

<InfoConquer/>

ADELIN SOFIAN

Clasa a X-a

**Colegiul Național “Petru Rareș” Suceava
prof. Îndrumători: MARIUS UDUDEC
MIHAIL ERHAN**

Cuprins

01

Introducere

Prezentarea proiectului (3)
Scopul și obiectivul proiectului (4)
Motivația și importanța proiectului (4)

02

Proiectare și dezvoltare

Descrierea conceptului proiectului (5 - 6)
Detalii despre implementare și dezvoltare
(Tehnologii, FE, BE, DB) (7 - 10)

03

Funcționalități

Descrierea funcționalităților oferite de proiect (10 - 15)
Exemple de cod pentru funcționalități cheie și
explicarea lor (16 - 20)

04

Evaluare și performanță

Evaluarea performanței proiectului în raport cu
obiectivele stabilite (21)
Analiza performanței (22)

05

Securitate

Descrierea măsurilor de securitate implementate
(23 - 25)

06

Concluzii

Sumar al rezultatelor și concluziilor obținute (26)
Perspectivele și direcțiile viitoare pentru proiect,
inclusiv dezvoltarea ulterioară a codului. (27)

Introducere

Prezentarea proiectului

- InfoConquer este o platformă educațională online care oferă o varietate de resurse și activități pentru cei pasionați de programare. Grupul țintă al proiectului este format din persoane ce doresc să-și îmbunătățească abilitățile de programare sau să învețe programare de la zero. Aceste persoane pot fi începători, intermediari sau chiar avansați în programare, mereu în căutare de noi cunoștințe și de îmbunătățire a abilităților existente.
- Principala caracteristică a proiectului este code judge, un instrument ce oferă feedback bazat pe cazuri de teste, esențial pentru îmbunătățirea abilităților de programare și înțelegerea conceptelor. În plus, proiectul utilizează tehnologia de la OpenAI pentru a furniza feedback direct pe cod, ajutând astfel utilizatorul să înțeleagă mai bine greșelile și să decidă ce trebuie să îmbunătățească sau să modifice.

Introducere

Scopul și obiectivul proiectului
Motivația și importanța proiectului

- Obiectivul principal al InfoConquer este să creeze o platformă educațională completă și accesibilă care să faciliteze învățarea programării pentru toate nivelurile de competență. Îmi propun să ofer utilizatorilor instrumentele și resursele necesare pentru a-și dezvolta abilitățile într-un mod structurat și eficient.
- Scopul meu este de a sprijini și îmbunătăți procesul de învățare al programării prin intermediul unei platforme intuitive care permite utilizatorilor să rezolve probleme de **algoritmică** în multiple limbaje de programare.
- Motivația din spatele InfoConquer vine din dorința de a oferi un instrument educațional care să răspundă nevoilor variate ale celor care învață programare. Observând dificultățile întâmpinate de mulți începători, precum și nevoia constantă de actualizare a cunoștințelor pentru cei avansați, am decis să dezvolt o platformă care să ofere suport adecvat pentru toate nivelurile.

Proiectare și dezvoltare

Descrierea conceptului proiectului

- InfoConquer este o platformă educațională online dedicată algoritmicii, axată pe furnizarea unei experiențe de învățare interactivă și eficientă pentru utilizatori de toate nivelele de competență în programare. Principala sa caracteristică este un instrument numit code judge, care evaluează codul scris de utilizatori prin rularea acestuia împotriva unor seturi de cazuri de test predefinite și oferă feedback imediat și detaliat.
- Din motive de securitate, fiecare cod este rulat într-un mediu izolat, creat cu ajutorul tehnologiei Docker.
- Platforma include și un sistem pentru concursuri de algoritmică , unde utilizatorii pot participa la competiții și provocări pentru a-și testa abilitățile și a se compara cu alți programatori. Acest aspect adaugă o dimensiune interactivă și competitivă la experiența de învățare a algoritmicii pe platforma InfoConquer.

Proiectare și dezvoltare

Descrierea conceptului proiectului

- În plus, InfoConquer oferă resurse educaționale sub formă de articole care acoperă diverse subiecte din domeniul programării, de la concepte de bază până la tehnici avansate. Aceste articole sunt concepute pentru a ajuta utilizatorii să-și largescă cunoștințele și să înțeleagă mai bine diferite aspecte ale programării.
- Astfel, conceptul proiectului InfoConquer se concentrează pe oferirea unei suite de instrumente și resurse care să permită utilizatorilor să-și îmbunătățească abilitățile de programare prin practică, feedback și participare la concursuri.

Proiectare și dezvoltare

Detalii despre implementare și dezvoltare

1. Tehnologii utilizate

- **Frontend:**
 - ReactJS
 - JavaScript 
 - CSS
 - HTML
 - Tailwind CSS 
 - NextUI 
- **Backend:**
 - NodeJS 
 - ExpressJS 
 - MongoDB 
 - GraphQL 
 - Apollo Server 
 - Third-Party services:
 - OpenAI 
- **DevOps and Security Tool:**
 - Git 
 - Docker 
 - Cloudflare's smart CAPTCHA 
 - Digital Ocean 

2. Frontend

În cadrul acestui proiect, am optat pentru integrarea librăriei ReactJS datorită capacitatea sale eficiente și flexibile de a dezvolta interfețe de utilizator. ReactJS facilitează crearea de componente reutilizabile, sporind astfel ușurința în menținerea și extinderea codului.

