/  
#carti_intoarse_fata {  
    justify-content: center;  
    flex-direction: row;  
}
```

```
#carti_intoarse_fata img {  
    width: auto;  
    height: 100px;  
    margin-bottom: 5px;  
}
```

```
/* dreapta*/
```

```
#carti_intoarse_dreapta {  
    justify-content: center;  
    flex-direction: column;  
}
```



```
#carti_intoarse_dreapta img {
  width: auto;
  height: 100px;
  margin-bottom: 5px;}
```

daca vreau sa le pun pe chenarul unui patrat, cele 3  
tipuri de carti intoarse, cum fac help



Dacă dorești să plasezi cele trei tipuri de cărți întoarse  
în jurul chenarului unui pătrat, poți utiliza o combinație  
de elemente `<div>` pentru a crea structura dorită.

Iată un exemplu de cod HTML și CSS care ar putea  
realiza acest aspect:

HTML:

html

 Copy code

```
<div id="chenar"> <div id="patrat"> <div
id="carti_intoarse_stanga">    </div> <div
id="carti_intoarse_fata">    </div> <div
id="carti_intoarse_dreapta">    </div> </div> </div>
```

CSS:

css

 Copy code

```
#chenar { display: flex; justify-content: center; align-items: center; width: 300px; height: 300px; background-color: #ccc; }
#patrat { position: relative; width: 200px; height: 200px; border: 2px solid #000; background-color: #fff; }
#carti_intoarse_stanga, #carti_intoarse_fata, #carti_intoarse_dreapta { display: flex; flex-direction: column; align-items: center; position: absolute; }
#carti_intoarse_stanga { top: 0; left: 0; }
#carti_intoarse_fata { top: 50%; left: 50%; transform: translate(-50%, -50%); }
#carti_intoarse_dreapta { bottom: 0; right: 0; }
```

În acest exemplu, #chenar reprezintă containerul principal în care se află pătratul și cele trei tipuri de cărți întoarse. #patrat reprezintă pătratul în sine, iar #carti\_intoarse\_stanga, #carti\_intoarse\_fata și #carti\_intoarse\_dreapta reprezintă cele trei secțiuni în care sunt plasate cărțile întoarse. Stilurile CSS aplicate controlează poziționarea și aspectul vizual al acestor elemente.

Asigură-te că înlocuiești link-urile către imaginile cărților întoarse cu link-urile corespunzătoare în codul tău. De asemenea, poți ajusta dimensiunile, culorile și alte proprietăți



patratul ala e imaginar, deja am un div cu un id care se numeste container joc(el e patratul meu imaginar)



Dacă deja ai un div cu id-ul "container\_joc" care reprezintă patratul tău imaginar, poți utiliza acest div ca și container pentru cele trei tipuri de cărți întoarse.

În codul CSS, poți actualiza stilurile pentru a poziționa și dimensiona corect cărțile întoarse în cadrul "container\_joc". De asemenea, va trebui să actualizezi id-urile secțiunilor pentru cărți întoarse astfel încât să se potrivească cu id-urile existente.

Iată un exemplu de cum ar putea arăta codul CSS actualizat:

css

 Copy code

```
#container_joc { position: relative; width: 200px; height: 200px; border: 2px solid #000; background-color: #fff; }
#carti_intoarse_stanga, #carti_intoarse_fata, #carti_intoarse_dreapta { display: flex; flex-direction: column; align-items: center; position: absolute; }
#carti_intoarse_stanga { top: 0; left: 0; }
#carti_intoarse_fata { top: 50%; left: 50%; transform: translate(-50%, -50%); }
#carti_intoarse_dreapta { bottom: 0; right: 0; }
#carti_intoarse_stanga img, #carti_intoarse_fata img, #carti_intoarse_dreapta img { width: auto; height: 100px; margin-bottom: 5px; }
```

În codul HTML, trebuie să actualizezi id-urile secțiunilor pentru cărțile întoarse astfel încât să se potrivească cu noile id-uri din CSS. De exemplu:

html

 Copy code

```
<div id="container_joc"> <div id="carti_intoarse_stanga"> <!-- Imagini cărți întoarse stânga --> </div> <div
```

