

<InfoConquer/>

ADELIN SOFIAN

Clasa a X-a
Colegiul Național "Petru Rareș" Suceava
prof. Îndrumători: MARIUS UDUDEC
MIHAIL ERHAN

Cuprins

1. Introducere

a. Prezentarea proiectului	3
b. Scopul și obiectivul proiectului	4
c. Motivația și importanța proiectului	4

2. Proiectare și dezvoltare

a. Structura proiectului	5
b. Arhitectura și tehnologiile folosite	(6 - 7)
c. Backend	9
d. Baza de date	10
e. Frontend	11
f. Deployment și testing	12

3. Funcționalități

a. Creație de conținut	13
b. Autentificarea cu email și OAuth Discord	14
c. Vizualizarea și rezolvarea de probleme	15
d. Judecător de cod integrat	16
e. Asistent OpenAI integrat	(16 - 17)
f. Concursuri de algoritmica	17
g. Profiluri de utilizator	18
h. Calendar cu probleme zilnice	(18- 19)
i. Rularea codului oferit de utilizator (explicatie)....	(20 - 21)

4. Securitate

a. Hashing & JWT	(22 - 23)
b. Protecția împotriva CSRF (Cross-Site Request Forgery).....	24
c. Isolate	25
d. Dompurify	26

5. Evaluare și performanță

a. Raportul Lighthouse	
b. Server caching & rate limiting	

Introducere

Prezentarea proiectului

- InfoConquer este un website destinat programatorilor, fie că sunt începători sau avansați. Utilizatorii pot rezolva probleme de algoritmică, primind feedback bazat pe teste încărcate de pe platformă. Problemele variază de la structuri de date fundamentale la teorii avansate ale grafurilor.
- Prin cadrul acestui proiect, am reușit să creez un website care să faciliteze învățarea și perfecționarea abilităților de programare pentru utilizatori de toate nivelurile.
- Cu probleme zilnice și feedback instantaneu, utilizatorii sunt motivați să învețe în mod constant și să își îmbunătățească abilitățile de rezolvare a problemelor.
- Prin concursurile organizate pe platformă, utilizatorii au ocazia să își testeze abilitățile împotriva altor programatori, promovând un mediu competitiv și colaborativ.
- Utilizatorii pot beneficia de suportul unui asistent virtual inteligent, care oferă sugestii și îmbunătățiri pentru codul scris, facilitând învățarea prin feedback personalizat.

Introducere

Scopul și obiectivul proiectului
Motivația și importanța proiectului

- Obiectivul principal al InfoConquer este să creeze o platformă 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 si dezvoltare

Structura proiectului

- Proiectul este structurat in doua parti principale:
 - client (partea de frontend a proiectului cu care utilizatorul interactioneaza direct)
 - server (partea de backend a proiectului, responsabil pentru gestionarea cererilor de la client, procesarea lor si trimiterea datelor corespunzatoare)

Client

- Aici se regăsesc toate resursele necesare și configurațiile pentru a crea interfața cu utilizatorul și pentru a interacționa cu serverul.
- Fișierele sunt bine organizate în foldere precum **pages/**, **components/**, **assets/**, **utils/**, **styles/**, **public/**, etc.

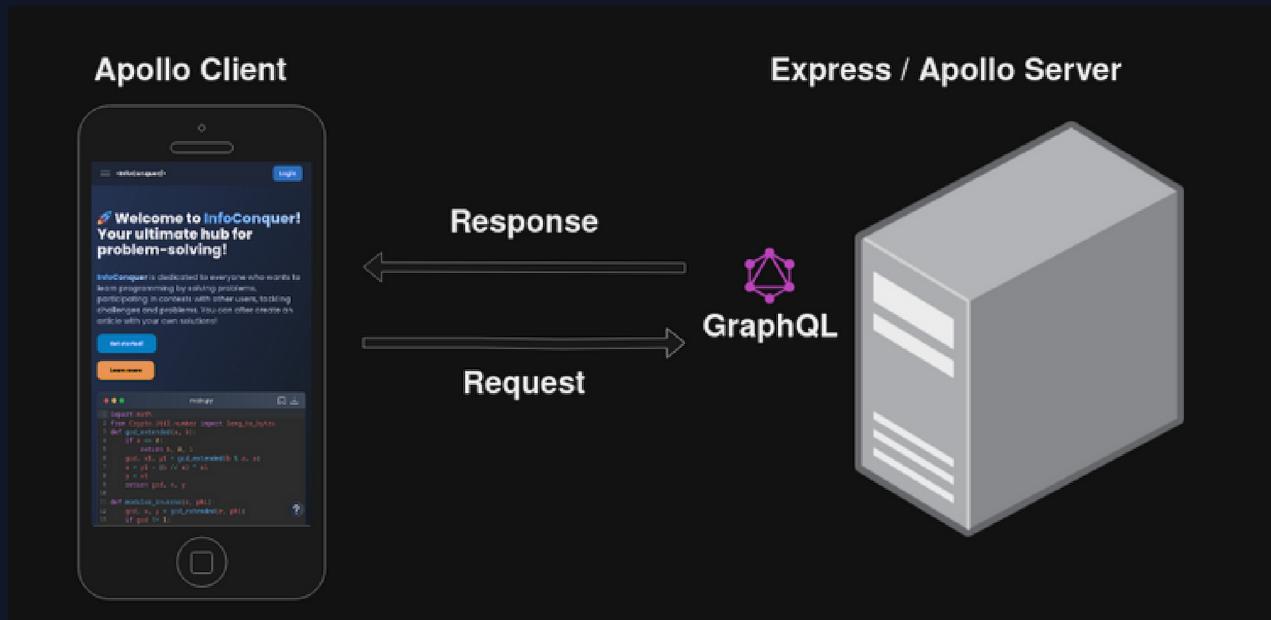
Server

- Este locul în care se regăsește toată logica de backend a aplicației web și este responsabil pentru gestionarea cererilor, interacțiunea cu baza de date, autentificare și autorizare etc.
- Fișierele sunt bine organizate in foldere, de exemplu directory-ul **graphql/** ce conține definițiile de tipuri și rezolvitorii pentru serverul GraphQL, modele pentru baza de date in **models/**, **utils/**, pentru functiile utilitare etc.