De asemenea, beneficiind de un DOM virtual, ReactJS optimizează performanța aplicațiilor prin minimizarea modificărilor efective ale DOM-ului. Alături de ReactJS, pentru elaborarea interfețelor de utilizator, mă bazez pe NextUI și TailwindCSS, două biblioteci puternice și flexibile ce simplifică și îmbunătățesc procesul de stilizare al aplicațiilor web.

Proiectare și dezvoltare

Detalii despre implementare și dezvoltare

NextUI oferă o diversitate de componente predefinite, ce pot fi personalizate și integrate cu ușurință în proiecte. Aceste componente sunt concepute pentru a fi accesibile și compatibile cu toate browserele moderne, oferind astfel o experiență consistentă utilizatorilor pe diverse dispozitive.

În contrast, TailwindCSS este un framework CSS utilitar ce facilitează crearea rapidă a design-urilor complexe prin utilizarea unui set de clase predefinite.

3. Backend

În cadrul backend-ului acestui proiect, am decis să utilizez NodeJS împreună cu ExpressJS datorită eficienței și flexibilității mediului runtime pe care îl oferă pentru dezvoltarea aplicațiilor web. NodeJS permite executarea codului JavaScript pe server, în timp ce ExpressJS reprezintă un cadru minimalist și rapid, ideal pentru crearea de API-uri robuste și scalabile.

Pentru gestionarea bazei de date, am ales MongoDB, o soluție NoSQL ce se distinge prin capacitatea sa de a gestiona datele nestructurate și semi-structurate. MongoDB este cunoscută pentru scalabilitatea și performanța sa, fiind o alegere potrivită pentru aplicațiile ce necesită diverse cerințe de stocare a datelor.

Proiectare și dezvoltare

Detalii despre implementare și dezvoltare

4. Baze de date și stocarea de date

Pentru acest proiect, am decis să utilizez o bază de date de tip NoSQL, MongoDB, cunoscută pentru structura sa orientată pe documente. Principala motivatie pentru alegerea acestei solutii este flexibilitatea oferită. Prin capacitatea sa de a stoca date sub forma de documente în format similar JSON, MongoDB oferă o adaptabilitate remarcabilă în manipularea datelor. De asemenea, MongoDB furnizează un cadru robust de agregare, facilitând procesarea și transformarea datelor printr-un sir de operațiuni, inclusiv filtrare, sortare, grupare și transformare.

MongoDB Atlas reprezintă un serviciu cloud integral gestionat, simplificând implementarea, securitatea și monitorizarea bazelor de date MongoDB.

Utilizez MongoDB pentru gestionarea datelor referitoare la utilizatori, probleme, articole, concursuri și anunțuri în cadrul proiectului meu. Pentru a asigura securitatea și confidențialitatea datelor, parolele sunt stocate sub formă de hash, nu în text simplu.

Schemele asociate pentru organizarea și accesarea datelor sunt detaliate în documentele atașate.

- User
- Article
- Problem
- Contest
- Announcement

Functionalități

Descrierea funcționalităților oferite de proiect

1. Autentificare și înregistrare

În acest proiect, am convenit să restricționez accesul la platformă doar pentru utilizatorii neautentificați, permitându-le să vizualizeze problemele/articolele, dar interacțiunile cu acestea să le fie limitate.

Utilizatorii pot crea un cont folosind un formular de înregistrare ce solicită informațiile necesare (adresă de email, nume de utilizator, parolă, confirmare a parolei), fiind esențială validarea lor ca fiind non-robotică.

După trimiterea formularului către server, acesta va fi verificat, iar utilizatorului îi va fi expediat un email de confirmare.

Crearea de conturi poate fi blocată din diverse motive, inclusiv:

- Există deja un cont asociat cu adresa de email/numele de utilizator furnizate.
- Parola nu respectă standardele de securitate.
- Utilizatorul nu a bifat căsuța de confirmare a identității non-robotice.
- Utilizatorul nu a validat adresa de email.

Functionalități

Descrierea funcționalităților oferite de proiect

2. Creare de conținut

Pentru a asigura o experiență completă și eficientă utilizatorilor, am dezvoltat pagini dedicate creării conținutului, inclusiv probleme, articole și anunțuri.

Procesul de creare a problemelor, articolelor și anunțurilor este facilitat de paginile noastre dedicate: [/problems/publish](#), [/articles/publish](#) și [/announcements/publish](#), respectiv.

Acstea pagini sunt concepute cu un formular exhaustiv pentru introducerea tuturor informațiilor necesare pentru fiecare tip de conținut.