```
id="carti_intoarse_fata"> <!-- Imagini cărți  
întoarse față --> </div> <div  
id="carti_intoarse_dreapta"> <!-- Imagini  
cărți întoarse dreapta --> </div> </div>
```

Asigură-te că înlocuiești comentariile din codul HTML cu link-urile corespunzătoare către imaginile cărților întoarse.

Aceste modificări ar trebui să poziționeze și să dimensioneze cărțile întoarse în cadrul "container\_joc", astfel încât să creeze efectul vizual dorit.



vreau sa imi puna pe marginea containerului, nu pe toata pagina web



Dacă dorești să plasezi cărțile întoarse doar pe marginea containerului și nu pe întreaga pagină web, poți utiliza poziționarea absolută relativă la containerul părinte.

Iată un exemplu de cod CSS pentru a obține acest efect:

css

Copy code

```
#container_joc { position: relative; width:  
200px; height: 200px; border: 2px solid #000;  
background-color: #fff; margin: 0 auto; /*  
Centrare pe orizontală */ }  
#carti_intoarse_stanga, #carti_intoarse_fata,  
#carti_intoarse_dreapta { display: flex; flex-  
direction: column; align-items: center;  
position: absolute; } #carti_intoarse_stanga {  
top: 0; left: -50px; /* Poziționare la  
marginea stângă */ } #carti_intoarse_fata {
```

```

top: 50%; right: -50px; /* Poziționare la
marginea dreaptă */ transform: translate(0,
-50%); } #carti_intoarse_dreapta { bottom: 0;
left: -50px; /* Poziționare la marginea stângă
*/ } #carti_intoarse_stanga img,
#carti_intoarse_fata img,
#carti_intoarse_dreapta img { width: auto;
height: 100px; margin-bottom: 5px; }

```

În acest exemplu, am adăugat și stilul `margin: 0 auto;` pentru elementul cu id-ul "container\_joc" pentru a centra containerul pe orizontală în cadrul paginii web. Poziționarea absolută a cărților întoarse se face relativ la acest container.

Asigură-te că înlocuiești comentariile din codul HTML cu link-urile corespunzătoare către imaginile cărților întoarse și încorporează codul CSS în fișierul tău CSS existent sau adaugă-l în `<style>` tag-ul din cadrul fișierului HTML.

Aceste modificări ar trebui să plaseze cărțile întoarse pe marginea containerului "container\_joc" într-o poziție relativă la acesta.



```

#container_joc {
  flex: 0 0 75%;

  display: flex;
  flex-direction: column;
  align-items: flex-start;
  /*margin-left: 1vw;*/
  width: fit-content;
  margin: 0 auto;
}
#carti_jucator {

```

```
        justify-content: flex-start;
        flex-direction: row;
    }
    #carti_jucator,
    #carti_intoarse_stanga,
    #carti_intoarse_fata,
    #carti_intoarse_dreapta {
        display: flex;

        align-items: center;
        position: absolute;
    }
    #carti_jucator img {
        width: 90px;
        height: auto;
    }
    #carti_intoarse_stanga {
        top: 0;
        left: 0;
        flex-direction: column;
    }

    #carti_intoarse_fata {
        top: 50%;
        left: 50%;
        /*transform: translate(-50%, -50%);*/
        flex-direction: row;
    }

    #carti_intoarse_dreapta {
        bottom: 0;
        right: 0;
        flex-direction: column;
    }

    #carti_intoarse_stanga img,
    #carti_intoarse_fata img,
    #carti_intoarse_dreapta img {
        width: 90px;
        height: 50px;
    }
```

```
margin-bottom: 5px;
}
```

vreau sa depinda toate de container\_joc



Pentru a face ca toate elementele (carti\_jucator, carti\_intoarse\_stanga, carti\_intoarse\_fata și carti\_intoarse\_dreapta) să depindă de elementul container\_joc, poți folosi poziționarea relativă pe container\_joc și poziționarea absolută pe celelalte elemente în raport cu acesta.

Iată codul CSS actualizat:

CSS

 Copy code