Proiectare și dezvoltare

Arhitectura și tehnologiile folosite

Arhitectură



- Proiectul prezintă o arhitectură de tip client-server, utilizând Apollo Client pentru gestionarea cererilor de date pe partea clientului și Apollo Server pentru expunerea API-ului GraphQL pe partea serverului.
- În această arhitectură, Apollo Client este responsabil pentru efectuarea cererilor GraphQL și pentru gestionarea stării aplicației în client, în timp ce Apollo Server, construit pe baza framework-ului Express, se ocupă de procesarea cererilor și furnizarea datelor solicitate.
- Această arhitectură modulară permite o dezvoltare eficientă și o separare clară a responsabilităților între client și server, facilitând scalabilitatea și întreținerea sistemului.

Proiectare și dezvoltare

Arhitectura și tehnologiile folosite

Tehnologii utilizate

- **Frontend:**

- ReactJS 
- JavaScript 
- CSS 
- HTML 
- Tailwind CSS 
- NextUI 
- Apollo Client 
- react-i18next 

- **Backend:**

- NodeJS 
- ExpressJS 
- MongoDB 
- GraphQL 
- Apollo Server 
- Isolate (<https://github.com/ioi/isolate>)

- **DevOps, deployment și testing**

- Git 
- Linode Cloud 
- Nginx 
- Cypress 
- Jest 

- **Third-Party services:**

- OpenAI 
- Cloudflare Smart Captcha 

Proiectare și dezvoltare

Backend

În cadrul proiectului, am decis să construiesc un server bazat pe un Node.js framework, express.

- 🔗 Pentru comunicarea dintre client și server, am folosit GraphQL, un limbaj de interogare pentru API-uri, ce are la bază graful de date și expune un singur endpoint: /graphql, prin care sunt trimise toate mutațiile și interogările.
- 🚀 Apollo Server funcționează că un add-on pentru framework-ul express care mă ajută să asamblez un API GraphQL.
- 📁 Ca bază de date am ales MongoDB, datorită structurii sale orientată pe documente, care îmi oferă flexibilitate. Serviciul destinat pentru a lucra cu datele din baza de date și a o hosta este MongoDB Atlas, un serviciu cloud integral.
- 🛡️ Pentru a crea un mediu izolat față de sistemul gazdă, cu scopul de a preveni codul malicios trimis de un utilizator să nu afecteze sistemul gazdă, am decis să folosesc GitHub - ioi/isolate, un sandbox construit pentru a rula programe care nu sunt sigure. Isolate se caracterizează printr-un mediul cu access limitat, prevenind afectarea sistemului gazdă. Documentație - isolate

Proiectare si dezvoltare

Baza de date

- Ca bază de date am ales MongoDB, datorită structurii sale orientată pe documente, care îmi oferă flexibilitate. Serviciul destinat pentru a lucra cu datele din baza de date și a o hosta este MongoDB Atlas, un serviciu cloud integral.

announcements	articles	contests	dailies
id <small>integer</small>	id <small>integer</small>	id <small>integer</small>	id <small>integer</small>
title <small>varchar NN</small>	title <small>varchar NN</small>	name <small>varchar NN</small>	problem <small>varchar NN</small>
content <small>text NN</small>	content <small>text NN</small>	description <small>text</small>	date <small>timestamp NN</small>
createdBy <small>varchar NN</small>	creator <small>varchar NN</small>	startDate <small>timestamp NN</small>	solvers <small>json</small>
createdAt <small>timestamp</small>	likes <small>json</small>	endDate <small>timestamp NN</small>	ended <small>boolean</small>
updatedAt <small>timestamp</small>	dislikes <small>json</small>	problems <small>json NN</small>	createdAt <small>timestamp</small>
	tags <small>json</small>	languages <small>json NN</small>	updatedAt <small>timestamp</small>
	excerpt <small>text</small>	createdBy <small>varchar NN</small>	
	createdAt <small>timestamp</small>	participants <small>json</small>	
	updatedAt <small>timestamp</small>	started <small>boolean</small>	
problems	reports	users	
id <small>integer</small>	id <small>integer</small>	id <small>integer</small>	
creator <small>varchar NN</small>	reporter <small>varchar NN</small>	username <small>varchar NN</small>	
title <small>varchar NN</small>	title <small>varchar NN</small>	email <small>varchar</small>	
description <small>text</small>	description <small>text NN</small>	password <small>varchar</small>	
requirements <small>text NN</small>	type <small>varchar NN</small>	verified <small>boolean</small>	
tags <small>json</small>	problem <small>varchar</small>	admin <small>boolean NN</small>	
category <small>varchar NN</small>	createdAt <small>timestamp</small>	solutions <small>json</small>	
subcategories <small>json NN</small>	updatedAt <small>timestamp</small>	solvedProblems <small>json</small>	
input <small>text</small>		codeForVerification <small>varchar</small>	
output <small>text</small>		lastEmailVerification <small>timestamp</small>	
tests <small>json NN</small>		discordID <small>varchar</small>	
timeExecution <small>float NN</small>		activity <small>json</small>	
limitMemory <small>float NN</small>		profilePicture <small>varchar</small>	
examples <small>json</small>		bio <small>text</small>	
indications <small>text</small>		createdAt <small>timestamp</small>	
languages <small>json NN</small>		updatedAt <small>timestamp</small>	
difficulty <small>varchar NN</small>			
restriction <small>text</small>			
itsForContest <small>boolean NN</small>			
rejectedSolutions <small>integer</small>			
acceptedSolutions <small>integer</small>			
successRate <small>float</small>			
rating <small>float</small>			
ratings <small>json</small>			
createdAt <small>timestamp</small>			
updatedAt <small>timestamp</small>			

Proiectare și dezvoltare

Frontend

Pentru construirea interfețelor de utilizator, am folosit librăria React, facilitând crearea de componente reutilizabile.

- Stilizarea interfeței de utilizator a fost făcută cu [Tailwind CSS](#), împreună cu [Next UI](#), o librărie de componente moderne.
- Cu ajutorul componentei CodeMirror, care este integrată pentru a oferi un editor de cod robust și interactiv, permite utilizatorilor să scrie și să editeze soluții pentru problemele propuse într-un mediu familiar și eficient.
- Folosind framework-ul [i18next](#), am reușit să fac posibilă internaționalizarea conținutului, oferind suport multi-lingv.
- Folosind [Apollo Client](#), o librărie pentru gestionarea stării aplicației și pentru comunicarea eficientă cu serverul backend cu GraphQL, am reușit să fac platforma mai scalabilă, reducând traficul de rețea și timpul de răspuns al aplicației, aducând DOAR datele necesare.

Proiectare și dezvoltare

Deployment și testing

În acest caz, folosesc Linode pentru gazduire și Nginx ca server web.