Un aspect important de menționat este că doar utilizatorii cu rolul de 'admin' au permisiunea de a posta probleme sau anunțuri, însă oricine poate publica anunțuri.

Pentru a garanta securitatea conținutului furnizat de utilizatori și pentru a preveni posibile vulnerabilități de tip XSS (cross-site scripting), am implementat soluția "DOM Purify". Această soluție se ocupă de curățarea și filtrarea conținutului HTML transmis pentru a elibera orice potențiale atacuri de tip cross-site scripting sau alte vulnerabilități.

Articolele pot fi evaluate pozitiv sau negativ, iar numărul acestora este monitorizat. Articolele pot fi editate direct pe site, însă problemele și anunțurile necesită modificări direct în baza de date.



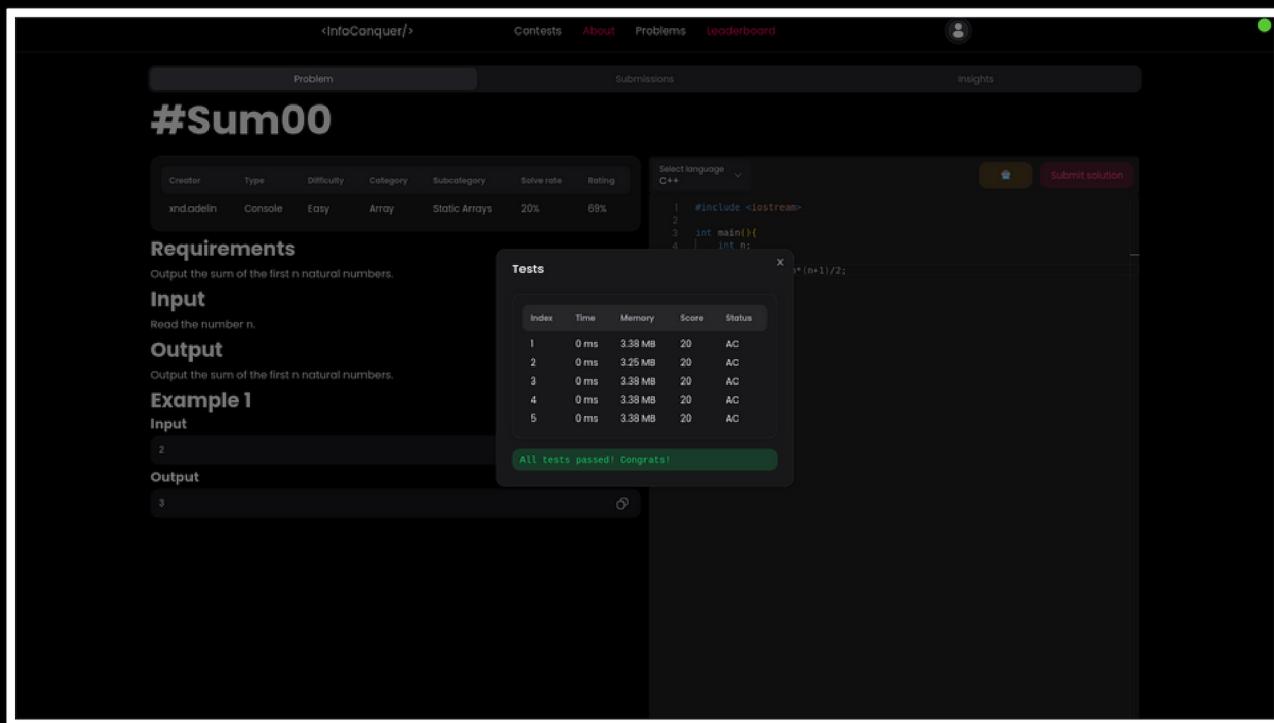
Functionalități

Descrierea funcționalităților oferite de proiect

3. Rezolvarea problemelor și AI Help

Scopul principal al platformei noastre constă în simplificarea procesului de rezolvare a problemelor de algoritmică pentru utilizatori, oferindu-le acces la o diversitate de probleme încadrate în diferite categorii și subcategorii.

Pe pagina dedicată unei anumite probleme, utilizatorii se întâlnesc cu un panou ce conține descrierea problemei și un editor de cod (Monaco Editor). Aici utilizatorii pot redacta și testa soluțiile propuse pentru problema respectivă. De asemenea, platforma dispune de un AI Helper ce utilizează tehnologia OpenAI pentru a furniza asistență utilizatorilor.



În plus, oferă utilizatorilor posibilitatea de a vizualiza submisii trimise și perspective asupra problemei, inclusiv numărul de rezolvări ale unei probleme în ultimele 7 zile, statisticile zilnice, primele 3 submisii acceptate și cele mai eficiente soluții în ceea ce privește gestionarea memoriei și timpul.

Functionalități

Descrierea funcționalităților oferite de proiect

4. Concursuri de algoritmică

Platforma oferă utilizatorilor posibilitatea de a participa la concursuri de algoritmică, oferindu-le o modalitate excelentă de a-și testa abilitățile și de a concura cu alți programatori pasionați.

