



**UNIVERSITATEA DIN BUCUREȘTI**

**FACULTATEA DE MATEMATICĂ ȘI  
INFORMATICĂ**



**SPECIALIZAREA MATEMATICĂ**

**Lucrare de licență**

# **QuizzApp, Aplicație de Quiz-uri**

**Absolvent**

**Dima Alexandru**

**Coordonator științific**

**Lect. Dr. Iulia Banu Demergian**

**București, iulie 2022**

## Rezumat

Pentru a concepe o aplicație web avem nevoie de componentele sale esențiale: HTML - pentru crearea paginilor, CSS - pentru stilizarea paginilor și JavaScript-pentru implementarea diverselor funcționalități utilizate în aplicației web. De asemenea, adăugând o bază de date, precum MySQL, putem gestiona, modifica și utiliza datele aplicației web într-o manieră organizată și eficientă.

În lucrarea mea de licență doresc să creez o aplicație web intitulată **QuizzApp**, care prezintă o pagină principală cu informații generale, un meniu în care utilizatorul își poate crea un cont și oportunitatea de a răspunde la o largă varietate de întrebări din diverse domenii. Întrebările sunt categorizate pe dificultate, fiind pe de o parte ușor de accesat, iar pe de altă parte o provocare pentru cei mai experimentați.

“Învățăm, aprofundăm și ne distrăm”. Aceasta ar fi aplicația **QuizzApp** descrisă în câteva cuvinte.

## Abstract

In order to create a web application we need its essential components: HTML-used for creating the pages, CSS-used for pages appearance and JavaScript-used for the variety of functionalities used in the web application. Also, using a database, like MySQL, we can manage, modify and use the web application data in an organized and efficient way.

In my License paper I wish to create a web application called **QuizzApp**, which presents a main page that contains general information, a menu where the user can create an account and the opportunity to answer to a large variety of questions from different categories. The questions are classified by difficulty, which can be easy to access or they can potentially become a challenge for those who are more experienced.

“We learn, improve and have fun”. This is **QuizzApp** described in a few words.

# Cuprins

Capitolul 1.....	4
Introducere.....	4
1.1 Scopul lucrării.....	4
1.2 Obiective.....	4
1.3 Structura lucrării.....	4
1.4 Motivația personală.....	5
1.5 Istoria aplicațiilor web.....	5
Capitolul 2.....	6
Preliminarii.....	6
2.1 Tehnologii folosite.....	6
2.1.1 HTML.....	6
2.1.2 CSS.....	6
2.1.3 JavaScript.....	6
2.1.4 MySQL.....	7
2.2 Crearea serverului.....	7
2.3 Diagrama conceptuală.....	7
Capitolul 3.....	9
Implementare.....	9
3.1 Validarea datelor.....	9
3.1.1 Validarea la autentificare.....	9
3.1.2 Validarea la înregistrare.....	10
3.2 Meniul de chestionare.....	12
3.3 Formatul chestionarelor.....	13
3.4 Pagina de istoric.....	15
3.5 Rutele aplicației.....	15
3.5.1 Rutele de navigare.....	15
3.5.2 Rute de resurse.....	16
Capitolul 4.....	18
Manual de utilizare.....	18
4.1 Pagina Principală.....	18
4.2 Pagina de autentificare.....	18
4.3 Pagina de înregistrarea a contului.....	19
4.4 Pagina utilizatorului.....	20
4.4.1 Pagina de chestionare.....	21
4.4.2 Pagina de profil.....	22
4.4.3 Pagina de istoric.....	22
Capitolul 5.....	24
Concluzii.....	24
5.1 Starea curentă a aplicației.....	24
5.2 Îmbunătățiri pentru versiunile următoare ale aplicației.....	24
Bibliografie.....	26

# Capitolul 1

## Introducere

### 1.1 Scopul lucrării

Scopul principal al lucrării de licență este de a gestiona o varietate largă de date într-un mod organizat, astfel încât utilizatorul să nu întâmpine dificultăți în accesarea diverselor funcționalități ale aplicației web. De asemenea, este un mod educativ și interactiv pentru persoanele pasionate de cultură generală care doresc să-și aprofundeze cunoștințele.

### 1.2 Obiective

În implementarea lucrării, am în vedere următoarele obiective:

- ➔ **Eficiență:** timpul efectuat pentru diversele funcționalități să fie cât mai scurt posibil.
- ➔ **Simplitate :** aplicația prezintă butoane/specificații ușor de înțeles pentru fiecare utilizator.
- ➔ **Generalitate:** datorită faptului ca vor exista mai multe domenii, fiecare cu o multitudine de întrebări, consider că accesul la aplicația web va fi realizat de persoane de diferite vârste. Sunt de părere că pentru cei tineri este o modalitate de a acumula noi cunoștințe pe diverse domenii iar pentru cei cu mai multă experiență va prezenta o provocare.

### 1.3 Structura lucrării

Lucrarea se va împărți în următoarele capitole:

1. **Introducere:** oferă o perspectivă generală asupra lucrării de licență.
2. **Preliminarii:** oferă informații despre resursele aplicației.
3. **Implementare:** detaliază modul în care programul a fost conceput și eventual optimizările adăugate pe parcurs.
4. **Manual de utilizare:** detaliază modul de funcționare al aplicației pentru ca utilizatorul să nu întâmpine dificultăți.
5. **Concluzii:** rezumă informațiile prezentate și eventualele modificări finale.

## 1.4 Motivația personală

Am ales această temă deoarece apreciez mult aplicațiile pe această direcție și mi-am dorit să dezvolt o aplicație similară care poate fi accesată și îndrăgită de persoane de diferite vârste. Cercetarea pe acest subiect m-a ajutat să aprofundez cunoștințele dobândite de-a lungul facultății.

## 1.5 Istoria aplicațiilor web

Inițial, aplicațiile web aveau propriul program client, având rol de interfață utilizator, care trebuia să fie instalat separat pe calculatorul utilizatorului. Pentru o actualizare pe server este necesară o actualizare a fiecărei aplicație client instalate, ceea ce provoacă costuri suplimentare.

În 1995 a fost introdus **JavaScript**-ul de către Netscape, acesta fiind folosit pentru programatorii care doreau să adauge elemente dinamice interfeței utilizatorului. Până la apariția acestui scripting client, toate datele trebuie să fie verificate și trimise spre server pentru a fi procesate, iar ulterior, rezultatele să fie trimise în pagini **HTML** care pot fi accesate de către client.

În 1996, apariția player-ului de animație vectorial **Flash**, introdus de către Macromedia, a contribuit la dezvoltarea aplicațiilor web datorită posibilității de a insera animații în paginile web. De asemenea, limbajul de scripting a evoluat, astfel interacțiunile în partea clientului puteau fi programate fără a menține o comunicare constantă cu serverul.

În 1999, obiectul **XMLHttpRequest** a fost introdus în Internet Explorer 5 ca un obiect ActiveX. În acea perioadă, atât **JavaScript** cât și **XML** erau deja dezvoltate, iar **AJAX** încă nu fusese inventat [5].

# Capitolul 2

## Preliminarii

### 2.1 Tehnologii folosite

Aplicația este construită folosind următoarele: **HTML**, **CSS**, **JavaScript**, **Node JS** și **MySQL**.