- Folosesc Linode pentru a găzdui aplicația. Linode oferă un VPS cu 4GB RAM, 2 CPU cores și 80GB de stocare, care este suficient pentru InfoConquer.
- Nginx este utilizat pentru a gestiona cererile HTTP și a le direcționa către aplicație. Este configurat pentru a servi atât clientul, cât și serverul Node.js.

Pentru a asigura calitatea aplicației InfoConquer, folosesc teste end-to-end și teste unități / de integrare.

- Testare End-to-End cu Cypress:
 - Cypress este utilizat pentru a verifica funcționalitatea completă a aplicației, asigurându-ne că toate componentele lucrează împreună conform așteptărilor.
- Testare Unități cu Jest:
 - Jest este utilizat pentru a testa funcțiile individuale și modulele aplicației pentru a asigura că logica aplicației funcționează corect.

Functionalități

Descrierea funcționalităților oferite de proiect



Creație de conținut

- Pentru a asigura o experiență completă și eficientă utilizatorilor, am dezvoltat pagini dedicate creării conținutului, inclusiv probleme, articole și anunțuri.
- Aceste 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, articole sau anunțuri.
- Pentru a garanta securitatea conținutului furnizat și pentru a preveni posibile vulnerabilități de tip XSS (cross-site scripting), am implementat soluția "DOMPurify". 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.

The screenshot shows a dark-themed web application interface for publishing a problem. At the top, there's a navigation bar with links for 'Contests', 'Problems', 'Leaderboard', and a search icon. Below the navigation is a large, light-colored input form titled 'Publish a Problem'. The first section, 'Basic Information', contains fields for 'Title' (with a required asterisk), 'Description', and 'Requirements'. The second section, 'Tags and Categories', includes a 'Add Tag' button, a 'Category' dropdown, and a question mark icon in the bottom right corner of the form area.

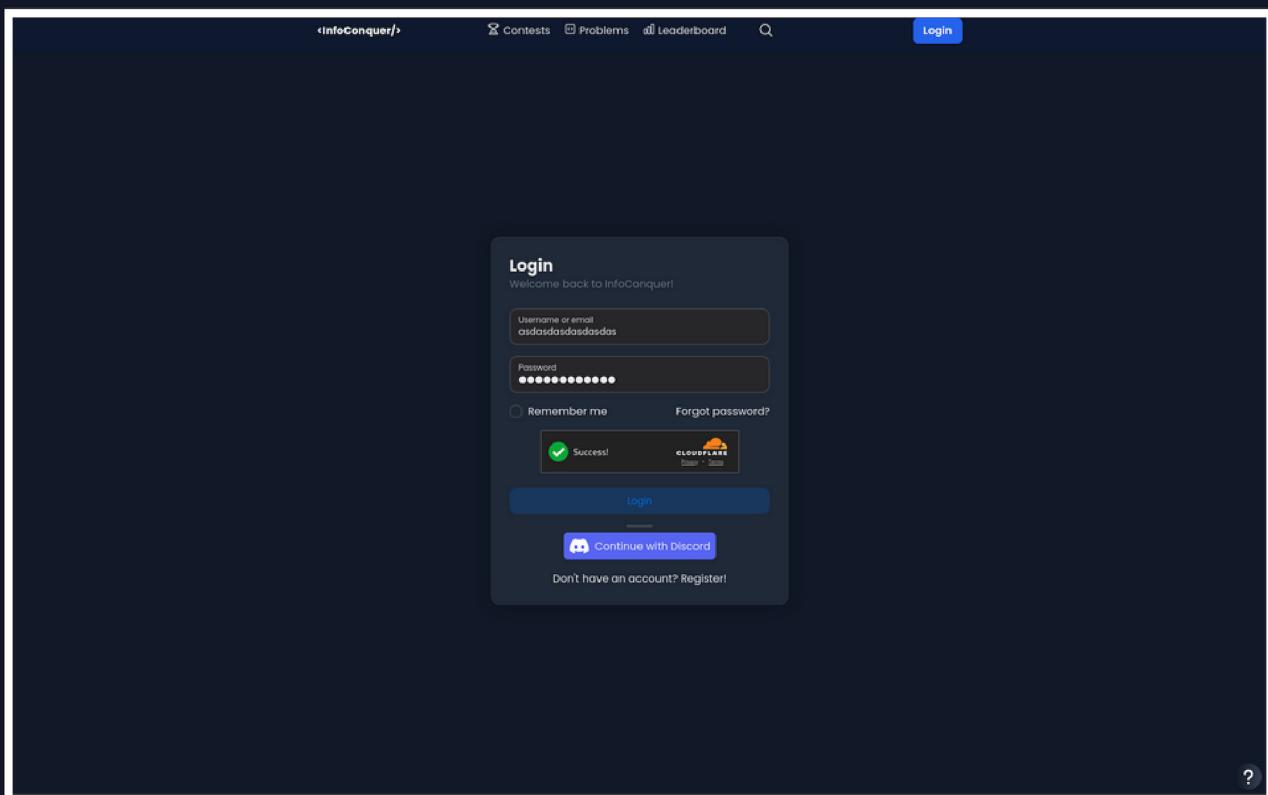
Functionalități

Descrierea funcționalităților oferite de proiect

🔒 Autentificarea cu email și OAuth

Discord

- ✉️ Autentificare prin email:
 - 🔑 Creare cont: Utilizatorii își pot crea un cont folosind adresa de email și o parolă.
 - 🔄 Resetare parolă: O opțiune pentru resetarea parolei prin trimitera unui link de resetare pe email.
 - 🔗 Verificare email: După crearea contului, utilizatorii trebuie să-și verifice adresa de email printr-un link trimis automat.
- 🔗 Utilizatorii pot face autentificare rapidă folosind contul lor Discord.