Fiecare concurs include următoarele detalii:

- Data de începere și data de încheiere
- Limbajele acceptate și problemele disponibile
- Descriere
- Clasamentul participanților

The screenshot shows a dark-themed web application interface. At the top, there is a navigation bar with links for 'Contests', 'About', 'Problems', and 'Leaderboard'. Below the navigation, there is a section titled 'Contests' with a trophy icon. Two contest cards are displayed side-by-side. Each card has a title, a 'Join contest' button, and a 'Join' button at the bottom right. The first contest is titled 'InfoEducatie' and the second is 'Lorem Ipsum'. Both cards show placeholder text for descriptions and languages.

Clasamentul concursului poate fi vizualizat în timp real, fiind ordonat în funcție de punctajul maxim obținut pentru fiecare problemă.

The screenshot shows a detailed view of the 'InfoEducatie' contest. It includes a 'Leaderboard' table and a 'Problems' table. The Leaderboard table lists participants with their names, scores, and submission times. The Problems table lists the problems with their names, difficulty levels, and submission counts. A large red number '13' is overlaid in the bottom right corner of the screenshot area.

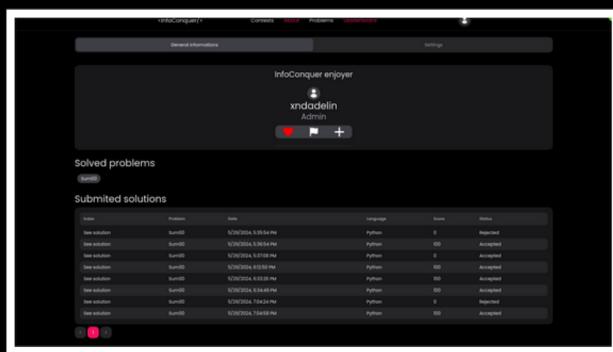
Functionalități

Descrierea funcționalităților oferite de proiect

5. Vizualizarea profilului și setari

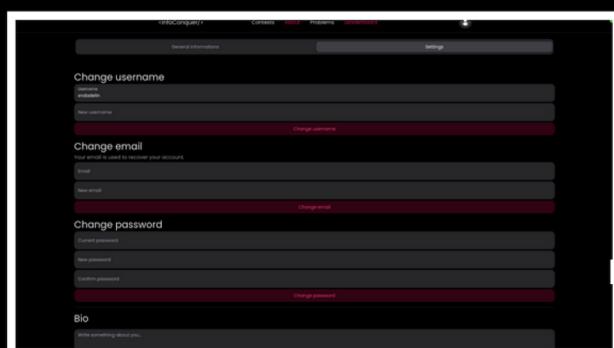
Pe pagina de profil, utilizatorii beneficiază de acces la informațiile personale și la un rezumat al activității lor pe platformă. Aceste informații includ:

- Un tab ce conține următoarele detalii:
 - Numele utilizatorului
 - Rolul acestuia
 - Problemele rezolvate
 - O listă cu problemele recent soluționate de către utilizator (inclusiv scorul, limbajul de programare, data etc.) și un link direct către soluțiile oferite.
- Un alt tab dedicat actualizării informațiilor personale, precum adresa de email, parola și numele de utilizator. De asemenea, utilizatorul poate modifica biografia asociată profilului său.



Informații generale

Setări



Functionalități

Descrierea funcționalităților oferite de proiect

6. Vizualizarea soluțiilor

Fiecare utilizator poate vizualiza soluția trimisă către server, care cuprinde următoarele elemente:

- Codul sursă
- Detalii despre problema abordată
- Informații referitoare la timpul de execuție și memoria utilizată de soluție pentru fiecare test, alături de scorul obținut și detaliile testului utilizat (output, input, output așteptat)
- Starea soluției pentru fiecare test AC/WA/TLE/MLE. În cazul unei erori de compilare sau de rulare, aceasta va fi afișată pe pagina soluției.

The screenshot shows a web interface for a programming competition. At the top, there are navigation links: <InfoConquer/>, Contests, About, Problems, and leaderboard. On the right side, there is a user profile icon.

The main content area has a dark background with white text. It displays the following information:

Solution for problem: Sum00

Problem	Sum00
Language	Python
Status	Accepted
Username	xndodelin
Date	5/29/2024, 5:38:54 PM
File memory	38 bytes
Solution accepted	Yes

Submitted code

```
n = int(input())
print(int(n*(n+1)/2))
```

Tests summary

Test	Execution time	Memory used	Score	Status	Test case
1	10 ms	8.5 MB	20	AC	See test case
2	10 ms	8.63 MB	20	AC	See test case
3	10 ms	8.63 MB	20	AC	See test case
4	10 ms	8.5 MB	20	AC	See test case
5	10 ms	8.5 MB	20	AC	See test case