#### 2.1.1 HTML

Acronimul **HTML** (*Hypertext Markup Language*), înseamnă „limbaje de marcare hipertext”. Acesta se bazează pe un limbaj de marcare pentru a crea documente care pot fi distribuite pe Internet.

**HTML**-urile pot fi create și modificate folosind un editor de text. Pentru a scrie un document **HTML** avem nevoie de un cuvânt cheie care este introdus între semnele (<>) și funcționează ca comenzi pentru formate de text, linkuri, imagini, etc [1].

#### 2.1.2 CSS

**CSS**-ul reprezintă un limbaj de stilizare al elementelor **HTML**, în special al tagurilor **HTML**. Denumirea de **CSS** provine de la expresia “Cascading Style Sheets”. Cu ajutorul **CSS**-ului putem modifica stilul și culoarea unui text, putem modifica culoarea fundalului, putem poziționa elemente în diferite moduri, etc. Stilizarea poate fi incorporată în pagina **HTML** sau importată dintr-un fișier extern [2].

#### 2.1.3 JavaScript

**JavaScript**-ul este un limbaj dinamic de programare, folosit în aplicațiile web, ce permite implementarea diverselor caracteristici care nu pot fi realizate doar de **HTML** și **CSS**. În general este folosit la crearea unor funcționalități, crearea unor legături între paginile **HTML** sau chiar crearea legăturii unei baze de date în domeniul aplicației web.

**JavaScript**-ul permite utilizarea formulelor matematice (biblioteca **Math**), poate modifica proprietăți **CSS** (spre exemplu, culoarea sau mărimea unui text) sau chiar să culeagă tot felul de

date din diverse zone. Acesta este un limbaj flexibil care ajută la realizarea principalelor funcționalități ale unei aplicații web [3].

### 2.1.4 MySQL

**MySQL** este un sistem de gestionare a bazelor de date relaționale cu sursă deschisă, susținut de Oracle, bazat pe limbajul “Structured Query Language” (**SQL**). **MySQL** poate fi utilizat în marea majoritate a platformelor, incluzând Linux, UNIX și Windows. Acesta se poate utiliza într-o gama largă de aplicații, dar cel mai des este asociat cu aplicațiile web. Cu ajutorul acestui limbaj putem stoca și manipula datele aplicației web [4].

## 2.2 Crearea serverului

Am creat fișierul **server.js**, în care am inițializat serverul cu ajutorul lui **Node JS**.

Utilitatea serverului este de a realiza legături între paginile aplicației și de a distribui date pe diverse pagini. Totodată, ajută la adăugarea datelor noi prin intermediul paginilor existente.

## 2.3 Diagrama conceptuală

Deoarece folosesc o bază de date pentru a stoca diversele date ale aplicației, am conceput diagrama conceptuală, care prezintă legăturile dintre tabelele folosite.

Tabele folosite sunt:

- Users - salvează datele utilizatorului.
- Players - salvează datele utilizatorului din pagina de utilizator.
- Categories - salvează datele categoriilor de întrebări.
- Favorites - rezolvă relația Many to Many dintre tabelele Players și Categories.
- Questions - salvează întrebările pentru fiecare categorie.
- Answers - salvează răspunsurile pentru fiecare întrebare.
- Histories - salvează data și ora completării unui chestionar de întrebări.
- Statistics - rezolvă relația Many to Many dintre tabelele Histories și Players.
- HintButtons - salvează numărul de **hint-uri** pe care îl are utilizatorul.

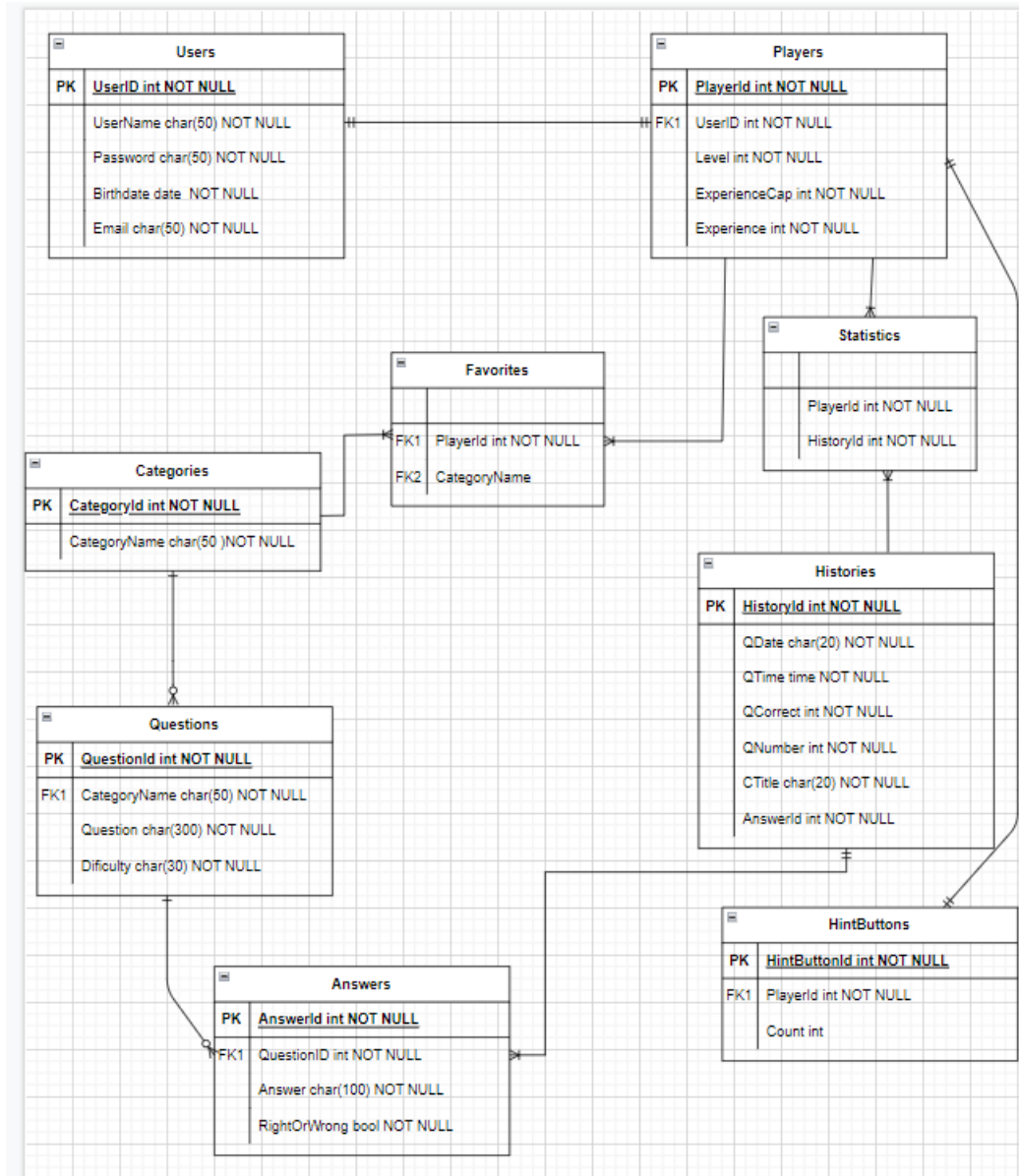


Figura 2.1 Diagrama conceptuală. Aceasta a fost creată folosind <https://draw.io/>.