```
#container_joc { flex: 0 0 75%; display: flex;
flex-direction: column; align-items: flex-
start; width: fit-content; margin: 0 auto;
position: relative; /* Adăugăm poziționarea
relativă */ } #carti_jucator { justify-
content: flex-start; flex-direction: row;
position: absolute; top: 0; /* Poziționăm
carti_jucator relativ la container_joc */
left: 0; } #carti_intoarse_stanga { flex-
direction: column; position: absolute; top: 0;
/* Poziționăm carti_intoarse_stanga relativ la
container_joc */ left: 0; }
#carti_intoarse_fata { flex-direction: row;
position: absolute; top: 50%; /* Poziționăm
carti_intoarse_fata relativ la container_joc
*/ left: 50%; transform: translate(-50%,
-50%); } #carti_intoarse_dreapta { flex-
direction: column; position: absolute; bottom:
0; /* Poziționăm carti_intoarse_dreapta
relativ la container_joc */ right: 0; }
#carti_jucator img, #carti_intoarse_stanga
img, #carti_intoarse_fata img,
#carti_intoarse_dreapta img { width: 90px;
height: 50px; margin-bottom: 5px; }
```

Asigură-te că înlocuiești link-urile către imaginile corespunzătoare pentru cărțile de joc și adaugă codul CSS actualizat în fișierul tău CSS sau în `<style>` tag-ul din fișierul HTML. Acum, toate elementele cartilor se vor poziționa relativ la `container_joc`.



iar acum vreau carti jucator sa fie la baza, iar restul sa fie in partea stanga, partea din fata si partea din dreapta



Pentru a poziționa `carti_jucator` la baza, iar celelalte elemente în partea stângă, partea din față și partea dreaptă, vom ajusta proprietățile CSS pentru fiecare element.

Iată codul CSS actualizat:

CSS

 Copy code

```
#container_joc { flex: 0 0 75%; display: flex;
flex-direction: column; align-items: flex-
start; width: fit-content; margin: 0 auto;
position: relative; } #carti_jucator {
justify-content: flex-start; flex-direction:
row; position: absolute; bottom: 0; /*
Poziționăm carti_jucator la baza container_joc
*/ left: 0; } #carti_intoarse_stanga { flex-
direction: column; position: absolute; top:
50%; /* Poziționăm carti_intoarse_stanga în
partea stângă a container_joc */ left: 0;
transform: translateY(-50%); }
#carti_intoarse_fata { flex-direction: row;
position: absolute; top: 50%; /* Poziționăm
carti_intoarse_fata în partea din față a
container_joc */ left: 50%; transform:
translate(-50%, -50%); }
#carti_intoarse_dreapta { flex-direction:
column; position: absolute; top: 50%; /*
Poziționăm carti_intoarse_dreapta în partea
```



```
dreapta a container_joc */ right: 0;
transform: translateY(-50%); } #carti_jucator
img, #carti_intoarse_stanga img,
#carti_intoarse_fata img,
#carti_intoarse_dreapta img { width: 90px;
height: 50px; margin-bottom: 5px; }
```

Cu acest cod CSS actualizat, carti\_jucator se va afla în partea de jos a container\_joc, iar celelalte elemente se vor poziționa în partea stângă (carti\_intoarse\_stanga), în partea din față (carti\_intoarse\_fata) și în partea dreaptă (carti\_intoarse\_dreapta) a container\_joc.

Asigură-te că înlocuiești link-urile către imaginile corespunzătoare pentru cărțile de joc și adaugă codul CSS actualizat în fișierul tău CSS sau în <style> tag-ul din fișierul HTML.



<https://www.searchamateur.com/pictures/solitaire-spades.jpg> poti sa vezi imaginea asta?



Ca model de limbă AI, nu pot accesa sau vizualiza conținut extern, cum ar fi imagini de la URL-uri specifice. Cu toate acestea, dacă dorești să afișezi imaginea respectivă în HTML, poți utiliza următorul cod HTML:

html

Copy code

```

```



Asigură-te că adaugi acest cod HTML în locul potrivit din fișierul tău HTML și să înlocuiești URL-ul imaginii cu URL-ul corespunzător imaginii pe care o dorești.



<https://www.searchchamateur.com/pictures/solitaire-spades.jpg> fa l sa arate asa



Pentru a afișa imaginea în HTML și pentru a-i aplica stilizarea corespunzătoare, poți utiliza următorul cod HTML și CSS:

HTML:

html

Copy code

```
<div id="image-container">  </div>
```



CSS:

css

Copy code

```
#image-container { display: flex; justify-  
content: center; } #image-container img {  
width: 100%; height: auto; max-width: 500px; }
```

Asigură-te că adaugi acest cod HTML și CSS în fișierul tău și personalizează valorile de lățime (`max-width`) și înălțime (`max-height`) ale imaginii pentru a obține aspectul dorit.





cum fac mai mari cartile jucator



Pentru a mări dimensiunea cărților jucătorului, poți ajusta valorile proprietăților CSS ale imaginilor respective. În codul CSS existent, găsește secțiunea care se referă la cărțile jucătorului (`#carti_jucator img`) și modifică valorile pentru `width` și `height`:

CSS

 Copy code