Functionalități

Exemple de cod pentru funcționalități cheie și explicarea lor

1. Verificarea utilizatorului

```
1 const getUser = async(context) => {
2     try{
3         const cookies = cookie.parse(context.req.headers.cookie) || ''
4         if(!cookies) return null;
5         const token = cookies.token;
6         const refreshtoken = cookies.refreshToken
7         if(!token && !refreshtoken){
8             return null;
9         }
10        const verified = jwt.verify(token, process.env.SECRET);
11        if(verified.username){
12            const username = verified.username
13            const user = await User.findOne({username, verified: true})
14            return user;
15        }else{
16            const refreshVerified = jwt.verify(refreshtoken, process.env.SECRET_REFRESH);
17            if(refreshVerified){
18                const newToken = generateToken(refreshVerified)
19                const newRefreshToken = generateRefreshToken(refreshVerified)
20                context.res.cookie('token', newToken, {
21                    httpOnly: true,
22                    secure: true
23                })
24                context.res.cookie('refreshToken', newRefreshToken, {
25                    httpOnly: true,
26                    secure: true
27                })
28                const refreshedUser = await User.findOne({username: refreshVerified.username, verified: true})
29                return refreshedUser
30            }
31        }
32        return null;
33    }catch(err){
34        if (err.name === 'TokenExpiredError') {
35            const refreshtoken = cookie.parse(context.req.headers.cookie).refreshToken
36            const refreshVerified = jwt.verify(refreshtoken, process.env.SECRET_REFRESH);
37            if (refreshVerified) {
38                try{
39                    const newToken = generateToken(refreshVerified);
40                    const newRefreshToken = generateRefreshToken(refreshVerified);
41
42                    context.res.cookie('token', newToken, {
43                        httpOnly: true,
44                        sameSite: 'strict',
45                        secure: true,
46                    });
47
48                    context.res.cookie('refreshToken', newRefreshToken, {
49                        httpOnly: true,
50                        sameSite: 'strict',
51                        secure: true,
52                    });
53                    const refreshedUser = await User.findOne({username: refreshVerified.username, verified: true})
54                    return refreshedUser
55                }catch(err){
56                    return null;
57                }
58            }
59        }
60    }
61 }
62 }
```

Funcționalități

Exemple de cod pentru funcționalități cheie și explicarea lor

Această funcție reprezintă procesul de identificare a unui utilizator și reîmprospătarea cookie-urilor JWT (JSON Web Token) în momentul expirării token-ului principal. Aceasta constituie elementul fundamental al fiecărei interogări sau mutații, în care sunt validate credențialele utilizatorului care inițiază o solicitare pentru a minimiza riscurile de atacuri cibernetice.

Principalele funcționalități ale acestei funcții includ:

- Parsarea cookie-urilor:
 - Acestea sunt analizate pentru a extrage token-urile.
 - Acestea sunt configurate cu 3 flag-uri:
 - httpOnly: true, împiedică accesul JavaScript-ului clientului la cookie-uri, reducând riscul de atac XSS.
 - secure: true, garantează că cookie-urile sunt transmise doar prin conexiuni HTTPS, diminuând posibilitatea unui atac de tip MITM.
 - sameSite: 'Strict', previne expedierea cookie-urilor în cereri cross-site, minimizând riscul de CSRF.
- Verificarea token-ului:
 - Token-ul este autentificat utilizând secretul din mediu de execuție pentru a valida identitatea utilizatorului.
- Reîmprospătarea token-ului:
 - În cazul expirării sau invalidității token-ului, funcția verifică refresh token-ul utilizând un alt secret din mediu de execuție și generează noi token-uri.

Functionalități

Exemple de cod pentru funcționalități cheie și explicarea lor

- Setarea Cookie-urilor:
 - Noile token-uri sunt stocate în cookie-uri cu diverse flag-uri pentru a menține sesiunea activă și securizată.
- Gestionarea adecvată a erorilor.
- Utilizatorii rămân autentificați chiar și după expirarea token-ului inițial, datorită mecanismului de refresh token.
- Funcția gestionează sesiunile utilizatorilor, facilitând reînnoirea automată a token-urilor și menținând utilizatorii autentificați fără necesitatea unei reconectări frecvente.
- Utilizarea flag-urilor specifice simplifică stocarea token-urilor și previne atacurile de tip XSS, CSRF și MITM.