# Capitolul 3

## Implementare

### 3.1 Validarea datelor

Este important ca atunci când dorim să ne autentificăm / înregistrăm să se verifice corectitudinea datelor. Dacă nu facem această verificare există posibilitatea ca utilizatorul să efectueze diverse acțiuni care pot cauza erori în aplicație.

#### 3.1.1 Validarea la autentificare

În momentul în care utilizatorul dorește să se autentifice în aplicație este esențial să verificăm datele introduse. Datele introduse de aceasta sunt căutate în baza de date cu ajutorul metodei **fetch**. În interiorul acestei metode verificăm dacă datele introduse corespund cu cele salvate în baza de date. Un mesaj corespunzător va fi afișat în cazul în care datele introduse de acesta nu corespund cu cele din baza de date, datele introduse de acesta vor fi șterse și va fi nevoit să reintroducă datele.

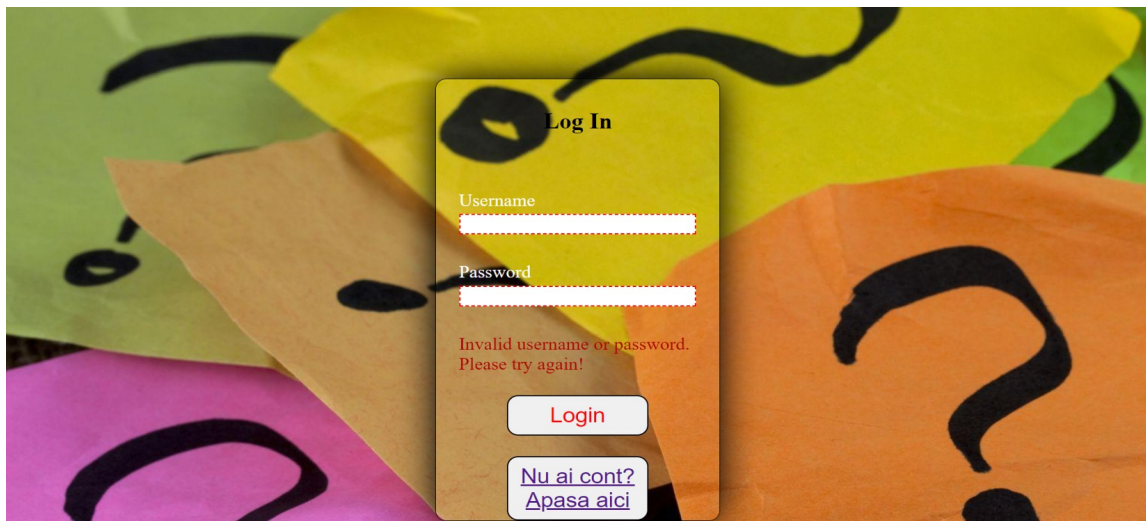


Figura 3.1 Validarea în momentul autentificării

Lipsa verificării sugerează accesarea paginii de profil indiferent de ceea ce introduce utilizatorul și poate cauza funcționarea eronată a aplicației.

### 3.1.2 Validarea la înregistrare

În momentul în care utilizatorul se înregistrează în aplicație este esențial să verificăm datele introduse. Datele introduse sunt căutate în baza de date cu ajutorul metodei **fetch**. În interiorul acestei metode verificăm dacă utilizatorul a completat fiecare **input**, dacă numele de utilizator introdus este diferit de utilizatorii deja existenți în baza de date și dacă parola respectă formatul impus (lungimea să fie mai mare sau egală de 6 caractere). Un mesaj corespunzător va fi afișat utilizatorului în cazul în care una sau mai multe din mențiunile anterioare nu este respectată și va fi nevoit să reintroducă datele.

