Table of Contents

[1. Introducere 4](#_Toc187518006)

[1.1 Descrierea generală a proiectului 4](#_Toc187518007)

[1.2 Funcționalități principale: 4](#_Toc187518008)

[1.3Tehnologii utilizate 4](#_Toc187518009)

[1.4 Obiectivul proiectului 5](#_Toc187518010)

[2. Arhitectura Sistemului 6](#_Toc187518011)

[2.1 Prezentare Generală 6](#_Toc187518012)

[2.2 Componentele Principale 6](#_Toc187518013)

[2.2.1 Frontend 6](#_Toc187518014)

[2.2.2 Backend 6](#_Toc187518015)

[2.2.3 Baza de Date 7](#_Toc187518016)

[2.3 Fluxul Aplicației 8](#_Toc187518017)

[2.4 Securitate 9](#_Toc187518018)

[Capitolul 3: Designul Aplicației 10](#_Toc187518019)

[3.1 Componentele Aplicației 10](#_Toc187518020)

[3.1.1 Frontend 10](#_Toc187518021)

[3.1.2 Backend 10](#_Toc187518022)

[3.2 Fluxul Aplicației 10](#_Toc187518023)

[3.3 Interfața Utilizatorului (UI) 11](#_Toc187518024)

[3.4 Designul Bazei de Date 11](#_Toc187518025)

[3.5 Funcționalitățile Aplicației 12](#_Toc187518026)

[3.5.1 Gestionarea Quizurilor 12](#_Toc187518027)

[3.5.2 Gestionarea Întrebărilor 12](#_Toc187518028)

[3.5.3 Evaluarea și Vizualizarea Rezultatelor 12](#_Toc187518029)

[Capitolul 4: Implementarea Aplicației 13](#_Toc187518030)

[4.1 Tehnologii Folosite 13](#_Toc187518031)

[4.1.1 Frontend 13](#_Toc187518032)

[4.1.2 Backend 13](#_Toc187518033)

[4.2 Structura Aplicației 14](#_Toc187518034)

[4.2.1 Frontend 14](#_Toc187518035)

[4.2.2 Backend 14](#_Toc187518036)

[4.3 Detalii Implementare 15](#_Toc187518037)

[4.3.1 Frontend - Componente și Funcționalități 15](#_Toc187518038)

[4.3.2 Backend - Endpoints și Funcționalități 15](#_Toc187518039)

# 1. Introducere

## 1.1 Descrierea generală a proiectului

Acest proiect reprezintă o aplicație web de tip quiz online, destinată să ofere o platformă interactivă pentru crearea și completarea de teste. Aplicația permite utilizatorilor să creeze quizuri cu întrebări multiple și să le evalueze pe baza răspunsurilor corecte. De asemenea, oferă un sistem de autentificare pentru a asigura securitatea și personalizarea experienței fiecărui utilizator.

**Scopul și Beneficiile Proiectului**

Scopul principal al acestui proiect este de a oferi o soluție eficientă și ușor de utilizat pentru organizarea și participarea la quizuri. Proiectul beneficiază atât administratorii, care pot crea și gestiona quizuri, cât și utilizatorii, care pot răspunde la întrebări și obține scoruri corecte, bazate pe performanțele lor.

## 1.2 Funcționalități principale:

* **Crearea și gestionarea quizurilor**: Administratorii pot crea quizuri cu multiple întrebări și opțiuni de răspuns.
* **Participarea la quizuri**: Utilizatorii pot răspunde la întrebări, iar rezultatele lor sunt evaluate automat.
* **Autentificare și autorizare**: Utilizatorii trebuie să se autentifice pentru a putea accesa quizurile, iar administratorii beneficiază de permisiuni suplimentare.
* **Evaluarea răspunsurilor**: După completarea quiz-ului, utilizatorii primesc un punctaj bazat pe corectitudinea răspunsurilor lor.