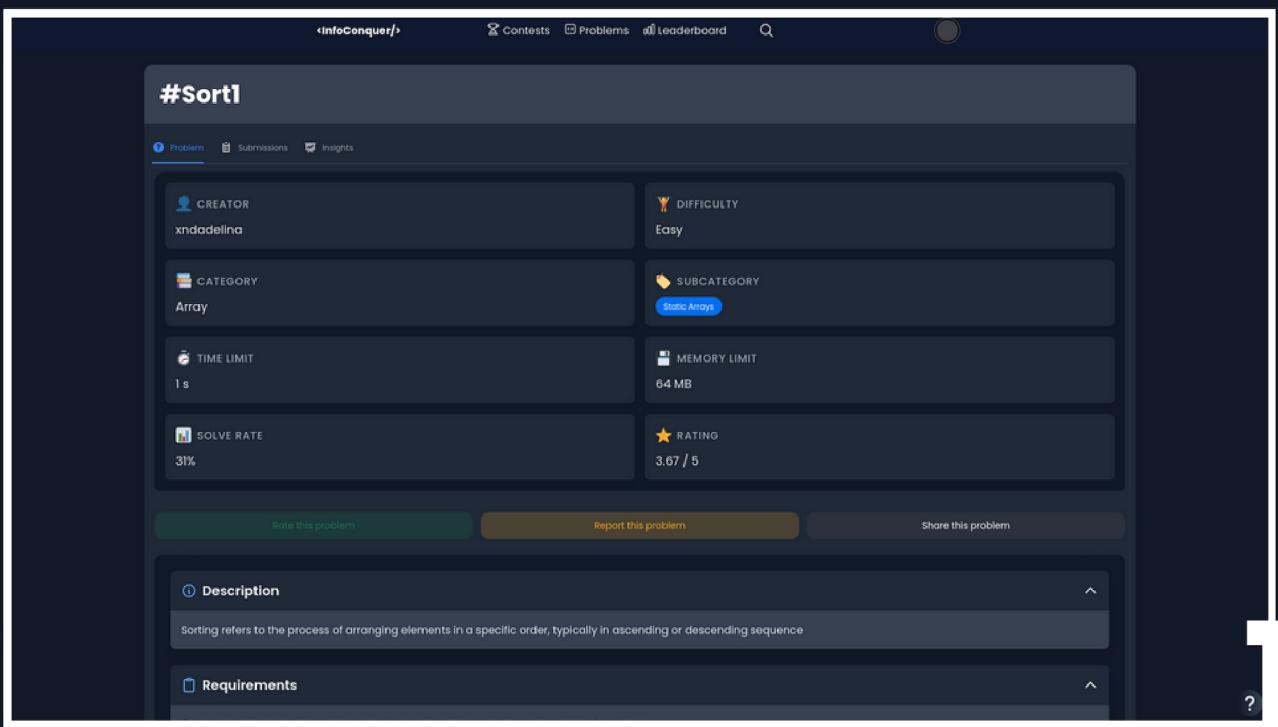


Functionalități

Descrierea funcționalităților oferite de proiect

Vizualizarea și rezolvarea de probleme

-  Utilizatorii pot rezolva probleme de algoritmică într-un limbaj de programare la alegere
-  Funcționalitate supusă unui formular completat de un admin, care reprezintă toate detaliile problemei (descriere, exemple, ieșire, intrare, limita de timp, limita de memorie, teste etc.)
-  Editor de cod integrat <https://uiwjs.github.io/react-codemirror/>
-  Statisticile problemei (grafic cu numărul de rezolvări în ultimele 7 zile, primele submisii acceptate, cele mai bune soluții în ceea ce privește timpul și memoria)
-  Aplicația permite vizualizarea detaliată a soluțiilor pentru problemele de programare, oferind informații esențiale despre fiecare soluție, inclusiv codul sursă, rezultatele fiecărui test și alte detalii relevante.



The screenshot shows a dark-themed web application interface for a programming competition. At the top, there's a navigation bar with links for 'Contests', 'Problems', 'Leaderboard', and a search icon. Below the navigation, a specific problem card is displayed for a challenge titled '#Sort1'. The card contains the following details:

- Creator:** xndadelina
- Difficulty:** Easy
- Category:** Array
- Subcategory:** String Arrays
- Time Limit:** 1 s
- Memory Limit:** 64 MB
- Solve Rate:** 31%
- Rating:** 3.67 / 5

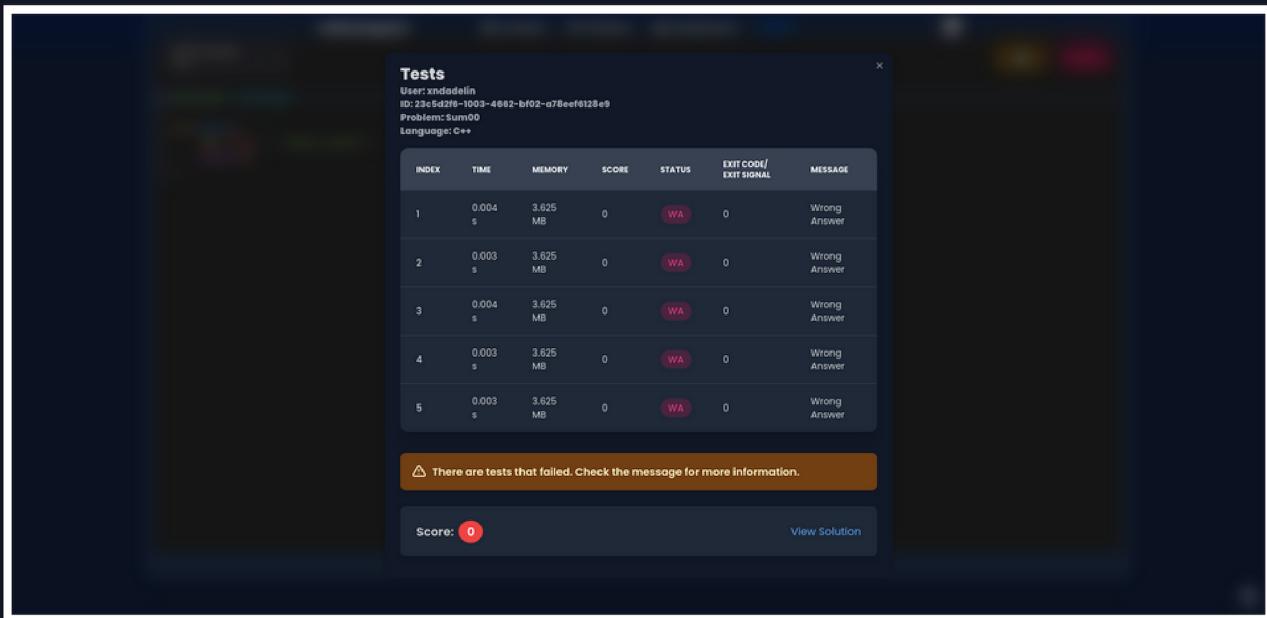
Below the card, there are three buttons: 'Rate this problem', 'Report this problem', and 'Share this problem'. Further down, there are sections for 'Description' and 'Requirements'. The 'Description' section contains the text: "Sorting refers to the process of arranging elements in a specific order, typically in ascending or descending sequence". The 'Requirements' section is partially visible at the bottom.