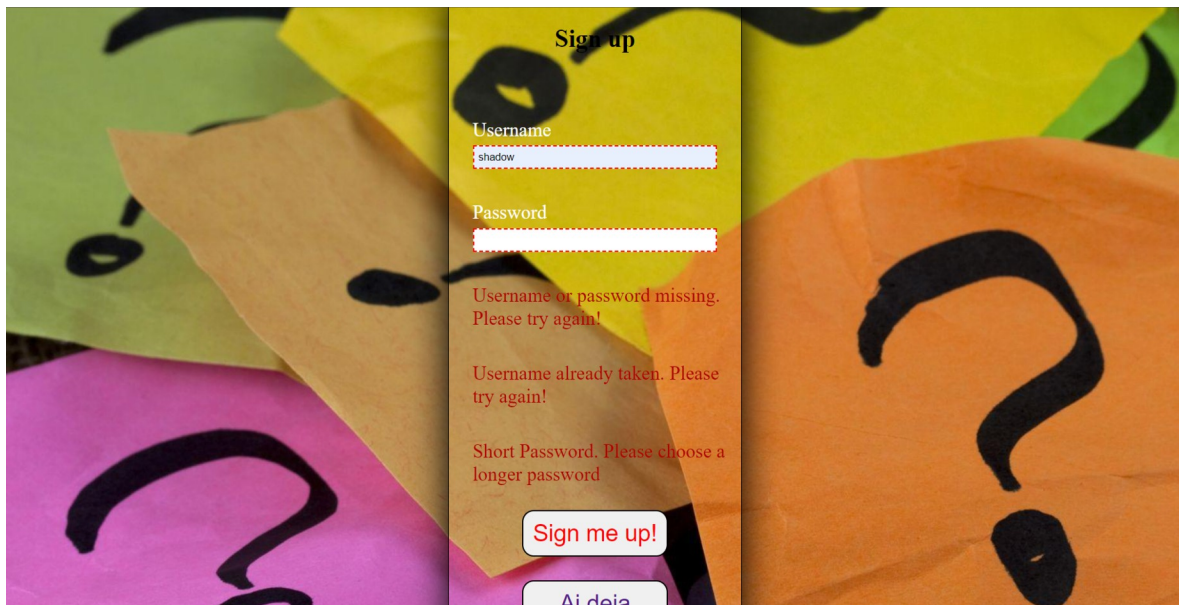


Figura 3.2 Exemplu 1 de validare la înregistrare

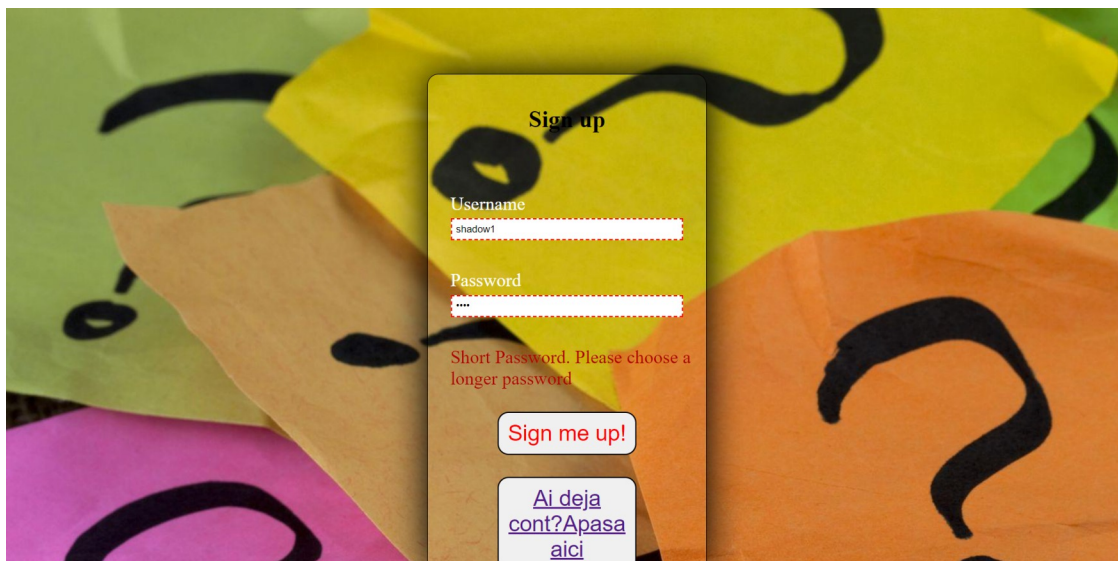


Figura 3.3 Exemplu 2 de validare la înregistrare

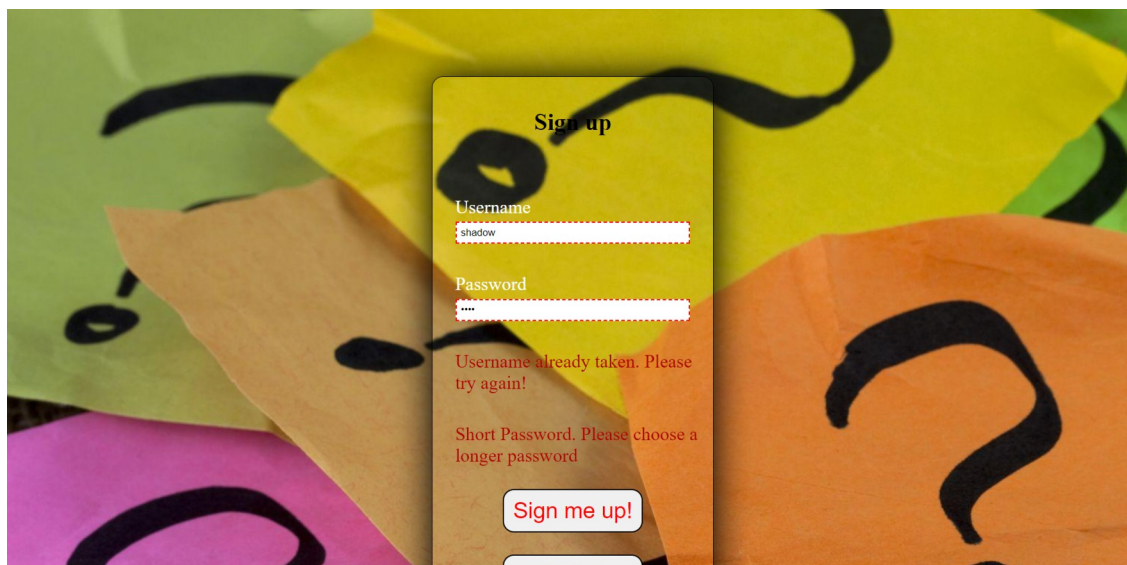


Figura 3.4 Exemplu 3 de validare la înregistrare

Lipsa acestei verificări conduce la crearea unor conturi invalide care vor afecta conținutul aplicației.

## 3.2 Meniul de chestionare

Meniul de chestionare este alcătuit din 2 butoane și un **div** (element HTML) care conține titlul categoriei, respectiv butoanele cu dificultăți. Datorită implementării în **JavaScript**, acestea vor fi mereu actualizate în meniu la fiecare încărcare a paginii.

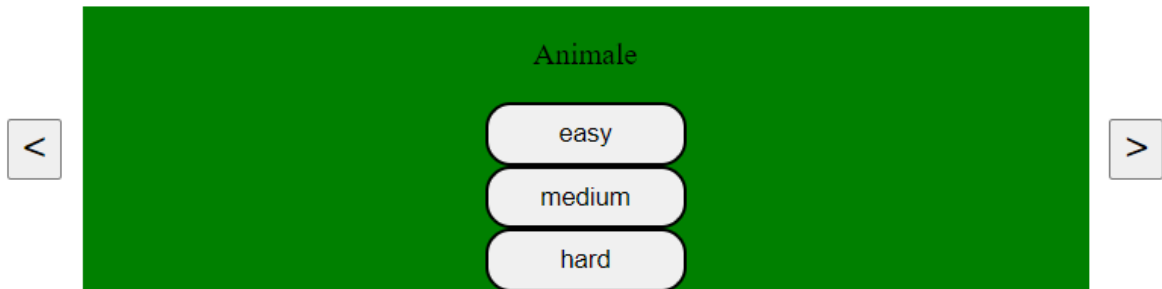


Figura 3.5 Meniul de chestionare

Cele două butoane ajută la schimbarea categoriei curente. Permutarea acestor **div** -uri se face cu ajutorul claselor. Fiecare **div** prezintă clasele “option”, “current” sau “hide”. Clasa **option** adaugă stilizare meniului, clasa **current** specifică categoria curent vizibilă pe ecran iar clasa **hide** specifică categoriile care nu sunt vizibile curent.

La apăsarea săgeții din stânga sau din dreapta se schimbă categoria curentă, schimbându-se și clasele.

```
<div id="Animale" class="option current">...</div>
<div id="Fotbal" class="option hide">...</div>
<div id="Matematica" class="option hide">...</div>
```

Figura 3.6 Vizualizarea categoriei curente

```
<div id="Animale" class="option hide">...</div>
<div id="Fotbal" class="option hide">...</div>
<div id="Matematica" class="option current">...</div>
```

Figura 3.7 Vizualizarea categoriei curente apăsând pe săgeata din stânga(Categoria inițială:  
Animale)

```
<div id="Animale" class="option hide">...</div>
<div id="Fotbal" class="option current">...</div>
<div id="Matematica" class="option hide">...</div>
```

Figura 3.8 Vizualizarea categoriei curente apăsând pe săgeata din dreapta(Categoria inițială: Animale)

### 3.3 Formatul chestionarelor

Formatul chestionarelor este realizat după un șablon modificat după fiecare întrebare. Se generează butonul care începe chestionarul, întrebările și răspunsurile, opțiunea de a ajunge la următoarea întrebare și butonul de finalizare. Pentru fiecare set de întrebări și răspunsuri, se generează taguri **HTML** și după ce este ales un răspuns (în acest moment, se actualizează numărul de răspunsuri corecte, în funcție de corectitudinea răspunsului) apare butonul care afișează următoarea întrebare. La apăsarea acestui buton, întrebarea și răspunsurile curente sunt înlocuite de următorul set de întrebări și răspunsuri. Analog meniului de chestionare, funcționarea elementelor este influențată de clasele sale. Datorită lor, putem afișa doar anumite elemente, iar celelalte să fie ascunse și vice versa.