```
#carti_jucator img { width: 120px; height: auto; }
```

Prin modificarea valorii `width` la 120px (sau orice altă valoare dorită), vei mări dimensiunea cărților jucătorului. Asigură-te că ajustezi aceste valori în funcție de preferințele tale pentru dimensiunea cărților.



faci un flex box mai mare de 3 pe 3 in care pozitiile impare sa fie ocupate de cart



Desigur! Poți crea un flex container mai mare de 3x3 pentru a poziționa cărțile în locațiile impare. Iată un exemplu de cod CSS care ar putea realiza acest lucru:

CSS

 Copy code

```
.container { display: flex; flex-wrap: wrap; width: 300px; height: 300px; } .card { width: 100px; height: 100px; background-color:
```

```
#f0f0f0; margin: 5px; } .card:nth-child(odd) {
order: 1; }
```

În acest exemplu, `.container` reprezintă elementul care conține cărțile, iar `.card` reprezintă fiecare carte individuală. Utilizăm proprietatea `order` pentru a poziționa cărțile impare înaintea celor pare în fluxul flexibil. Dimensiunile și stilul pot fi personalizate în funcție de preferințele tale.

Iată un exemplu de cod HTML pentru a folosi aceste stiluri:

html

 Copy code

```
<div class="container"> <div
class="card">1</div> <div class="card">2</div>
<div class="card">3</div> <div
class="card">4</div> <div class="card">5</div>
<div class="card">6</div> <div
class="card">7</div> <div class="card">8</div>
<div class="card">9</div> </div>
```

Această structură de cod va afișa un container flexibil cu un total de 9 cărți, iar cărțile impare vor fi poziționate în locațiile impare ale grilei.



```
<div id="container_joc" style="display: flex">
  <div id = "carti_intoarse_stanga">
    
    
    
    
    
```

```
1.3/red_joker.svg">  
      
      
      
</div>
```

```
<div id = "carti_intoarse_fata">  
      
      
      
      
      
      
      
      
      
</div>
```

```
<div id = "carti_intoarse_dreapta">  
      
      
      
      
      
    
```

```

1.3/red_joker.svg">
    
    
</div>
<div id="carti_jucator" style=""
class="carte_de_joc">
    
    
    
    
    
    
    

```

</div> si vreau ca divul container joc sa fie affisat sub forma unui grid de 3x3 astfel incat carti jucator sa fie pe celula [3,2], carti intoarse dreapta sa fie pe celula [2,3], carti intoarse stanga sa fie pe celula [2,1] si carti intoarse fata pe celula [1,2]



Pentru a realiza afişarea sub formă de grid de 3x3 cu cele patru div-uri, poți utiliza proprietatea CSS `grid-template-areas`. Iată cum ar putea arăta codul CSS pentru a realiza acest lucru:

css

Copy code

```

#container_joc { display: grid; grid-template-
columns: 1fr 1fr 1fr; grid-template-rows: 1fr

```

```
1fr 1fr; grid-gap: 10px; width: 600px; height:
600px; } #carti_jucator { grid-area: 3 / 2 / 4
/ 3; } #carti_intoarse_dreapta { grid-area: 2
/ 3 / 3 / 4; } #carti_intoarse_stanga { grid-
area: 2 / 1 / 3 / 2; } #carti_intoarse_fata {
grid-area: 1 / 2 / 2 / 3; }
```

Iar în structura HTML, poți utiliza aceleași div-uri și id-uri precum le-ai prezentat, dar adăugând un div în jurul lor cu id-ul "container\_joc":

html

 Copy code

```
<div id="container_joc"> <div
id="carti_intoarse_stanga"> <!-- Cărți
întoarse stânga --> </div> <div
id="carti_intoarse_fata"> <!-- Cărți întoarse
față --> </div> <div
id="carti_intoarse_dreapta"> <!-- Cărți
întoarse dreapta --> </div> <div
id="carti_jucator"> <!-- Cărți jucător -->
</div> </div>
```

În acest fel, div-urile vor fi poziționate în grila de 3x3 în conformitate cu specificațiile CSS din codul de mai sus. Poziționarea fiecărui div în celula corespunzătoare este definită prin utilizarea proprietății `grid-area`.



am o problema, #container\_joc {  
flex: 0 0 75%;  
display: grid;  
flex-direction: column;  
align-items: flex-start;  
width: fit-content;  
margin: 0 auto;  
position: relative;  
}