2. Testarea codului

```
1 const fs = require('fs');
2 const { execSync } = require('child_process');
3 const { generateDockerfile } = require('../generateDockerfile');
4 const compilerCPP = (code, idSolution, language, _, codeNameFile, extension, problem) => {
5     fs.mkdirSync(idSolution);
6     generateDockerfile(language, idSolution, problem);
7     fs.writeFileSync(`${idSolution}/${codeNameFile}`, code);
8     const fileMemory = fs.readFileSync(`${idSolution}/${codeNameFile}`).size;
9     try {
10         let compilationResult = null;
11         try{
12             //built the container => compile the code
13             execSync(`docker build --no-cache -t ${extension}-image .`, { cwd: idSolution })
14             //check for compilation errors
15             compilationResult = execSync(`docker run -i ${extension}-image cat compilation_error.txt > ${idSolution}/compilation_error.txt`)
16             const compilationError = fs.readFileSync(`${idSolution}/compilation_error.txt`, 'utf-8');
17             if(compilationError){
18                 return {
19                     error: compilationError,
20                     memorieFisier: fileMemory,
21                     status: 0
22                 }
23             }
24         }catch(error){
25             return {
26                 error: 'The docker container could not be created',
27                 memorieFisier: fileMemory,
28                 status: 0
29             }
30         }
31         return {
32             result: compilationResult,
33             memorieFisier: fileMemory,
34             status: 1
35         }
36     } catch (error) {
37         return {
38             error: error.stdout,
39             memorieFisier: fileMemory,
40             status: 0
41         }
42     }
43 }
44 module.exports = {compilerCPP}
```

Functionalități

Exemple de cod pentru funcționalități cheie și explicarea lor

Această funcție reprezintă nucleul unui sistem de testare automată a codului. Scopul principal al acestei funcții constă în compilarea codului sursă primit și în verificarea posibilelor erori de compilare. Etapele principale sunt:

- Generarea și utilizarea unui Dockerfile pentru un mediu izolat de compilare/rulare:
 - Funcția generează un Dockerfile specific pentru fiecare limbaj de programare și problemă dată.
 - Dockerfile-ul stabilește mediul de execuție și setările necesare pentru a compila și rula codul în mod sigur și izolat, folosind containere Docker.
- Compilarea codului:
 - Funcția inițiază procesul de compilare prin intermediul comenziilor Docker. Codul este compilat într-un container Docker izolat pentru a asigura coerentă și pentru a evita dependențele de mediu.
- Capturarea erorilor de compilare:
 - În cazul în care apar erori în timpul compilării, acestea sunt capturate și redirecționate către un fișier de erori (compilation_error.txt). Ulterior, fișierul este verificat pentru a identifica eventualele erori.
- Gestionarea erorilor și returnarea rezultatelor:
 - Funcția furnizează informații despre succesul sau eșecul procesului de compilare. În cazul apariției erorilor de compilare, acestea sunt integrate în rezultatul final, împreună cu dimensiunea fișierului sursă. Dacă compilarea este finalizată cu succes, funcția returnează rezultatul compilării.

Functionalități

Exemple de cod pentru funcționalități cheie și explicarea lor

Pentru evaluarea codului, vă rog să găsiți link-ul către fișierul grader.js [aici](#): [<https://github.com/xndadelin/Info-Conquer/blob/main/server/utils/graders/grader.js>].

- Compilarea codului sursă:
 - Se apelează funcția **compiler** pentru a compila codul sursă într-un mediu izolat.
 - Dacă există erori de compilare, acestea transmise și procesul de evaluare se oprește.
- Executarea testelor:
 - Pentru fiecare test din **testCases**, se creează fișierele de intrare și ieșire corespunzătoare.
 - Se execută comanda corespunzătoare limbajului de programare în container-ul Docker, măsurând timpul și memoria utilizată, dar și captând ieșirea standard.
 - Rezultatele testelor sunt colectate și procesate conform rezultatului obținut din execuția codului.
- Returnarea rezultatelor:
 - Numele utilizatorului, codul și numele problemei.
 - Scorul total obținut.
 - Informații despre fiecare test, inclusiv statusul, timpul de execuție, memoria utilizată etc.
 - Informații despre compilare, cum ar fi eventualele erori.

Evaluare și performanță

Evaluarea performanței proiectului în raport cu obiectivele stabilite

În urma unei analize amănunțite a proiectului în conformitate cu obiectivele stabilite, constat că platforma InfoConquer a reușit să îndeplinească în mod satisfăcător majoritatea acestora.

Platforma a reușit să ofere funcționalități esențiale, cum ar fi rezolvarea problemelor de algoritmica, publicarea de articole și anunțuri, organizarea de concursuri și administrarea profilurilor utilizatorilor. Implementarea acestor funcționalități s-a desfășurat conform obiectivelor inițiale ale proiectului.