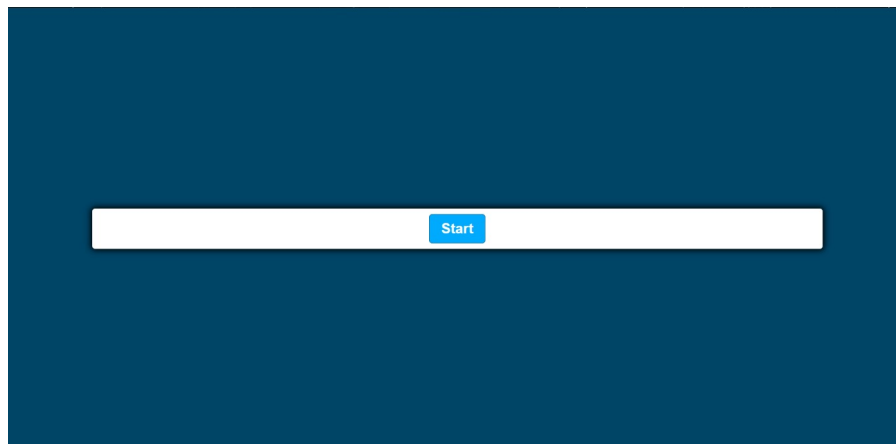


Figura 3.9 Pagina de chestionare înainte de începerea chestionarului

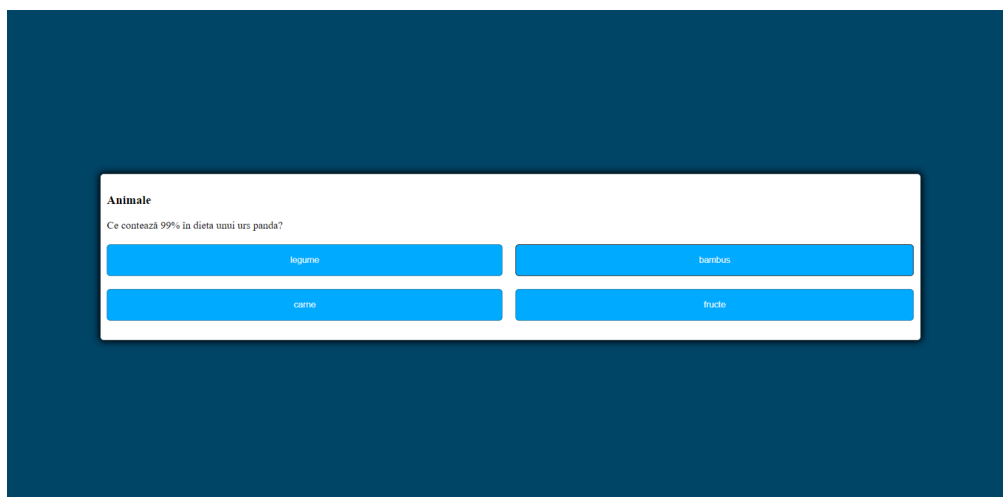


Figura 3.10 Afișare după apăsarea butonului Start

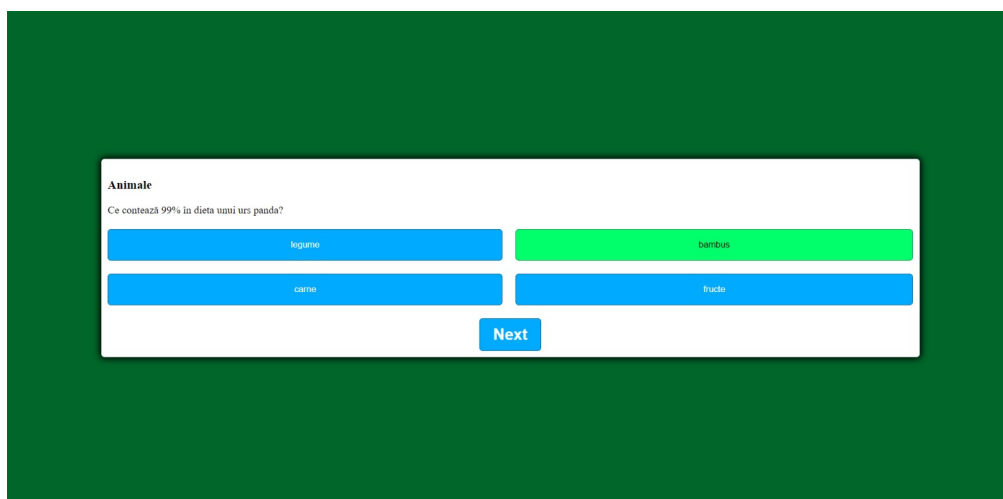


Figura 3.11 Apariția butonul care afișează următoarea întrebare

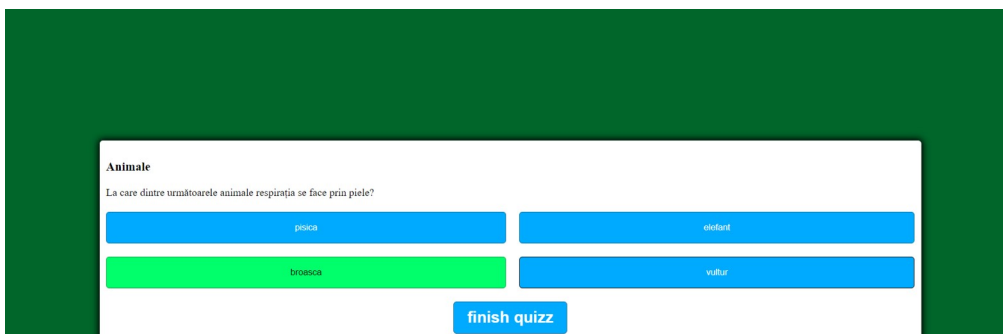


Figura 3.12 Apariția butonului de finalizare a chestionarului

În funcție de răspunsurile date de utilizator, acesta va obține experiență. Răspunsurile, atât corecte cât și cele greșite, vor oferi puncte de experiență, însă întrebările al căror răspuns este greșit vor oferi mai puțină experiență decât cele cu răspuns corect.

## 3.4 Pagina de istoric

Pagina de istoric este actualizată la fiecare completare de chestionar. Se salvează în baza de date, folosind apeluri ale metodei **fetch**, numele categoriei din care a fost rezolvat chestionarul, data, ora, numărul de răspunsuri corecte și numărul total de întrebări ale chestionarului. Pentru a determina data și ora am folosit apeluri ale funcțiilor de manipulare a datelor calendaristice(de exemplu, *getDate()*, *getMonth()*, *getFullYear()*).

## 3.5 Rutele aplicației

Rutele unei aplicații sunt utile pentru a crea paginile la care utilizatorul are acces și pentru a le alimenta cu diverse date. În aplicație, se folosesc două tipuri de rute: rute de navigare și rute care conțin resurse.

### 3.5.1 Rutele de navigare

Rutele de navigare ajută utilizatorul să primească acces la paginile aplicației. Se creează cereri în fișierul **JavaScript** pentru a accesa paginile statice ale aplicației.

Exemplu de cod pentru accesul la pagina principală folosind ruta “/”:

```
app.get("/", (_, res) => {  
    res.sendFile(__dirname + "/html/paginaPrincipala.html")  
});
```

În cazul în care ruta nu este creată în fișierul JavaScript corespunzător va fi accesată pagina de eroare 404, care transmite utilizatorului că pagina accesată nu există.