```
#carti_jucator {
    justify-content: flex-start;
    flex-direction: row;
    position: absolute;
    bottom: 0; /* Poziționăm carti_jucator la baza
container_joc */
    left: 10%;
}

#carti_intoarse_stanga {
    display: flex;
    flex-direction: column;
    position: absolute;
    top: 50%; /* Poziționăm carti_intoarse_stanga în
partea stângă a container_joc */
    left: 0;

}

#carti_intoarse_fata {
    flex-direction: row;
    position: absolute;
    top: 0;
    left: 10%;

}

#carti_intoarse_dreapta {
    display: flex;
    flex-shrink: 1;
    flex-direction: column;
    position: absolute;
    top: 50%; /* Poziționăm carti_intoarse_dreapta în
partea dreaptă a container_joc */
    right: 0;

}

#carti_jucator img{
```



```
width: 120px;
height: auto;

}
#carti_intoarse_stanga img,
#carti_intoarse_fata img,
#carti_intoarse_dreapta img {
  width: 100px;
  height: 100px;
  margin-bottom: 1px;
}
#container_joc {
  display: grid;
  grid-template-columns: 1fr 1fr 1fr;
  grid-template-rows: 1fr 1fr 1fr;
  grid-gap: 10px;
  width: 600px;
  height: 600px;
}
```

```
#carti_jucator {
  grid-area: 3 / 2 / 4 / 3;
}
```

```
#carti_intoarse_dreapta {
  grid-area: 2 / 3 / 3 / 4;
}
```

```
#carti_intoarse_stanga {
  grid-area: 2 / 1 / 3 / 2;
}
```

```
#carti_intoarse_fata {
  grid-area: 1 / 2 / 2 / 3;
}
```

cartile intoarse ies din grid, cum pot sa le fac sa arate bine



---

Dacă cartile\_intoarse ies din grid, acest lucru se poate datora poziționării absolute utilizate pentru div-urile

respective. Pentru a le face să se încadreze în grilă, poți încerca următoarele modificări în CSS:

css

 Copy code

```
#carti_intoarse_stanga { display: flex; flex-direction: column; position: relative; /* Schimbăm poziționarea în relative */ grid-area: 2 / 1 / 3 / 2; /* Păstrăm poziționarea în grilă */ } #carti_intoarse_fata { flex-direction: row; position: relative; /* Schimbăm poziționarea în relative */ grid-area: 1 / 2 / 2 / 3; /* Păstrăm poziționarea în grilă */ } #carti_intoarse_dreapta { display: flex; flex-direction: column; position: relative; /* Schimbăm poziționarea în relative */ grid-area: 2 / 3 / 3 / 4; /* Păstrăm poziționarea în grilă */ }
```

Prin schimbarea poziționării în **relative** pentru aceste div-uri, ele ar trebui să rămână în grilă și să se potrivească corect în cadrul celulelor respective.

Asigură-te că ai eliminat sau modificat codul CSS care definește poziționarea absolută pentru acele div-uri.

Dacă problema persistă sau dacă ai nevoie de alte ajustări, te rog să îmi furnizezi codul HTML și CSS actualizat, astfel încât să pot ajuta mai precis cu problema întâmpinată.



adica cartile intoarse din dreapta si din stanga trebuie sa fie pozitionate pana in cartile jucatorului



Dacă dorești ca `cartile_intoarse_stanga` și `cartile_intoarse_dreapta` să fie poziționate până în dreptul `cartilor_jucator`, poți adăuga o valoare de `right` la poziționarea absolută a div-urilor respective.

Iată cum poți modifica CSS-ul:

CSS

 Copy code

```
#carti_intoarse_stanga { display: flex; flex-direction: column; position: absolute; top: 50%; /* Poziționăm carti_intoarse_stanga în partea stângă a container_joc */ left: 0; right: calc(100% + 10px); /* Poziționează div-ul până în dreptul cartilor_jucator */ }
#carti_intoarse_dreapta { display: flex; flex-direction: column; position: absolute; top: 50%; /* Poziționăm carti_intoarse_dreapta în partea dreaptă a container_joc */ right: 0; left: calc(100% + 10px); /* Poziționează div-ul până în dreptul cartilor_jucator */ }
```

Prin adăugarea proprietății `right` și specificând valoarea `calc(100% + 10px)`, div-urile `carti_intoarse_stanga` și `carti_intoarse_dreapta` vor fi plasate în partea stângă, respectiv dreaptă, până în dreptul `cartilor_jucator`.

Asigură-te că ajustezi valoarea `10px` în funcție de spațiul dorit între aceste div-uri și `cartile_jucator`.

Aceasta ar trebui să rezolve problema poziționării div-urilor în relație cu `cartile_jucator`.