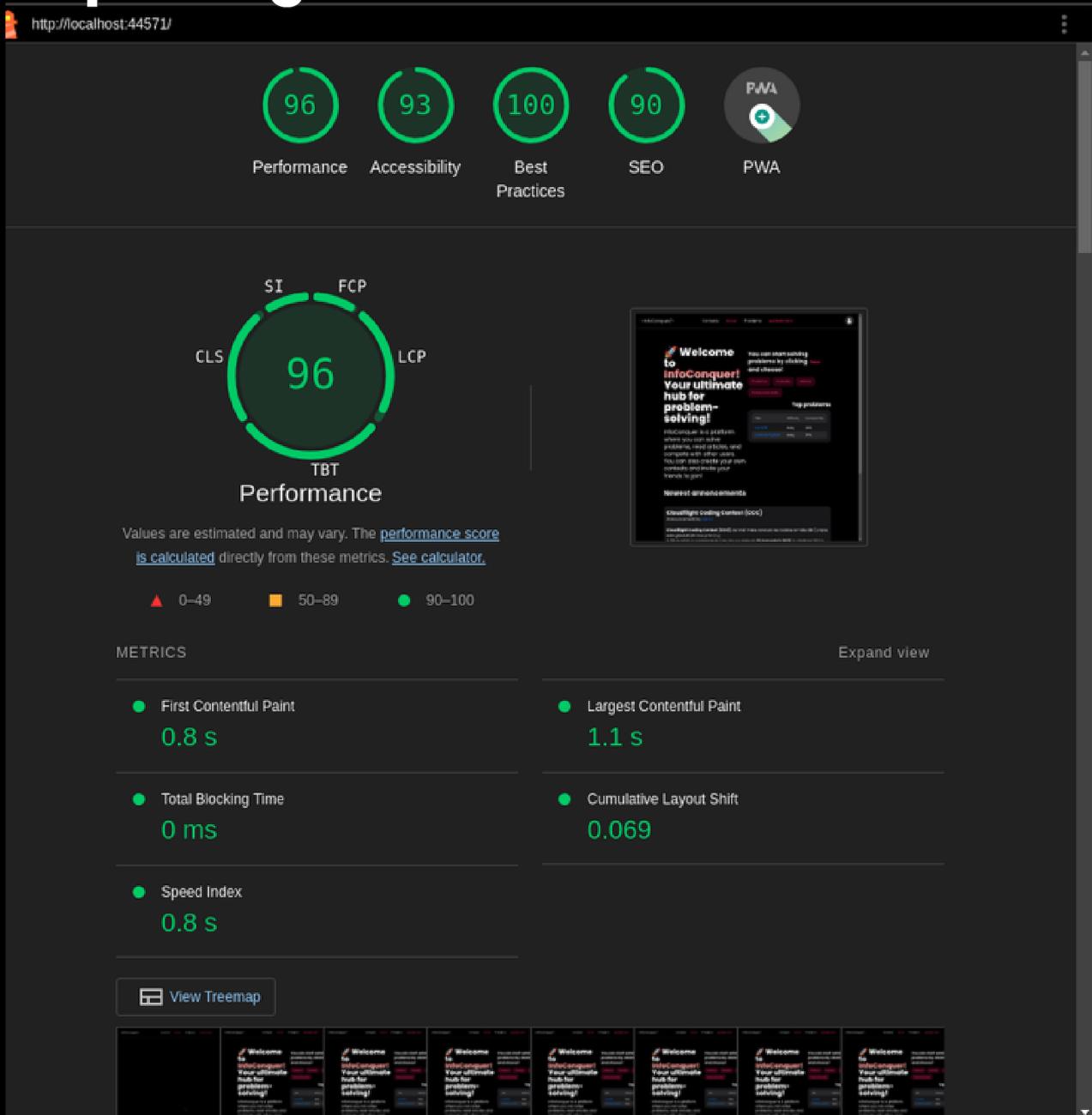
Interfața platformei a fost concepută pentru a oferi o experiență plăcută și intuitivă utilizatorilor. Navigarea facilă și accesibilitatea funcționalităților contribuie la crearea unei experiențe pozitive pentru utilizatori.

Măsurile de securitate implementate au asigurat protecția datelor utilizatorilor și integritatea platformei.

Evaluare și performanță

Analiza performanței

Raport Lighthouse



Raportul Lighthouse reprezintă un instrument creat de Google pentru evaluarea și optimizarea calității paginilor web. Acesta constituie un instrument automatizat de audit ce poate fi utilizat pentru analiza oricărei pagini web. Raportul Lighthouse oferă informații detaliate referitoare la performanța, accesibilitatea, bunele practici și optimizarea pentru motoarele de căutare a site-ului web.

Securitate

Descrierea măsurilor de securitate implementate

În cadrul proiectului, s-au implementat mai multe măsuri de securitate pentru a proteja datele utilizatorilor și a asigura integritatea platformei, reducând riscul de vulnerabilitate la posibile atacuri.

- Criptarea parolelor prin utilizarea hashing-ului (bcrypt):
 - Prin această metodă, în caz de compromitere a datelor, parolele nu vor fi expuse în format text simplu, ci vor fi criptate folosind un algoritm de hashing, împreună cu un salt, prevenind astfel utilizarea unor atacuri de tip rainbow table.
- Criptarea datelor în tranzit (doar în cadrul platformei gazduite):
 - Toate informațiile transferate între client și server sunt criptate folosind protocolul HTTPS (SSL/TLS), cu ajutorul certificatelor digitale. Această măsură asigură confidențialitatea și integritatea datelor în timpul transmiterii, protejându-le împotriva interceptării sau modificării.
- Autentificarea utilizatorilor și gestionarea sesiunilor:
 - Token-urile **JWT** sunt generate după autentificarea cu succes a utilizatorului și sunt utilizate pentru identificarea acestuia în cererile ulterioare. Fiecare token este semnat cu o cheie unică, iar cheile de semnătură sunt păstrate într-un mediu securizat. Procesul de generare și validare a token-urilor este implementat cu atenție pentru a preveni eventuale vulnerabilități de securitate.