Functionalități

Descrierea funcționalităților oferite de proiect

Judecător de cod integrat

-  Evaluează automat soluțiile trimise de utilizatori, oferind feedback instantaneu și detaliat pe baza testelor predefinite
-  Folosește un sandbox securizat, similar cu cel utilizat la IOI (International Olympiad in Informatics)
-  Mai multe detalii: (<https://github.com/ioi/isolate>)

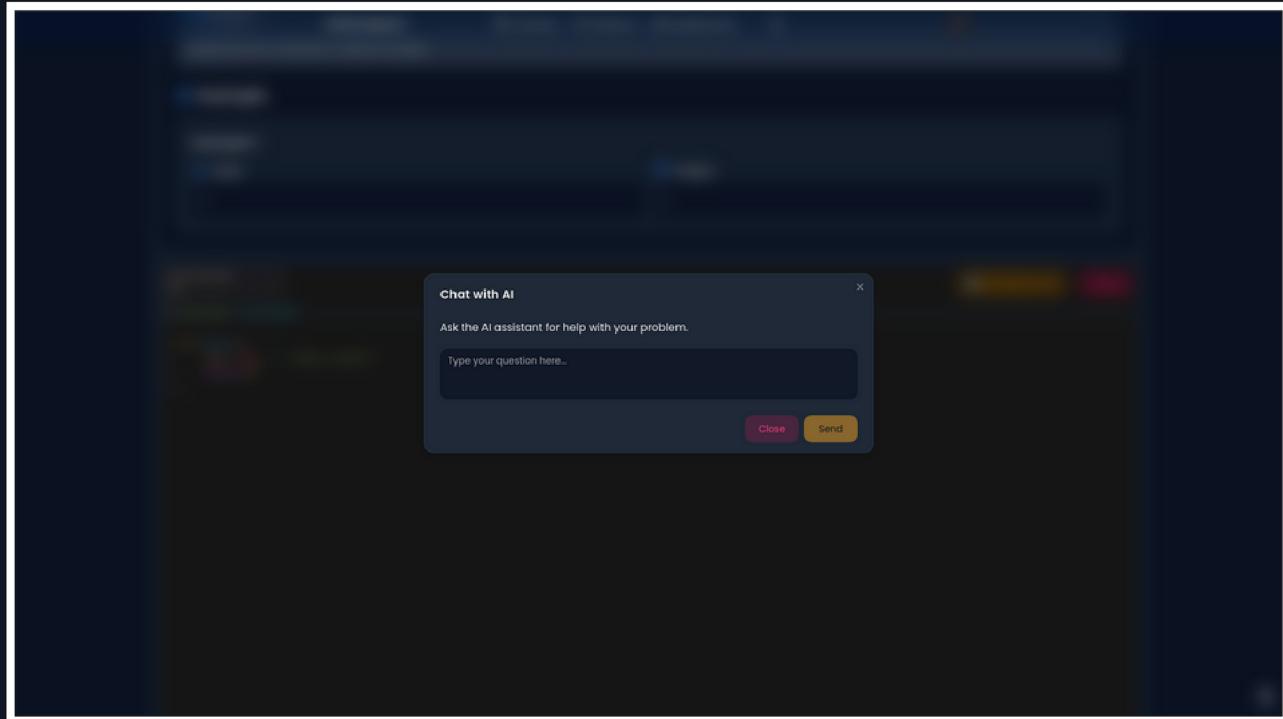


Asistent OpenAI integrat

-  Încorporat în editorul de cod, dar și în editorul de text TinyMCE
-  Ajută utilizatorii să primească feedback asupra codului lor, menționând ce poate să optimizeze, ce să schimbe și să-i ajute să rezolve problema
-  Folosește modelul GPT-4 de la [OpenAI] (<https://openai.com/>)

Functionalități

Descrierea funcționalităților oferite de proiect



🏆 Concursuri de algoritmică

- 🌐 Platforma poate găzdui concursuri pentru toți utilizatorii, indiferent de nivel
- 📊 Pagina concursului include clasament în timp real, detaliile concursului și un grafic privind solvabilitatea problemelor

InfoEducatie

Creare de: xndadelin
Data de început: 6/2/2024, 12:00:00 AM
Data de sfârșit: 8/3/2024, 12:00:00 AM
Durată: 1032.00 hours
Limbaj: C#, C++, C
Descriere: Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum
Probleme: Sum00, CaesarCipher, Sort, SumSquares
Participanți: 5
A început: Da
S-a terminat: Nu

Clasament

RANK	USERNAME	SUM00	CAESARCIPHER	SORTI	SUMSQUARES	TOTAL
1	xndadelin	100	100	100	100	400
2	xndadelin	100	0	0	100	200
3	xnd.ovi	0	0	0	0	0
4	Alex	0	0	0	0	0
5	xndadelin	0	0	0	0	0

Previous Next ?