```
#container_joc {
  display: grid;
```

```
grid-template-columns: 1fr 1fr 1fr;  
grid-template-rows: 1fr 1fr 1fr;  
grid-gap: 10px;  
width: 600px;  
height: 600px;  
}
```

```
#carti_jucator {  
  grid-area: 3 / 2 / 4 / 3;  
}
```

```
#carti_intoarse_dreapta {  
  grid-area: 1 / 3 / 3 / 4;  
}
```

```
#carti_intoarse_stanga {  
  grid-area: 1 / 1 / 3 / 2;  
}
```

```
#carti_intoarse_fata {  
  grid-area: 1 / 2 / 2 / 3;  
}
```

acum vreau ca carti intoarse stanga sa pece de pe  
prima linie si carti intoarse stanga sa plece de pe  
ultima linie



---

Dacă dorești ca `carti_intoarse_stanga` să fie plasat pe prima linie a gridului, iar `carti_intoarse_dreapta` să fie plasat pe ultima linie, trebuie să modificăm valorile `grid-area` pentru aceste div-uri.

Iată cum poți face modificările:

CSS

 Copy code

```
#carti_intoarse_stanga { grid-area: 1 / 1 / 2  
/ 4; /* Modificăm valorile pentru a ocupa  
prima linie complet */ }  
#carti_intoarse_dreapta { grid-area: 3 / 1 / 4
```

```
/ 4; /* Modificăm valorile pentru a ocupa  
ultima linie complet */ }
```

Prin actualizarea valorilor `grid-area`,  
`carti_intoarse_stanga` va fi plasat pe prima linie (de la  
celula (1, 1) până la celula (2, 4)) și  
`carti_intoarse_dreapta` va fi plasat pe ultima linie (de  
la celula (3, 1) până la celula (4, 4)).

Aceste modificări ar trebui să plaseze corect div-urile  
`carti_intoarse_stanga` și `carti_intoarse_dreapta` pe  
gratul de 4x3.



ce e un flexbox



---

Un flexbox (flex container) este un tip de container  
flexibil în CSS care permite aranjarea și alinierea  
elementelor sale interne (flex items) într-o direcție  
orizontală sau verticală.

Cu ajutorul flexbox-ului, poți crea layout-uri flexibile și  
responsive, gestionând mai ușor poziționarea,  
dimensiunile și ordinea elementelor pe un container.  
Flexbox-ul este ideal pentru a crea bare de navigare,  
bare laterale, bare de antet sau orice altă secțiune a  
unei pagini web care necesită aranjarea și alinierea  
flexibilă a elementelor.

Pentru a utiliza flexbox-ul în CSS, trebuie să definești  
un container flexibil (flex container) prin aplicarea  
proprietății `display: flex;` sau `display: inline-flex;`  
asupra elementului părinte care va deveni containerul  
flex. Apoi, poți controla comportamentul și aspectul

flexbox-ului prin utilizarea unor proprietăți precum `flex-direction`, `justify-content`, `align-items`, `flex-wrap` și altele.

Iată un exemplu simplu de utilizare a flexbox-ului:

html

 Copy code

```
<div class="flex-container"> <div class="flex-item">Element 1</div> <div class="flex-item">Element 2</div> <div class="flex-item">Element 3</div> </div>
```

css

 Copy code

```
.flex-container { display: flex; justify-content: center; /* Aliniere orizontală a elementelor în centru */ align-items: center; /* Aliniere verticală a elementelor în centru */ } .flex-item { margin: 10px; }
```

În acest exemplu, clasa CSS `.flex-container` definește un container flexibil, iar clasa CSS `.flex-item` definește elementele interne (flex items). Proprietățile `justify-content` și `align-items` sunt utilizate pentru a alinia elementele în centru, atât pe orizontală, cât și pe verticală.

Flexbox-ul oferă multe alte opțiuni și proprietăți pentru a controla modul în care elementele sunt poziționate și aliniate într-un container flexibil. Este recomandat să consultați resurse suplimentare și ghiduri despre flexbox pentru a înțelege în detaliu utilizarea și posibilitățile acestui concept în CSS.



xhr.open("GET", 'http://localhost:5000/lobbyMembers/'