Schimbarea rutelor a fost realizată în două moduri: prin metoda **replace** accesată din **location** și prin adăugarea atributului **href**. De exemplu, în pagina principală putem accesa pagina de autentificare, respectiv de înregistrare utilizând meniul.

### 3.5.2 Rute de resurse

Rutele de resurse sunt cele care generează date pentru diverse funcționalități ale aplicației. În acestea, putem salva date pe care să le utilizăm ulterior (de exemplu, întrebările și răspunsurile selectate dintr-o categorie aleasă de utilizator), modifică date existente (de exemplu, bara de experiență și nivelul utilizatorului) sau adaugă unele noi în baza de date (de exemplu, în pagina de înregistrare, am salvat datele necesare pentru crearea unui cont).

Exemplu de rută unde sunt generate întrebări pentru chestionar:

```
app.get("/dataquizz/:Category/:Difficulty",(req, res) => {
  let quizz = `select question as question, answer as answer, RightOrWrong as correct
    from QUESTIONS q,ANSWERS a, CATEGORIES c where q.idQuestion = a.idQuestion
    and c.Idcategory = q.Idcategory and c.categoryName = '${req.params.Category}'
    and q.difficulty = '${req.params.Difficulty}';`
  db.query(quizz, (error, result) => {
    let v = [];
    for(let i = 0; i < result.length; i+= 4)
    {
      v.push({
        question: result[i].question,
        answers: [{
          text: result[i].answer, correct: result[i].correct
        }, {
          text: result[i+1].answer, correct: result[i+1].correct
        }, {
          text: result[i+2].answer, correct: result[i+2].correct
        }, {
          text: result[i+3].answer, correct: result[i+3].correct
        }]
      })
    }

    error ? res.send({message: "Something went wrong..."}) : res.send(v);})
});
```



Exemplu de rută unde sunt modificate date existente

```
app.put("/updatedataPlayer/:PlayerLevel/:ExperienceCap/:Experience/:idPlayer",(req, res) => {
  let update = `UPDATE PLAYERS
  SET  PlayerLevel = '${req.params.PlayerLevel}', ExperienceCap=
  '${req.params.ExperienceCap}', Experience = '${req.params.Experience}'
  WHERE idPlayer = '${req.params.idPlayer}';`
  db.query(update, (error, result) => {
    error ? res.send({message: "Something went wrong..."}) : res.send(result);})
})
```

Exemplu de rută unde adaug date noi:

```
app.get("/checkregister/:username/:password",(req, res)=> {
  let register = `insert into users(userName,password) values ('$
  {req.params.username}','${req.params.password}')`;
  db.query(register, (error, result) => {
    error ? res.send({message: "Something went wrong..."}) : res.send(result);
  })
})
```

Aceste rute conțin comenzi SQL care mă ajută să manipulez datele în funcție de nevoie. Creez o cerere la baza de date care va trimite un rezultat la ruta selectată.

# Capitolul 4

## Manual de utilizare

### 4.1 Pagina Principală

Această pagină conține:

- Un meniu, care conține opțiunile de autentificare sau înregistrare.
- O scurtă prezentare a aplicației QuizzApp.
- O listă cu categoriile curent existente, care prezintă o scurtă prezentare.

Lista categoriilor existente este actualizată în timp real folosind JavaScript.

### 4.2 Pagina de autentificare

Această pagină poate fi accesată apăsând pe butonul din meniu creat în pagina principală. În această pagină, utilizatorul trebuie să introducă numele de utilizator și parola pentru a se conecta în aplicație. În cazul în care numele de utilizator sau parola nu există în baza de date, utilizatorul nu va avea acces la pagina de profil și este nevoit să reintroducă datele corect.

Această pagină conține un **form**, alcătuit din următoarele taguri HTML: **label**, **input** și **buton**. **Label**-urile specifică ce date trebuie introduse în **input**-ul aflat sub ele, iar butoanele sunt pentru autentificare, respectiv pentru a comuta cu pagina de înregistrare a contului.

În momentul autentificării unui utilizator, se realizează o căutare în baza de date atât a numelui de utilizator cât și a parolei introduse în zonele de **input** respective și se verifică existența acestora.

În cazul în care utilizatorul nu are un cont creat, acesta poate accesa pagina de înregistrare, unde își poate crea un cont.

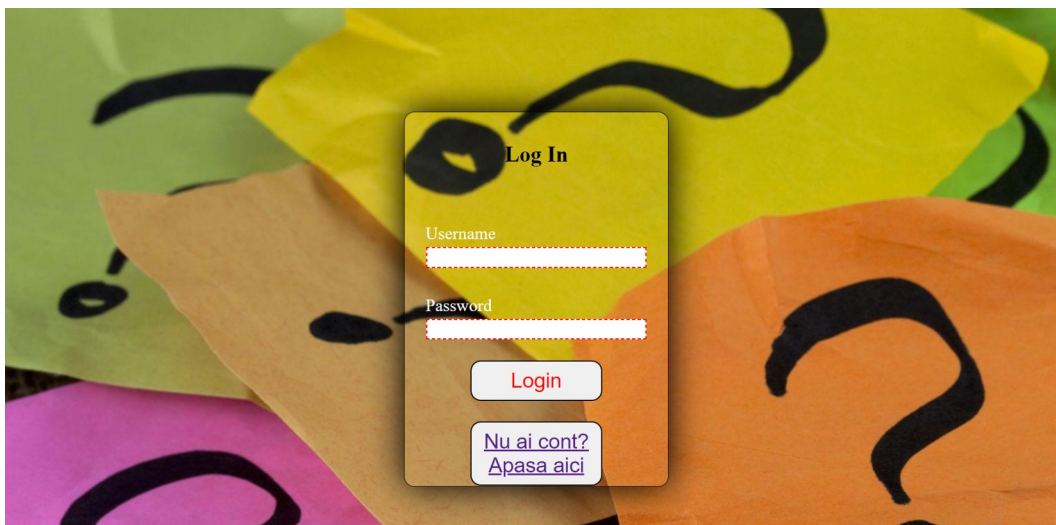


Figura 4.1. Pagina de autentificare

### 4.3 Pagina de înregistrarea a contului

Această pagină poate fi accesată de asemenea apăsând pe butonul respectiv din meniul creat în pagina principală. În această pagină, utilizatorul este nevoit să introducă un nume de utilizator și o parolă pentru a crea un cont, astfel având acces la pagina sa de profil.

Această pagină conține un **form**, alcătuit din următoarele taguri HTML: **label**, **input** și **buton**. Label-urile specifică ce date trebuie introduse în input-ul aflat sub ele, iar butoanele sunt pentru înregistrarea contului, respectiv pentru a comuta cu pagina de autentificare.

În momentul adăugării unui utilizator, în baza de date, se inserează informațiile necesare pentru crearea unui cont. După ce contul este creat, utilizatorul se poate întoarce la pagina de autentificare unde se poate conecta în aplicație.