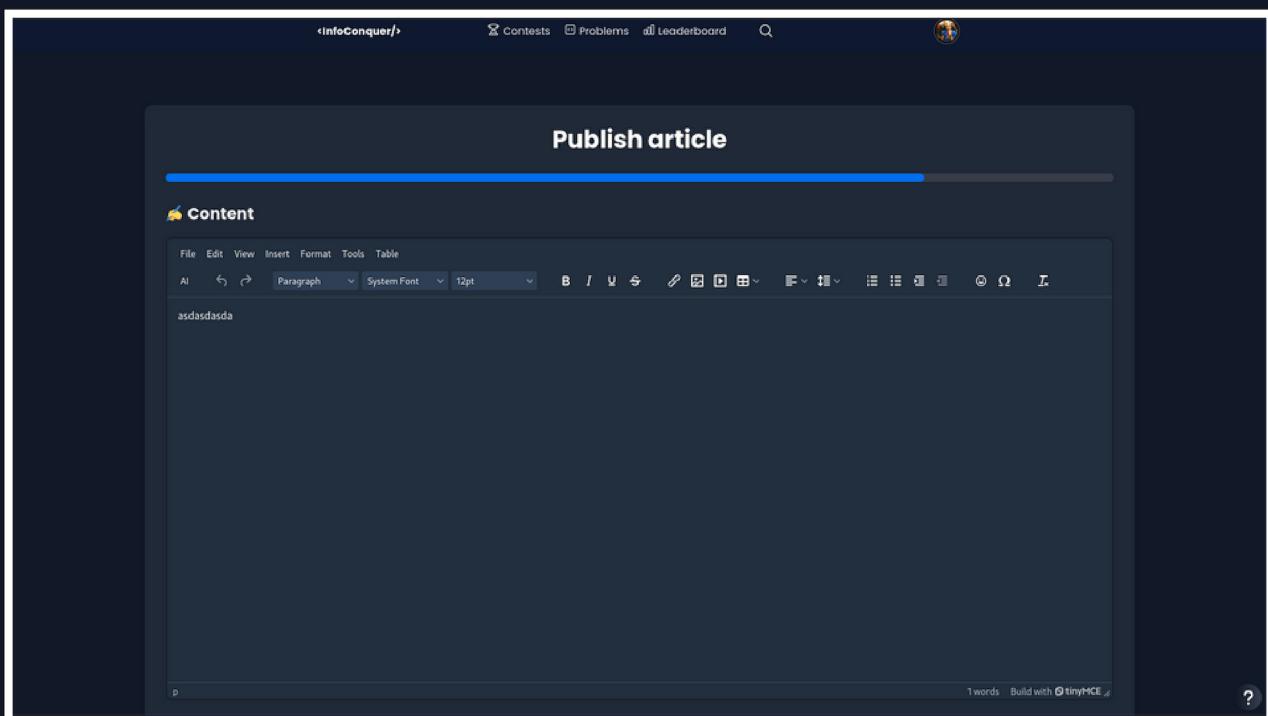
Functionalități

Descrierea funcționalităților oferite de proiect



Postarea de articole și anunțuri

- Utilizatorii admini au posibilitatea de a posta articole și anunțuri, cu ajutorul editorului de text TinyMCE, la care am adăugat un add-on unde utilizatorii pot cere asistență de la Inteligența Artificială
- Funcționalitate de creare și publicare de anunțuri
- Cititorul are opțiunea de a aprecia / dezaprecia articolul



Profiluri de utilizator

- Informații generale despre utilizator (avatar, nume, status, dată creării)
- Statistici (probleme rezolvate, submisii totale, soluții acceptate)
- Vizualizare activitate
- Setări de profil (editare poză de profil, email, username, parolă, biografie)

Functionalități

Descrierea funcționalităților oferite de proiect

The screenshot shows a user profile for 'xndadelin'. The profile includes a circular profile picture, the username 'xndadelin', the title 'Admin', and the creation date 'Created at: 7/6/2024, 8:56:17 PM'. Below this, there are three statistics boxes: '3 Problems solved' (blue), '97 Total submissions' (purple), and '38 Accepted solutions' (green). Below the stats, there are tabs for 'General Information' and 'Settings'. Under 'Solved Problems', there is a table with columns for PROBLEM, LANGUAGE, SCORE, STATUS, and DATE. The table lists four entries for 'Sum00' solved in C++ with a score of 100 and status 'Accepted' on 7/6/2024. At the bottom right of the profile section is a question mark icon.



Calendar cu probleme zilnice

- 🎲 Propune o problemă aleatorie în fiecare zi
- 🏆 Încurajează utilizatorii să-și mențină seria de rezolvare, conservând o activitate mai considerabilă

The screenshot shows a monthly calendar for July 2024. Each day of the month has a corresponding colored box indicating the problem assigned for that day. The colors represent different problem types or categories. The days are labeled with their date, day of the week, and the problem name. For example, July 1st is a Monday with the problem 'CaesarCipher', while July 23rd is a Tuesday with the problem 'TEST'. Some days like July 7th, 14th, and 21st have multiple problems listed. The bottom row of the calendar shows days 28 through 31, all of which are labeled 'No daily yet.'

Functionalități

Rularea codului oferit de utilizator

Această funcționalitate permite rularea codului sursă oferit de utilizator într-un mediu izolat, verificându-l împotriva unui set de teste și raportând rezultatele. M-am folosit de Isolate, un sandbox securizat, construit pentru a rula programe nesigure, oferind un acces limitat la mediu, prevenind codul malicios să infiltreze în sistemul gazdă.

Cum funcționează?

- **Initializare:** 
 - Generează un ID unic pentru fiecare soluție și înregistrează timpul de primire.
- **Pregătirea sandbox-ului:** 
 - Creează un mediu izolat pentru rularea codului.
 - Populează sandbox-ul cu codul sursă și fișiere relevante.
- **Compilarea codului:** 
 - Compilă codul sursă folosind comanda specifică limbajului de programare.
 - Dacă este necesar, aplică cerințe suplimentare (de exemplu, crearea unui JAR pentru Java).
- **Verificarea erorilor:**
 - Verifică și raportează erorile de compilare dacă acestea există.
- **Rularea testelor:** 
 - Răspunde fiecărui caz de testare cu input-ul furnizat.
 - Înregistrează rezultatul, timpul de execuție și utilizarea memoriei.

Functionalități

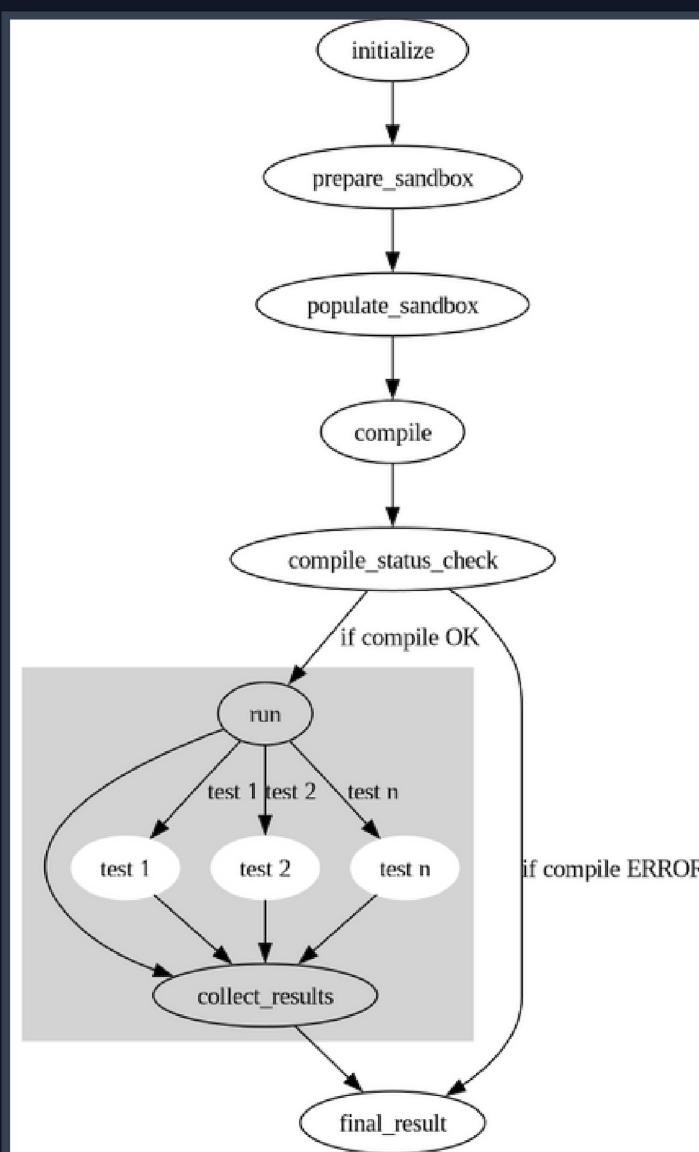
Rularea codului oferit de utilizator

- **Evaluarea Rezultatelor:** 

- Compară ieșirea generată cu ieșirea așteptată.
- Verifică dacă timpul și utilizarea memoriei sunt în limitele specificate.
- Clasifică rezultatele ca fiind "Accepted" (**AC**), "Wrong Answer" (**WA**), "Time Limit Exceeded" (**TLE**), "Memory Limit Exceeded" (**MLE**), etc.