+

```

document.getElementById("lobby_name").innerHTML
xhr.onload = function () {
    console.log(xhr.response)
    vec = JSON.parse(xhr.response)
    console.log(vec)
    var addedMembers = []; // Variabila pentru a tine
evidenta membrilor adaugati
    for (x in vec) {
        console.log(vec[x])
        if (!addedMembers.includes(vec[x])) { //
Verificam daca membrul exista deja in lista
            console.log(vec[x])
            var span = document.createElement("span");
            span.innerHTML = vec[x];

document.getElementsByClassName("membrii")
[0].appendChild(span);
            addedMembers.push(vec[x]); // Adaugam
membrul in lista de membri adaugati
        }
    }cum pot sa imi iau de aici membrii pe care sa ii
pun sub fiecare div cu carti

```



Pentru a utiliza membrii obținuți prin intermediul cererii XHR și a-i afișa sub fiecare div cu carti, poți adăuga codul necesar în bucla `for` existentă. Iată cum poți face acest lucru:

javascript

 Copy code

```

xhr.open("GET",
'http://localhost:5000/lobbyMembers/' +
document.getElementById("lobby_name").innerHTML);
xhr.onload = function () {
    console.log(xhr.response); var vec =
JSON.parse(xhr.response); console.log(vec);
var addedMembers = []; // Variabila pentru a
tine evidenta membrilor adaugati for (x in
vec) { console.log(vec[x]); if

```

```
(!addedMembers.includes(vec[x])) { //
Verificam daca membrul exista deja in lista
console.log(vec[x]); var span =
document.createElement("span"); span.innerHTML
= vec[x];
document.getElementsByClassName("membrii")
[0].appendChild(span);
addedMembers.push(vec[x]); // Adaugam membrul
in lista de membri adaugati // Adaugare membru
sub fiecare div cu carti var cartiJucator =
document.querySelectorAll("#carti_jucator
.carte_de_joc"); cartiJucator.forEach(function
(carte) { var membruSpan =
document.createElement("span");
membruSpan.innerHTML = vec[x];
carte.appendChild(membruSpan); }); } } };
xhr.send();
```

În acest exemplu, am adăugat un nou fragment de cod pentru a itera prin fiecare element `.carte_de_joc` din div-ul `#carti_jucator` și a adăuga un nou element `<span>` care conține membrul corespunzător din lista de membri obținută prin cererea XHR. Astfel, fiecare div cu carti va avea membrul corespunzător afișat sub ele.

Asigură-te că selectezi corect div-ul `#carti_jucator` și clasele sau identificatorii pentru a afișa membrii în locurile potrivite în codul tău.



```
xhr.open("GET",
'http://localhost:5000/lobbyMembers/' +
document.getElementById("lobby_name").innerHTML)
xhr.onload = function () {
  console.log(xhr.response)
  vec = JSON.parse(xhr.response)
  console.log(vec)
  var addedMembers = []; // Variabila pentru a tine
```



```

evidenta membrilor adaugati
    for (x in vec) {
        console.log(vec[x])
        if (!addedMembers.includes(vec[x])) { //
Verificam daca membrul exista deja in lista
            console.log(vec[x])
            var span = document.createElement("span");
            span.innerHTML = vec[x];

document.getElementsByClassName("membrii")
[0].appendChild(span);
            addedMembers.push(vec[x]); // Adaugam
membrul in lista de membri adaugati
        }
    }
}

xhr.send()
</script>
<body>

```

```

<h1 class="lobby-name">{{.LobbyName}}</h1>

```

```

<div class="membrii">
    <h2>Membrii</h2>
</div>
<div id = "container_mare">
<div id="container_joc" style="display: flex">
    <div id = "carti_intoarse_stanga">
        
        
        
        
        

```

```
  
  
  
</div>
```

```
<div id = "carti_intoarse_fata">  
    
    
    
    
    
    
    
    
    
</div>
```

```
<div id = "carti_intoarse_dreapta">  
    
    
    
    
    
    
  
```

```

        
        
    </div>
    <div id="carti_jucator" style=""
class="carte_de_joc">
        
        
        
        
        