## 1.3Tehnologii utilizate

Acest proiect a fost construit folosind următoarele tehnologii:

* **Frontend**: React.js pentru crearea interfeței utilizatorului. Acesta utilizează funcționalități moderne precum hooks și routing pentru a crea o experiență dinamică și interactivă.
* **Backend**: Spring Boot pentru a gestiona logica server-side și pentru a furniza un API RESTful ce poate comunica cu frontend-ul. De asemenea, se utilizează Spring Security pentru gestionarea autentificării și securității aplicației.
* **Baza de date**: MySQL pentru stocarea informațiilor despre utilizatori, quizuri și întrebări.
* **Autentificare**: JWT (JSON Web Token) pentru a asigura autentificarea și autorizarea utilizatorilor pe platformă.
* **Testare**: Unități de testare au fost implementate atât pentru backend, cât și pentru frontend pentru a asigura integritatea aplicației.

## 1.4 Obiectivul proiectului

Obiectivul principal al acestui proiect este de a crea o aplicație scalabilă, ușor de utilizat, care să ofere o experiență completă de participare și administrare a quizurilor. Aplicația va permite administratorilor să gestioneze quizurile și întrebările, iar utilizatorilor să participe la teste și să își evalueze performanțele.

# 2. Arhitectura Sistemului

## 2.1 Prezentare Generală

Aplicația de tip quiz online utilizează o arhitectură client-server, cu un frontend interactiv și un backend care gestionează logica aplicației, stocarea datelor și autentificarea utilizatorilor. Sistemul este structurat astfel încât să permită o interacțiune eficientă între utilizatori și administratori, respectând principiile separării responsabilităților.

## 2.2 Componentele Principale

Arhitectura sistemului este împărțită în două părți majore: frontend și backend, fiecare având un rol clar definit.

### 2.2.1 Frontend

Frontend-ul aplicației este dezvoltat folosind **React.js**, o bibliotecă JavaScript populară pentru construirea interfețelor de utilizator. Acesta se ocupă de prezentarea datelor și interacțiunea utilizatorilor cu aplicația. Principalele componente includ:

* **Componenta de autentificare**: Permite utilizatorilor să se autentifice folosind un nume de utilizator și o parolă. După autentificare, utilizatorii pot accesa quizurile disponibile.
* **Pagina de listare a quizurilor**: Afișează o listă de quizuri disponibile, cu opțiunea de a le vizualiza și completa.
* **Pagina de participare la quiz**: Permite utilizatorilor să răspundă la întrebări, să selecteze răspunsuri și să trimită testul pentru evaluare.
* **Pagina de rezultate**: După completarea unui quiz, utilizatorii pot vizualiza rezultatele obținute, inclusiv scorul și răspunsurile corecte.

### 2.2.2 Backend

Backend-ul este construit folosind **Spring Boot**, un framework Java care permite dezvoltarea rapidă a aplicațiilor. Acesta gestionează logica aplicației, comunicarea cu baza de date, autentificarea utilizatorilor și evaluarea quizurilor. Componentele backend-ului includ:

* **Controlerul de Quiz**: Permite crearea, gestionarea și obținerea quizurilor. Acesta furnizează endpointuri pentru a adăuga, actualiza și lista quizurile.
* **Controlerul de Întrebări**: Gestionază întrebările fiecărui quiz, inclusiv adăugarea și actualizarea acestora.
* **Controlerul de Autentificare**: Permite autentificarea utilizatorilor, generarea de token-uri JWT pentru sesiuni sigure și autorizarea utilizatorilor pentru acces la resurse protejate.
* **Controlerul de Evaluare a Quizurilor**: Primește răspunsurile utilizatorilor și calculează scorul în funcție de răspunsurile corecte.
* **Servicii de Utilizator și Quiz**: Acestea sunt responsable cu logica de business, cum ar fi validarea răspunsurilor și gestionarea datelor utilizatorilor și quizurilor.

### 2.2.3 Baza de Date

Baza de date folosită în acest sistem este **MySQL**, care stochează informațiile despre utilizatori, quizuri, întrebări și răspunsuri. Structura bazei de date include următoarele entități principale:

* **Utilizatori**: Informații despre utilizatori, inclusiv numele, adresa de e-mail și parola criptată.
* **Quizuri**: Detalii despre quizuri, inclusiv titlul, descrierea, categoria și numărul de întrebări.
* **Întrebări**: Întrebările fiecărui quiz, inclusiv opțiunile de răspuns și răspunsul corect.
* **Rezultate**: Informații despre scorurile utilizatorilor, inclusiv numărul de întrebări corecte și punctajul obținut.