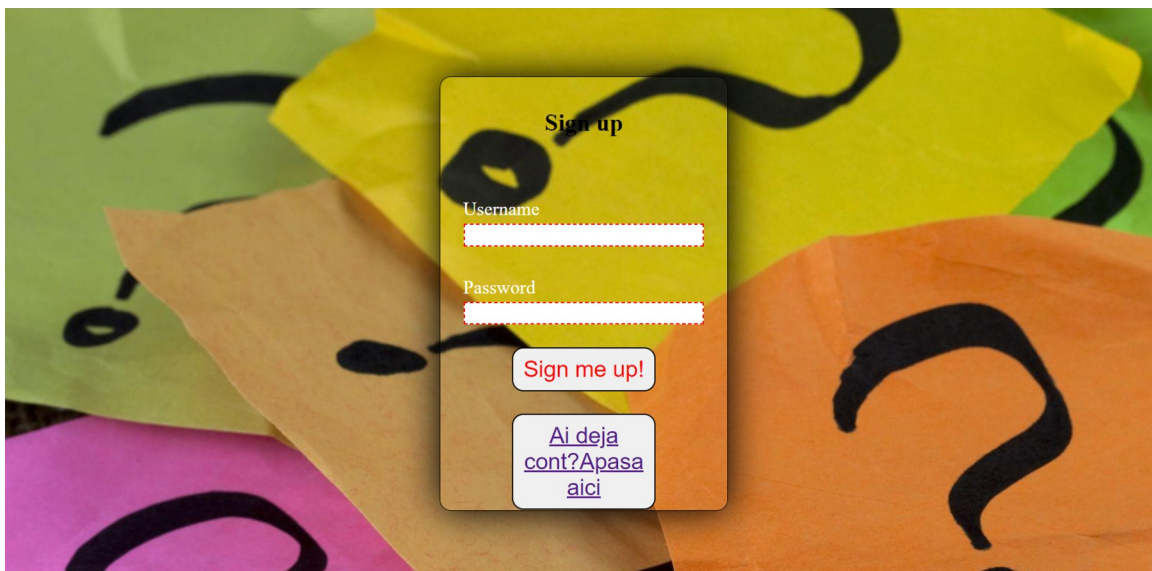


Figura 4.2. Pagina de înregistrare

## 4.4 Pagina utilizatorului

Această pagină poate fi accesată cu succes în momentul în care utilizatorul se conectează cu un cont deja existent.

Fiecare utilizator își poate vizualiza detaliile contului (numele, nivelul, bara de experiență, avatarul și alte diverse opțiuni) într-o zonă specială. Numele, nivelul și bara de experiență sunt preluate cu ajutorul bazei de date, zona de avatar (Figura 3.1) este creată folosind taguri HTML (**select**, **option**, respectiv **img**), iar butoanele existente sunt: Profil, Istoric și Deconectare.

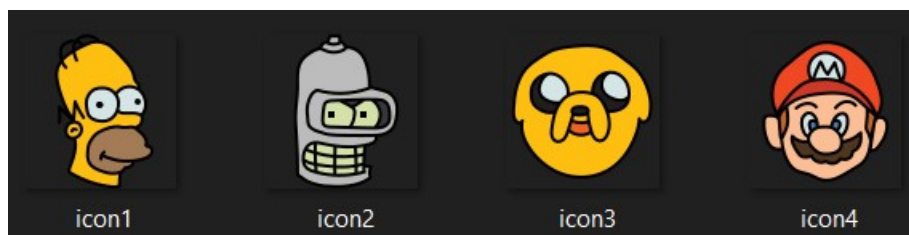


Figura 4.3. Exemple de iconițe care pot fi alese pentru avatar [6]

Pentru a căpăta experiență, utilizatorul trebuie să completeze diverse chestionare. În funcție de performanța sa, va acumula puncte de experiență. În momentul în care utilizatorul

colectează destule puncte de experiență, acesta va avansa în nivel. În această situație, acesta va câștiga recompense ce îl vor ajuta mai târziu.

#### 4.4.1 Pagina de chestionare

Accesul la această pagină este realizat prin selectarea unui domeniu din pagina de utilizator sau accesarea opțiunii **Daily Challenge**.

Chestionarele sunt categorizate pe mai multe domenii, fiecare având o dificultate: ușor, mediu sau greu. În funcție de dificultatea aleasă, se va genera un chestionar cu întrebările aferente acestuia. Utilizatorul va răspunde la întrebări apăsând pe butonul corespunzător răspunsului corect. Dacă răspunsul dat este corect, fundalul își va schimba culoarea în verde și respectiv în roșu dacă răspunsul este incorect. În momentul în care este selectat un răspuns se vor întâmpla următoarele: dacă răspunsul este corect, fundalul și butonul își vor schimba culoarea în verde. Dacă răspunsul este greșit, răspunsul corect este aflat, iar fundalul și butonul își vor schimba culoarea în roșu. La finalizarea chestionarului, se va afișa numărul de întrebări pentru care utilizatorul a dat răspunsul potrivit.