        
        

```

</div> vreau fiecare membru (maxim 4) sa apara sub fiecare div: jucatorul care are deschisa pagina sa apara sub carti jucator, si restul sa apara sub carti intoarse



Pentru a afișa fiecare membru sub fiecare div corespunzător, trebuie să modificați bucla `for` astfel încât să adauge membrii în locurile potrivite. Iată o modificare a codului în concordanță cu cerința dvs.:

javascript

 Copy code

```

xhr.open("GET",
'http://localhost:5000/lobbyMembers/' +
document.getElementById("lobby_name").innerHTML);
xhr.onload = function () {
console.log(xhr.response); var vec =

```

```

JSON.parse(xhr.response); console.log(vec);
var addedMembers = []; // Variabila pentru a
tine evidenta membrilor adaugati var
currentPlayer =
document.getElementById("lobby_name").innerHTML;
// Numele jucatorului curent for (x in vec) {
console.log(vec[x]); if
(!addedMembers.includes(vec[x])) { //
Verificam daca membrul exista deja in lista
console.log(vec[x]); var span =
document.createElement("span"); span.innerHTML
= vec[x]; if (vec[x] === currentPlayer) { //
Membrul curent
document.getElementById("carti_jucator").appendCl
} else { // Membru diferit de cel curent
document.getElementById("carti_intoarse_stanga").
document.getElementById("carti_intoarse_fata").a
document.getElementById("carti_intoarse_dreapta")
} addedMembers.push(vec[x]); // Adaugam
membrul in lista de membri adaugati } } };
xhr.send();

```

În acest exemplu, am adăugat o variabilă `currentPlayer` care stochează numele jucătorului curent obținut din elementul cu id-ul "lobby\_name". Apoi, în bucla `for`, verificăm dacă membrul curent este jucătorul curent și adăugăm span-ul corespunzător sub div-ul `#carti_jucator`. În caz contrar, adăugăm clone ale span-ului sub div-urile `#carti_intoarse_stanga`, `#carti_intoarse_fata` și `#carti_intoarse_dreapta`.

Asigurați-vă că aveți un identificator unic pentru fiecare div în care doriți să afișați membrii și modificați codul în consecință pentru a selecta acele div-uri corespunzătoare.

De asemenea, rețineți că codul furnizat este o schiță și ar putea necesita ajustări suplimentare în funcție de structura HTML și logica aplicației dvs.



vreau sa imi apara membrul care e logat sub cari jucator, iar ceilalti jucatori(daca sunt conectati) sa apara sub carti intoarse fata/spate/dreapta



Dacă doriți ca membrul care este conectat să apară sub #carti\_jucator, iar ceilalți membri să apară sub #carti\_intoarse\_stanga, #carti\_intoarse\_fata și #carti\_intoarse\_dreapta, puteți modifica codul în consecință. Iată o versiune modificată a codului:

javascript

Copy code

```
xhr.open("GET",
'http://localhost:5000/lobbyMembers/' +
document.getElementById("lobby_name").innerHTML);
xhr.onload = function () {
console.log(xhr.response); var vec =
JSON.parse(xhr.response); console.log(vec);
var addedMembers = []; // Variabila pentru a
ține evidența membrilor adăugați var
currentPlayer =
document.getElementById("lobby_name").innerHTML;
// Numele jucatorului curent for (x in vec) {
console.log(vec[x]); if
(!addedMembers.includes(vec[x])) { //
Verificam dacă membrul există deja în lista
console.log(vec[x]); var span =
document.createElement("span"); span.innerHTML
= vec[x]; if (vec[x] === currentPlayer) { //
Membrul curent
document.getElementById("carti_jucator").appendCl
} else { // Membru diferit de cel curent
document.getElementById("carti_intoarse_stanga").
document.getElementById("carti_intoarse_fata").ap
document.getElementById("carti_intoarse_dreapta")
} addedMembers.push(vec[x]); // Adăugăm
```

```
membrul in lista de membri adaugati } } };  
xhr.send();
```

În acest caz, membrul curent (jucătorul conectat) este adăugat sub `#carti_jucator`, iar ceilalți membri sunt adăugați sub div-urile respective `#carti_intoarse_stanga`, `#carti_intoarse_fata` și `#carti_intoarse_dreapta`.

Asigurați-vă că aveți un identificator unic pentru fiecare div în care doriți să afișați membrii și modificați codul în consecință pentru a selecta acele div-uri corespunzătoare.

Vă rugăm să rețineți că codul furnizat este o schiță și ar putea necesita ajustări suplimentare în funcție de structura HTML și logica aplicației dvs.

---