- **Raportare:** 

- Înțocmește un raport detaliat despre fiecare test.
- Include rezultate, erori de compilare, utilizarea memoriei și timpul de execuție.



grader
workflow

Securitate

Hashing and JWT

Securitatea este un aspect esențial al platformei InfoConquer, având în vedere că utilizatorii interacționează cu serverul și își gestionează datele personale.

-  **Hashing-ul parolelor**

- Hashing-ul este o tehnică folosită pentru a securiza parolele utilizatorilor înainte de a le stoca în baza de date. În InfoConquer, utilizăm biblioteca bcrypt pentru a realiza acest lucru. bcrypt este o funcție de hash rezistentă la atacuri brute-force, care include un salt pentru a îmbunătăți securitatea. Fiecare parolă este amestecată cu un salt unic înainte de hashing, ceea ce face ca două parole identice să aibă hash-uri diferite.

-  **JWT (JSON Web Tokens)**

- Pentru autentificare, InfoConquer folosește JSON Web Tokens (JWT). JWT este un standard pentru transmiterea informațiilor între două părți într-un mod compact și securizat, folosind un token digital. Acestea sunt folosit pentru autentificarea și autorizarea utilizatorului.
- Atunci când utilizatorul se conectează serverul emite două tokene, unul principal și unul de reimprospătare, pentru ca utilizatorul să stea conectat și pentru a nu fie nevoie să se reconecteze la un moment prestabil.

Securitate

Hashing and JWT

- Primul token este utilizat pentru a accesa resurse protejate de pe server. El conține informații despre utilizator și drepturile acestuia, și trebuie inclus în fiecare cerere API pentru a verifica accesul. Are o durată de viață de 15 minute.
- Al doilea token permite utilizatorului să obțină un nou access token fără a trebui să se autentifice din nou. Este utilizat pentru a prelungi sesiunea utilizatorului fără a necesita reintroducerea credențialelor. Are o durată de viață de 7 zile.
- Token-urile sunt transmise prin HTTPS, pentru a nu fi interceptate, și sunt stocate ca cookie-uri cu flaguri de securitate pentru a minimiza riscurile asociate cu expunerea sau furtul acestora. Aceste flaguri includ:

Securitate

Protectia împotriva CSRF (Cross-Site Request Forgery)

CSRF (Cross-Site Request Forgery) este un tip de atac în care un atacator induce un utilizator autenticat să execute acțiuni nedorite pe un site web pe care acesta este deja autenticat. Atacurile CSRF profită de încrederea pe care un site o are în browserul utilizatorului pentru a executa cereri neautorizate în numele acestuia.

- La fiecare cerere, serverul trebuie să verifice dacă există un token CSRF valid. Acest token, generat și asociat cu sesiunea utilizatorului, este trimis împreună cu cererea și validat de server pentru a preveni atacurile CSRF.

```
const { getUser } = require('./getUser');
const cookie = require('cookie');

module.exports = async(req, res, next) => {
    const context = {req, res}
    const user = await getUser(context);

    if(user) {
        const tokenFromCookie = cookie.parse(req.headers.cookie).csrfToken;
        const tokenFromHeader = req.headers['x-csrf-token'];

        if(tokenFromCookie && tokenFromHeader && tokenFromCookie === tokenFromHeader) next();
        else res.status(403).json({message: 'CSRF token is missing or invalid'});
    }else{
        next();
    }
}
```

Securitate

Isolate

- Isolate este un instrument pentru crearea unui mediu controlat în care procesele pot fi rulate fără a avea acces la resurse externe sau fără a influența alte procese.
- Utilizând Linux Containers, Isolate permite rularea aplicațiilor într-un sandbox, limitându-le resursele și accesul la sistemul de fișiere.
- Pentru a evita conflictele, Isolate gestionează execuția instanțelor, așteptând finalizarea unei instanțe înainte de a începe alta dacă sunt mai multe în așteptare. (deci, nu este necesara implementarea unei cozi)

```
const run = (language, sandboxPath, testCase, inputPath, outputPath, memory, runtime) => {
  const command = languages[language].run;

  fs.writeFileSync(inputPath, testCase.input);
  fs.writeFileSync(outputPath, '');
  let exitcode = null, exitsig = null, killed = null, max_rss = null, message = null, status =
  null, time = null;

  try {
    execSync(`isolate --box-id=1 --wait --mem=${memory} --time=${runtime}
--meta=${path.join(sandboxPath, 'box', 'meta.txt')} --stderr=cerr.txt --stdin=input.txt
--stdout=output.txt --run -- "${command}"`, { cwd: path.join(sandboxPath, 'box') });
  } catch (error) {
    return {
      username,
      code,
      problem,
      language,
      score: 0,
      tests: [],
      fileMemory: get_file_size(sandboxPath, languages[language].file),
      date: new Date(),
      compilationError: null,
      success: false,
      id_solution: idSolution,
      status: 'ERROR'
    }
  }

  const meta = read_meta(sandboxPath)

  exitcode = meta.exitcode ? parseInt(meta.exitcode) : null;
  exitsig = meta.exitsig ? parseInt(meta.exitsig) : null;
  killed = meta.killed ?? null;
  max_rss = meta['max-rss'] ? parseInt(meta['max-rss']) : null;
  message = meta.message ?? null;
  status = meta.status ?? null;
  time = meta.time ? parseFloat(meta.time) : null;

  const output = fs.readFileSync(outputPath).toString();
  const cerr = fs.readFileSync(path.join(sandboxPath, 'box', 'cerr.txt')).toString()

  return { output, exitcode, exitsig, killed, max_rss, message, status, time, cerr };
}
```

Securitate

Dompurify

DOMPurify este o bibliotecă JavaScript folosită pentru a curăța conținutul HTML de cod malicios sau nevalid. Este utilizată pentru a preveni atacuri de tip Cross-Site Scripting (xss) prin eliminarea scripturilor sau a altor elemente potențial periculoase din documentele web. DOMPurify asigură că datele care sunt prezentate utilizatorilor sunt sigure și nu conțin cod rău intenționat.