A screenshot of a computer screen

Description automatically generated

## 2.3 Fluxul Aplicației

1. **Autentificarea utilizatorului**: Utilizatorii se autentifică în aplicație, iar backend-ul returnează un token JWT, care este stocat în browserul utilizatorului pentru autentificare ulterioară.
2. **Vizualizarea quizurilor**: Utilizatorii pot accesa lista de quizuri disponibile, care este prezentată pe frontend. Administratorii pot adăuga sau modifica quizuri.
3. **Participarea la quiz**: Utilizatorii pot selecta un quiz, răspunde la întrebări și trimite răspunsurile pentru evaluare.
4. **Evaluarea quizului**: Backend-ul compară răspunsurile date de utilizatori cu răspunsurile corecte din baza de date și calculează scorul utilizatorului.
5. **Vizualizarea rezultatelor**: După completarea quizului, utilizatorul poate vizualiza rezultatele și poate obține feedback asupra răspunsurilor sale.

## 2.4 Securitate

Pentru a asigura un nivel înalt de securitate, aplicația folosește **JWT (JSON Web Token)** pentru autentificare și autorizare. Acest sistem asigură că doar utilizatorii autentificați pot accesa resursele protejate, iar sesiunile utilizatorilor sunt gestionate pe bază de tokenuri.

De asemenea, backend-ul implementează **Spring Security**, care protejează aplicația împotriva atacurilor comune, cum ar fi atacurile XSS și CSRF.

Arhitectura aplicației este concepută pentru a fi scalabilă și ușor extensibilă. Se pot adăuga noi funcționalități, cum ar fi generarea de statistici detaliate pentru utilizatori sau implementarea unor quizuri mai complexe. De asemenea, aplicația poate fi extinsă pentru a suporta mai mulți utilizatori simultan, fără a compromite performanța.

# Capitolul 3: Designul Aplicației

Designul aplicației reprezintă structura și funcționalitatea esențială a aplicației de quiz, inclusiv interfața utilizatorului (UI) și logica de backend. În acest capitol, vom discuta despre principalele componente ale aplicației, cum funcționează fluxul de utilizare și cum sunt implementate diversele funcționalități ale aplicației.

## 3.1 Componentele Aplicației

Aplicația este formată din două mari componente: **Frontend** și **Backend**, fiecare având un rol distinct în gestionarea fluxului de informații și în interacțiunea cu utilizatorul.

### 3.1.1 Frontend

Frontend-ul aplicației este implementat în **React**, care asigură o interfață dinamică și intuitivă pentru utilizatori. Componentele principale includ:

* **Pagina de autentificare**: Permite utilizatorilor să se logheze sau să se înregistreze în aplicație.
* **Pagina principală (Dashboard)**: Afișează lista de quizuri disponibile și permite utilizatorilor să aleagă quizurile pe care vor să le încerce.
* **Pagina de quiz**: Permite utilizatorilor să participe la quizuri, răspunzând întrebărilor.
* **Pagina de rezultate**: Afișează scorul și răspunsurile corecte la finalul quizului.

### 3.1.2 Backend

Backend-ul aplicației este implementat folosind **Spring Boot**, care furnizează API-uri REST pentru gestionarea quizurilor, întrebărilor și utilizatorilor. Componentele principale includ:

* **Autentificare și gestionarea sesiunilor**: Se utilizează JWT pentru autentificarea utilizatorilor și protejarea resurselor.
* **Gestionarea quizurilor**: Permite crearea, actualizarea și ștergerea quizurilor.
* **Gestionarea întrebărilor**: Permite adăugarea, actualizarea și ștergerea întrebărilor asociate quizurilor.
* **Evaluarea quizurilor**: Permite calcularea scorului pe baza răspunsurilor oferite de utilizator.