Securitate

Descrierea măsurilor de securitate implementate

- Protectia împotriva atacurilor **CSRF** (Cross-Site Request Forgery):
 - S-au implementat măsuri de securitate menite să prevină atacurile CSRF, care pot permite atacatorilor să efectueze acțiuni neautorizate în numele utilizatorilor autentificați.
- Protecție asupra **cookie-urilor**:
 - **httpOnly: true**, previne accesul JavaScript-ului din partea client la cookie-uri, reducând riscul de **XSS**
 - **secure: true**, asigură că cookie-urile sunt trimise doar prin conexiuni HTTPS, astfel reducând şansele pentru un atac de MITM.
 - **sameSite: 'Strict'**, previne trimiterea cookie-urilor împreună cu cererile cross-site, reducând riscul de CSRF.
- Mediu **izolat** pentru rularea codului:
 - Mediul izolat pentru rularea codului reprezintă o măsură suplimentară de securitate implementată în cadrul proiectului. Aceasta asigură că codul utilizatorilor este executat într-un mediu controlat și izolat, reducând astfel riscul de exploatare a vulnerabilităților și de compromitere a sistemului.
 - Utilizez tehnologia Docker pentru a implementa mediul izolat. Docker este o platformă de containere care permite rularea aplicațiilor în containere virtualizate și independente. Aceste containere oferă o separare clară între diferitele instanțe ale aplicației și limitează impactul potențial al unui atac asupra altor componente ale sistemului.

Securitate

Descrierea măsurilor de securitate implementate

- DOMPurify
 - În cadrul proiectului, folosesc DOMPurify pentru a asigura securitatea și integritatea conținutului HTML furnizat de utilizatori sau din alte surse nesigure.

Concluzii

Sumar al rezultatelor și concluziilor obținute

În cadrul acestui proiect, am dezvoltat o platformă funcțională și securizată pentru rezolvarea problemelor de algoritmică.

Principalele caracteristici ale platformei includ:

- Sistemul de autentificare și înregistrare:
 - Utilizatorii pot înregistra conturi și se pot autentifica folosind adrese de email și parole.
 - Parolele sunt securizate prin criptare cu algoritm de hashing bcrypt, garantând securitatea acestora în situația unei scurgeri de date.
- Interfața pentru rezolvarea problemelor:
 - Utilizatorii pot selecta probleme din diverse categorii și subcategorii disponibile.
 - Fiecare problemă are o pagină dedicată ce include o descriere detaliată și un editor de cod (Monaco Editor) pentru redactarea soluțiilor.
- Crearea și gestionarea conținutului:
 - Platforma integrează pagini specializate pentru publicarea problemelor, articolelor și anunțurilor.
 - Accesul la postarea de probleme și anunțuri este restricționat doar utilizatorilor cu rol de 'admin', în timp ce articolele pot fi publicate de către oricine.
- Concursuri de algoritmică:
 - Organizarea concursurilor oferă utilizatorilor posibilitatea de a se implica în competiții de programare, facilitând astfel dezvoltarea și evaluarea competențelor lor.
- Vizualizarea profilului și a setărilor:
 - Utilizatorii beneficiază de acces la un profil personalizat, unde pot examina problemele rezolvate, soluțiile propuse și alte informații relevante.

Concluzii

Perspectivele și direcțiile viitoare pentru proiect, inclusiv dezvoltarea ulterioară a codului.

- Îmbunătățirea opțiunilor de autentificare:
 - Integrarea opțiunilor de autentificare prin Google și Discord are ca scop simplificarea procesului de înregistrare și autentificare pentru utilizatori, oferindu-le o experiență mai intuitivă și familiară.
 - Implementarea autentificării multi-factor adaugă un strat suplimentar de securitate, garantând protecția conturilor utilizatorilor împotriva accesului neautorizat.
- Implementarea rate limiting:
 - Introducerea unui sistem de limitare a ratei va preveni atacurile de tip brute force și va proteja serverul de supraîncărcare. Acest mecanism va restricționa numărul de cereri pe care un utilizator le poate face într-un interval de timp specific, îmbunătățind securitatea și performanța platformei.
- Adăugarea de funcționalități interactive:
 - Crearea unor ghiduri și tutoriale interactive va asista utilizatorii în înțelegerea mai profundă a funcționalităților platformei și în rezolvarea problemelor de algoritmica. Aceste resurse educationale vor contribui la dezvoltarea abilităților de programare ale utilizatorilor.

Prin implementarea acestor îmbunătățiri și direcții de dezvoltare, InfoConquer se angajează să furnizeze o platformă de înaltă calitate, sigură și eficientă, adaptată cerințelor utilizatorilor săi, cu scopul de a sprijini dezvoltarea abilităților acestora în programare.