Exemplu:

```
<script>alert('This is a malicious script!');</script>
<p>This is a normal paragraph.</p>
```

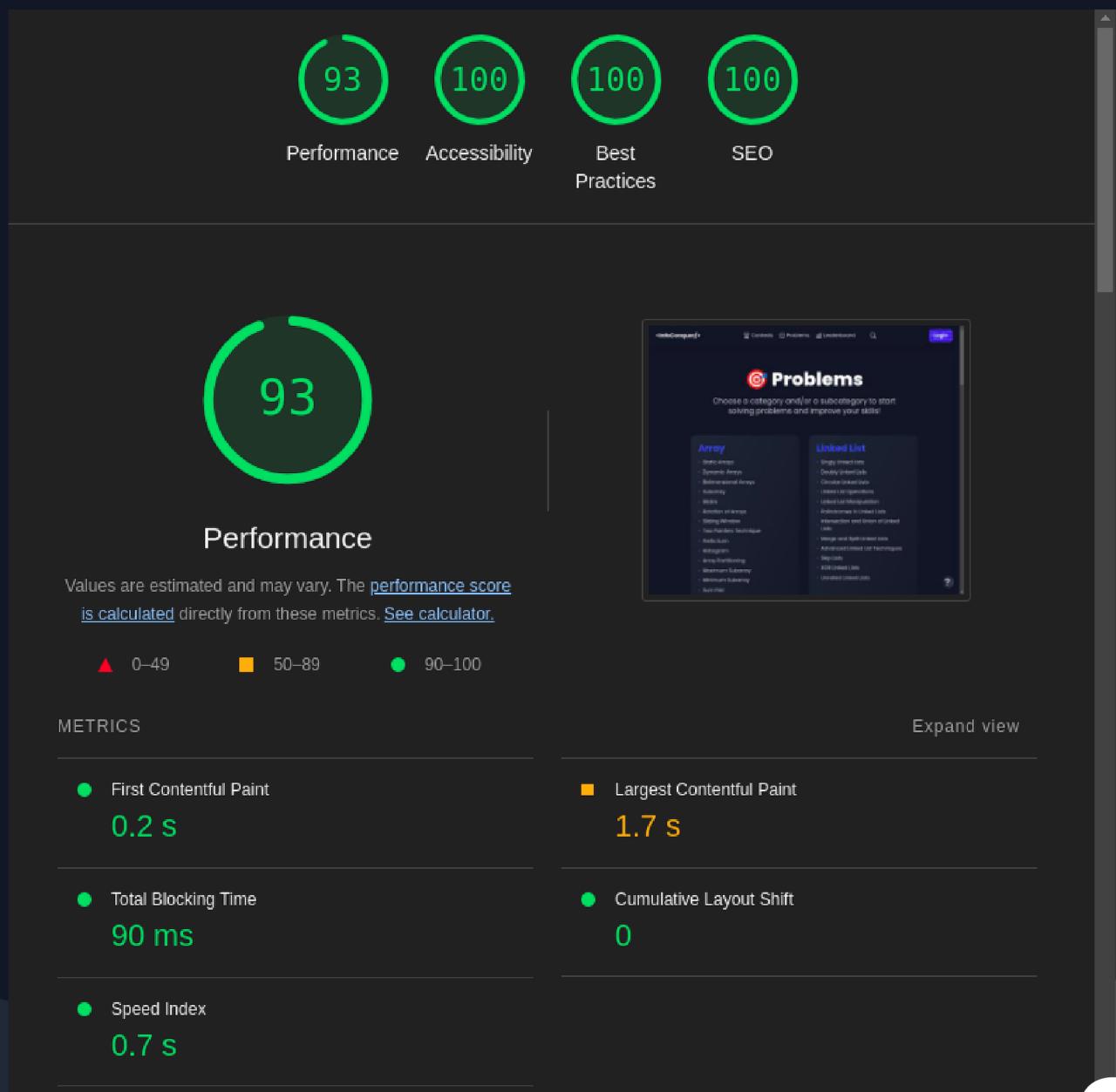
- Acesta un script JavaScript malicios care va afișa o alertă atunci când HTML-ul este executat. Scopul este de a arăta cum un atacator ar putea insera cod malicios într-o aplicație web.
- DOMPurify va elimina tag-ul `<script>` și orice conținut din el, lăsând doar conținutul permis. În acest caz, rezultatul curățat va fi:

```
// the script has been removed from the input
<p>This is a normal paragraph.</p>
```

Evaluare și perfomanță

Raportul 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 optimizare pentru motoarele de căutare a site-ului web.



Evaluare și perfomanță

Server caching & rate limiting

- Caching-ul serverului este o tehnică utilizată pentru a îmbunătăți performanța și eficiența unui server web. Scopul principal este de a reduce timpul de răspuns și sarcina pe server prin stocarea temporară a răspunsurilor la cereri frecvent efectuate.
- Atunci când un client face o cerere pentru o resursă care a fost deja stocată în cache, serverul poate returna această resursă din cache în loc să o genereze din nou, economisind astfel timp și resurse.
- Rate limiting este o tehnică folosită pentru a controla și restricționa numărul de cereri (requests) pe care un client le poate face către server într-o anumită perioadă de timp.
- Scopul principal al rate limiting-ului este să protejeze serverul de suprasarcină, să prevină abuzurile și să îmbunătățească performanța și securitatea aplicațiilor web. (aka DDOS attacks)
- Actual, sunt două rate limiting-uri: unul bazat pe IP, limitând la 10 cereri pe secundă per IP cu un burst de 20, și unul global, limitând la 1000 cereri pe secundă cu un burst de 500 pentru întregul server. Acestea ajută la prevenirea abuzurilor și menținerea performanței serverului. (pot fi modificate între timp, în funcție de traficul site-ului).

Evaluare și perfomanță

Server caching & rate limiting

- În plus, caching-ul este configurat pentru a stoca răspunsurile în cache, cu o dimensiune maximă de 10 GB și un timp de valabilitate de 1 minut pentru răspunsuri generale și 10 minute pentru codurile de stare 200, 301 și 302. Aceste setări ajută îmbunătățirea performanței și reducerea încărcării serverului.