## 3.2 Fluxul Aplicației

Aplicația urmează un flux bine definit, care permite utilizatorilor să interacționeze ușor cu quizurile și să obțină rapid rezultatele. Iată pașii principali ai fluxului aplicației:

1. **Autentificarea utilizatorului**:
   * Utilizatorul accesează pagina de autentificare, introduce credențialele (username și parolă), iar aplicația verifică autentificarea.
   * După autentificare, utilizatorul primește un token JWT care este utilizat pentru a accesa resursele protejate ale aplicației.
2. **Selectarea unui quiz**:
   * După autentificare, utilizatorul este redirecționat către pagina principală unde poate vizualiza o listă de quizuri.
   * Utilizatorul selectează un quiz și este direcționat către pagina respectivă, unde poate începe să răspundă la întrebări.
3. **Participarea la quiz**:
   * Utilizatorul răspunde la întrebările quizului selectat.
   * Răspunsurile sunt stocate temporar în aplicație și sunt trimise la server pentru evaluare după ce utilizatorul trimite quizul.
4. **Evaluarea și afișarea rezultatelor**:
   * După ce utilizatorul trimite quizul, backend-ul evaluează răspunsurile și calculează scorul.
   * Utilizatorul primește un raport cu răspunsurile corecte, scorul obținut și întrebările nefinalizate.

## 3.3 Interfața Utilizatorului (UI)

Interfața utilizatorului este simplă, dar eficientă, fiind realizată pentru a fi ușor de folosit chiar și de utilizatorii care nu au experiență tehnică. Designul se axează pe o experiență de utilizare intuitivă, cu următoarele caracteristici:

* **Pagini responsabile**: Aplicația este complet responsivă și se adaptează pentru a funcționa pe dispozitive mobile și desktop.
* **Formulare clare**: Câmpurile de input pentru răspunsuri sunt ușor de accesat, iar utilizatorii pot selecta opțiunile dorite folosind radio buttons.
* **Feedback imediat**: Utilizatorii primesc un feedback instantaneu atunci când termină un quiz, iar rezultatele sunt afișate într-un mod clar și concis.

## 3.4 Designul Bazei de Date

Baza de date este construită astfel încât să sprijine gestionarea eficientă a quizurilor, întrebărilor și răspunsurilor. Iată structura principală a bazei de date:

* **Tabelele principale**:
  + **Utilizatori**: Stochează informațiile despre utilizatori (username, parolă, rol, etc.).
  + **Quizuri**: Stochează detalii despre quizuri (titlu, descriere, număr de întrebări, etc.).
  + **Întrebări**: Stochează întrebările fiecărui quiz, împreună cu opțiunile de răspuns și răspunsul corect.
  + **Răspunsuri**: Stochează răspunsurile utilizatorilor la întrebările quizului.
  + **Rezultate**: Stochează rezultatele fiecărui quiz completat de utilizatori (scor, întrebări corecte, etc.).

## 3.5 Funcționalitățile Aplicației

### 3.5.1 Gestionarea Quizurilor

* Crearea și editarea quizurilor.
* Vizualizarea tuturor quizurilor disponibile.

### 3.5.2 Gestionarea Întrebărilor

* Adăugarea, actualizarea și ștergerea întrebărilor.
* Asocierea întrebărilor cu quizuri.

### 3.5.3 Evaluarea și Vizualizarea Rezultatelor

* Evaluarea răspunsurilor utilizatorului și calcularea punctajului.
* Vizualizarea rezultatelor complete, inclusiv răspunsurile corecte și scorul final.

# Capitolul 4: Implementarea Aplicației

În acest capitol, vom discuta despre procesul de implementare al aplicației, detaliind tehnologiile utilizate, structura codului și pașii necesari pentru a construi aplicația. Vom aborda atât frontend-ul, cât și backend-ul aplicației, explicând cum interacționează între ele și cum sunt implementate funcționalitățile esențiale.

## 4.1 Tehnologii Folosite

Aplicația este construită folosind o arhitectură pe două nivele: **frontend** și **backend**, fiecare cu tehnologii specifice care permit o dezvoltare rapidă și scalabilă.

### 4.1.1 Frontend

Frontend-ul aplicației este construit cu **React**, o bibliotecă JavaScript populară pentru construirea interfețelor de utilizator. React permite dezvoltarea de aplicații web dinamice și interactive, iar utilizatorii pot interacționa cu aplicația într-un mod fluid și rapid.