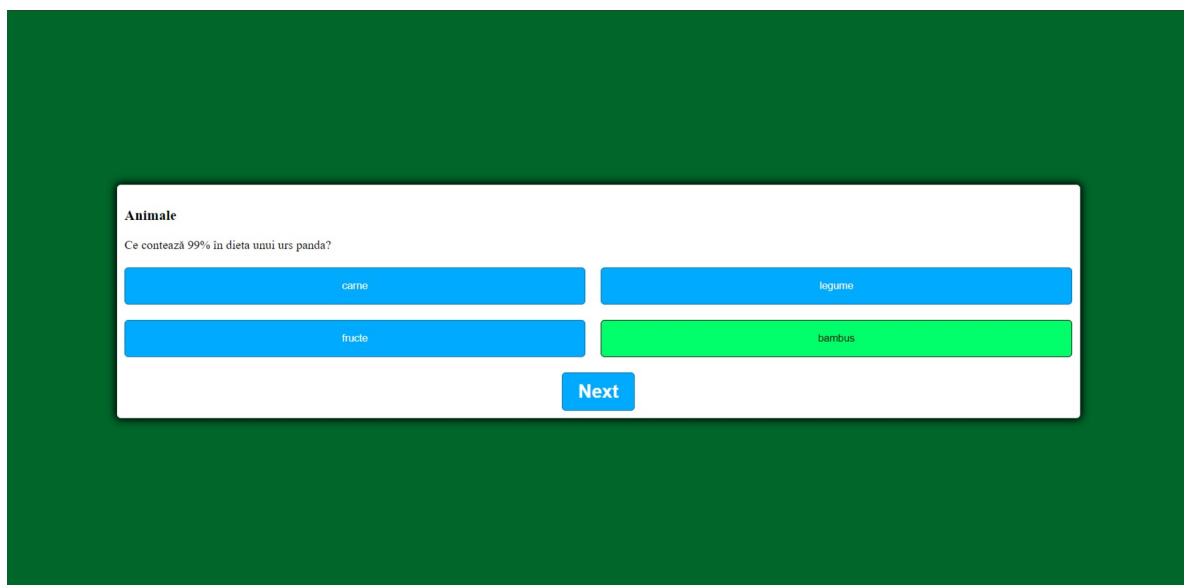


Figura 4.4. Exemplu la afișare de răspuns corect

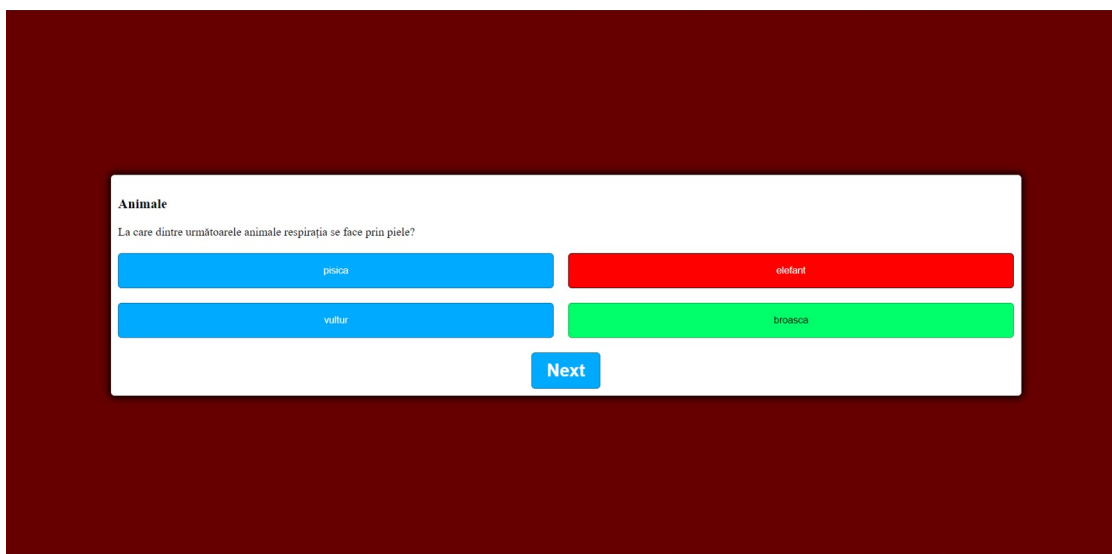


Figura 4.5. Exemplu la afișare de răspuns greșit

Opțiunea **Daily Challenge** reprezintă o provocare în care utilizatorul poate participa zilnic și în care poate răspunde la întrebări din toate domeniile existente. Spre deosebire de un chestionar pe un domeniu specific, numărul de întrebări este dublat, întrebările pot fi de orice dificultate, iar punctele de experiență acumulate din această provocare sunt exponențial crescute.

#### 4.4.2 Pagina de profil

În această pagină vor fi afișate informații detaliate despre utilizatorul autentificat în sesiunea curentă. Aici, acesta poate să-și editeze profilul, să adauge informații în plus de cele impuse la conectarea în aplicație (spre exemplu, țara de proveniență, adresa de email, diverse hobby-uri, ziua nașterii și multe altele). De asemenea, vor fi afișate și informații despre realizările utilizatorului (numărul de chestionare parcurse, domeniul preferat de chestionar, numărul de răspunsuri corecte sau greșite etc).

#### 4.4.3 Pagina de istoric.

În această pagină vor fi afișate detalii despre numele domeniului din care face parte chestionarul, data și ora finalizării acestuia, numărul de răspunsuri corecte și numărul de întrebări

total. Obiectivul acestei pagini este de a realiza o statistică asupra performanței utilizatorului din sesiunea curentă.

## User History

Latest Questionary finished				
Domeniu	Data	Ora	NrRaspunsuriCorecte	NrIntrebari
Animale	16-06-2022	16:03:55	3	3
Matematica	16-06-2022	16:27:00	3	3
Animale	16-06-2022	16:34:57	3	3

Figura 4.5. Pagina de istoric.

# Capitolul 5

## Concluzii

### 5.1 Starea curentă a aplicației

În prezent, aplicația prezintă o pagină principală cu detalii despre conținutul acesteia, metoda de înregistrare, respectiv autentificare și pagina utilizatorului. După ce utilizatorul își creează un cont sau se conectează cu un cont deja existent, acesta are oportunitatea să-și testeze cunoștințele asupra diverselor categorii de chestionare, poate participa la provocarea zilnică, să avanseze în nivel și să acumuleze diverse **achievement-uri**.

### 5.2 Îmbunătățiri pentru versiunile următoare ale aplicației

În versiunea curentă, respectiv în versiunile următoare ale aplicației voi încerca să adaug următoarele îmbunătățiri în aplicație:

- Adăugarea unui sistem în care utilizatorii își pot trimite cereri de prietenie.
- Implementarea unei funcțiuni în care doi utilizatori prieteni pot participa într-o competiție. Aceasta constă în selectarea unui domeniu și completarea unui chestionar pe domeniul ales. Utilizatorul cu cele mai multe răspunsuri corecte la acest chestionar va câștiga.
- Ajustarea paginii principale, respectiv cea de utilizatori, prin adăugarea zonei de recenzii. O recenzie poate fi adăugată doar de un utilizator autentificat, dar afișarea recenziilor va putea fi vizibilă și de cei neautentificați.
- Îmbunătățirea stilizării pe paginile existente.
- Opțiunea de **sugestie**, în care utilizatorii pot sugera diverse domenii și posibile întrebări din respectivul domeniu.
- Opțiunea de **hint**, cu ajutorul căreia utilizatorul primește un ajutor pentru a putea să răspundă la întrebări mai ușor. Acestea se pot obține la avansarea în nivel, la finalizarea provocării zilnice sau alte metode care vor apărea în versiunile următoare.



- Timp limită pentru a răspunde la întrebări iar punctele de experiență să fie influentate de timpul rămas(cu cât răspunsul corect este oferit mai rapid, cu atât se pot acumula mai multe puncte de experiență).
- Opțiunea de a vizualiza un chestionar rezolvat în pagina de istoric.

# Bibliografie

- [1] David Hemmendinger, HTML, <https://www.britannica.com/technology/HTML>, accesat 03/06/2022
- [2] Cameron McKenzie, CSS (cascading style sheets), <https://www.theserverside.com/definition/cascading-style-sheet-CSS> , accesat: 03/06/2022
- [3] Dillion Megida What is JavaScript? A Definition of the JS Programming Language, <https://www.freecodecamp.org/news/what-is-javascript-definition-of-js/> , accesat: 01/06/2022
- [4] Lindsay Moore, MySQL, <https://www.techtarget.com/searchoracle/definition/MySQL>, accesat: 01/06/2022
- [5] Nicolae Sfetcu, Istoria aplicațiilor web, <https://www.telework.ro/ro/istoria-aplicatiilor-web/> , accesat: 03/06/2022
- [6] Resursă pentru avatarele de tip icon, <https://icons8.com/icon/set/characters/doodle>, accesat: 02/06/2022
- [7] Documentație **MySQL**, <https://www.npmjs.com/package/mysql>, accesat: 15/06/2022
- [8] Resursă pentru întrebări folosite în baza de date <https://ro.jf-sjbrito.pt/84-best-animal-trivia-questions>, accesat: 15/06/2022
- [9] Govinda Sai, How to draw a star in HTML5 SVG?, <https://www.tutorialspoint.com/How-to-draw-a-star-in-HTML5-SVG>, accesat: 15/06/2022
- [10] Resursă pentru întrebări folosite în baza de date, <https://www.goal.com/en/news/best-football-quiz-questions-trivia-answers/1dfwcyp3388zg1lon8wlit8q42>, accesat: 16/06/2022
- [11] Resursă pentru întrebări folosite în baza de date, <https://www.cuemath.com/learn/fun-maths-questions/> , accesat: 16/06/2022