Tehnologiile folosite în frontend sunt:

* **React**: Biblioteca principală pentru construirea interfeței utilizatorului.
* **React Router**: Utilizat pentru a gestiona navigarea între pagini.
* **Axios**: Folosit pentru a efectua cereri HTTP către backend-ul aplicației.
* **CSS**: Folosit pentru stilizarea interfeței și crearea unui design responsiv.
* **JWT (JSON Web Tokens)**: Utilizat pentru autentificarea utilizatorilor și protejarea rutei de acces.

### 4.1.2 Backend

Backend-ul aplicației este construit cu **Spring Boot**, un framework Java care permite dezvoltarea rapidă de aplicații web. Spring Boot facilitează crearea de aplicații de tip RESTful cu un minim de configurare.

Tehnologiile utilizate în backend includ:

* **Spring Boot**: Framework-ul principal pentru crearea aplicației de backend.
* **Spring Security**: Folosit pentru autentificarea și autorizarea utilizatorilor, inclusiv integrarea cu JWT.
* **Spring Data JPA**: Folosit pentru accesul la baza de date și gestionarea entităților.
* **MySQL**: Baza de date relațională folosită pentru stocarea datelor aplicației, precum quizuri, întrebări și răspunsuri.
* **JWT (JSON Web Tokens)**: Folosit pentru autentificarea utilizatorilor și protejarea rutei de acces.

## 4.2 Structura Aplicației

Aplicația este organizată în două părți principale: frontend și backend. Fiecare parte are o structură specifică care facilitează dezvoltarea și întreținerea aplicației.

### 4.2.1 Frontend

Frontend-ul conține fișierele și componentele care sunt responsabile de interacțiunea cu utilizatorul. Structura principală a folderelor este următoarea:

* **public/**: Conține fișierele publice (de exemplu, index.html).
* **src/**: Conține fișierele sursă pentru aplicație:
  + **components/**: Componentele React care formează interfața utilizatorului (de exemplu, paginile pentru autentificare, dashboard, quizuri).
  + **services/**: Contine serviciile pentru interacțiunea cu backend-ul (de exemplu, gestionarea cererilor API pentru quizuri, întrebări, autentificare).
  + **utils/**: Funcții și utilitare pentru diferite operațiuni (de exemplu, decodificarea token-ului JWT).
  + **styles/**: Fișiere CSS pentru stilizarea aplicației.
  + **App.js**: Fișierul principal care coordonează toate componentele și rutele aplicației.

### 4.2.2 Backend

Backend-ul conține fișierele și clasele care gestionează logica aplicației și comunicarea cu baza de date. Structura principală a folderelor este următoarea:

* **src/main/java/com/exam/**: Conține toate fișierele Java pentru aplicație:
  + **controller/**: Contine clasele care gestionează rutele HTTP (de exemplu, QuestionController, QuizController).
  + **model/**: Conține clasele care reprezintă entitățile aplicației (de exemplu, Quiz, Question, User).
  + **repository/**: Conține interfețele pentru interacțiunea cu baza de date.
  + **service/**: Conține logica de afaceri a aplicației (de exemplu, QuizService, QuestionService).
  + **config/**: Conține configurațiile pentru autentificare și securitate (de exemplu, configurarea JWT și securitatea aplicației).
  + **resources/**: Conține fișierele de configurare (de exemplu, application.properties).

## 4.3 Detalii Implementare

### 4.3.1 Frontend - Componente și Funcționalități

Frontend-ul este construit în jurul unor componente React care gestionează diversele funcționalități ale aplicației, inclusiv:

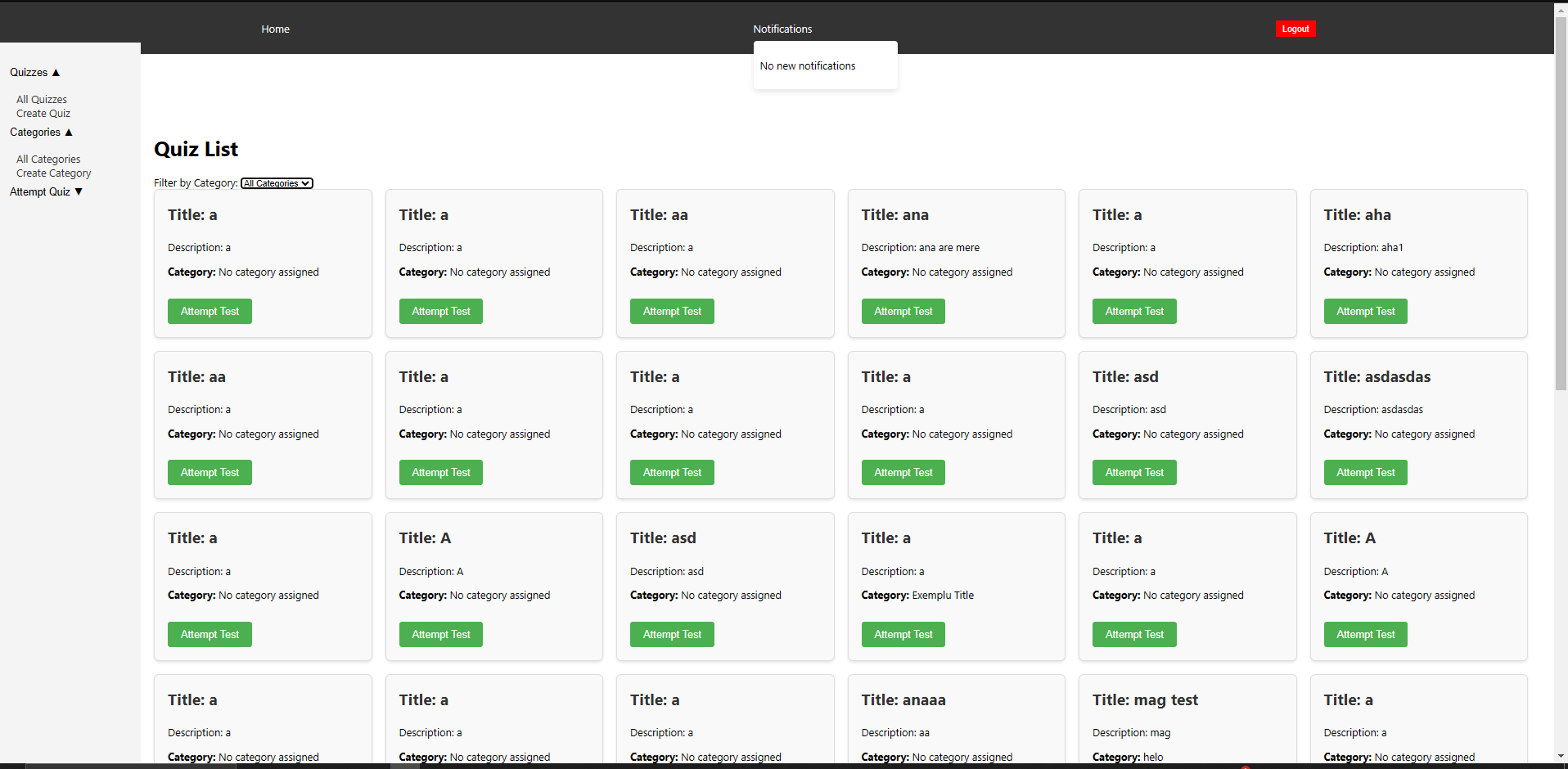
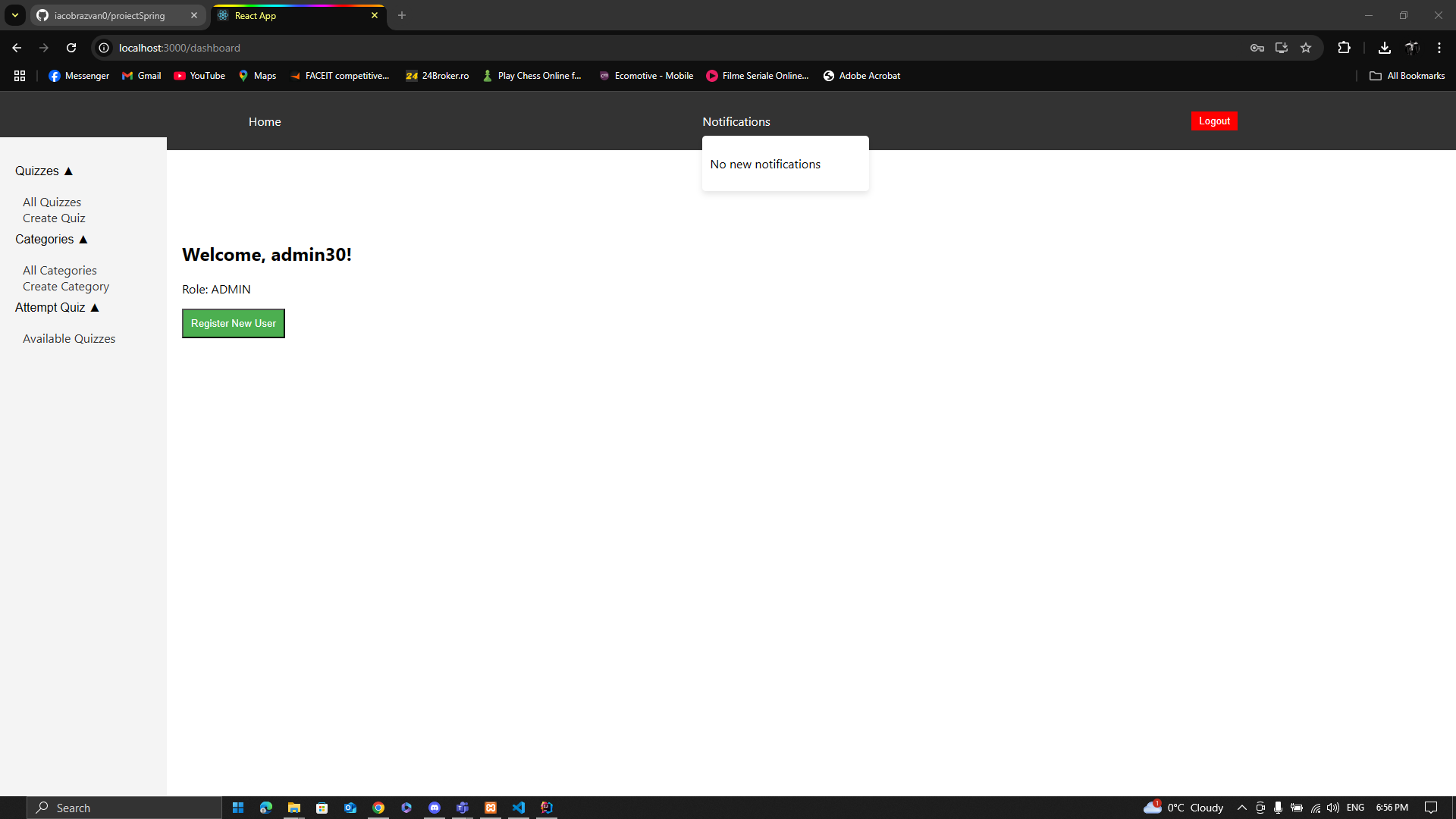
* **Componentele de autentificare**: Aceste componente permit utilizatorilor să se înregistreze și să se autentifice. După autentificare, aplicația salvează un token JWT în localStorage, care va fi folosit pentru autentificarea ulterioară.
* **Componentele de quizuri**: Acestea permit utilizatorilor să vizualizeze lista de quizuri disponibile, să aleagă un quiz și să răspundă la întrebările din acesta.
* **Componentele de rezultate**: După ce utilizatorul trimite un quiz, rezultatele sunt calculate și afișate, incluzând răspunsurile corecte și scorul obținut.

### 4.3.2 Backend - Endpoints și Funcționalități

Backend-ul oferă un set de endpointuri RESTful care sunt utilizate pentru gestionarea quizurilor și întrebărilor. Cele mai importante endpointuri sunt:

* **/quiz/getAllQuiz**: Returnează toate quizurile disponibile.
* **/quiz/{quizId}**: Returnează detalii despre un quiz specificat prin ID.
* **/question/{quizId}**: Returnează întrebările pentru un quiz specificat.
* **/question/evaluate-quiz**: Primește răspunsurile utilizatorilor și calculează scorul obținut.
* **/user/**: Endpointuri pentru gestionarea utilizatorilor, cum ar fi înregistrarea, autentificarea și recuperarea informațiilor despre utilizatori.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated