

Lucrare de masterat

Student: Fermuș V. Vasile-Octavian

Specializare: Sisteme Distribuite

Profesor coordonator: Dr. Vasile Alaiba



Funcționalități ale proiectului și cerințe/posibile implementări

Cuprins

1. Introducere
2. Generalități
   1. Front-End
   2. Back-End
3. Funcționalități
   1. Sistem de înregistrare/logare
   2. Sistem de mesagerie
   3. Sistem de postare a algoritmilor
   4. Sistem de cerere algoritmi
   5. Sistem de comentare/notare/adnotare a codului
   6. Sistem de notare a algoritmilor
   7. Căutare după popularitate/nume/limbaj de programare
   8. Suport multi-language
   9. Sistem de report
4. Concluzii
5. Introducere

Am ales acest tip de proiect din cauza faptului că am un an de experiență în domeniul Web, specializându-mă pe partea de Front-End, cu (mici) înclinații spre partea de Back-End.

Cred că o aplicație în care diferite persoane pot adăuga algoritmi pentru a fi notați/comentați și găsiți cu ușurință poate fi folositoare în viața de zi cu zi a unui programator în orice domeniu; posibilitatea de a arăta succint eventuale greșeli și notarea corectitudinii ajutând programatori din diferite domenii în munca lor ar fi foarte folositoare.

Mai mult de atât, în timp existența acestei aplicații Web ar oferi o alternativă simplă la motoarele de căutare (în cazul în care aplicația de față are succes în rândul utilizatorilor), cu feedback continuu din partea utilizatorilor.

Un alt motiv pentru care aleg această temă este pentru a învăța mai bine limbajul PHP și MySQL, nefiind familiarizat cu subtilitățile acestora. Voi prezenta în continuare modul în care îmi propun să pun în aplicare această lucrare precum și o prezentare succintă a fiecărei componente.

1. Generalități
   1. **Front-End**

Pentru interfață plănuiesc să folosesc pagini blade, pentru ușurință spre ceea ce privește codul HTML repetat (header/footer) și posibilitatea de a adăuga templates, pentru stilizare având în vedere CSS creat prin SASS. Pentru compilarea paginilor SASS și minimizarea fișierelor de orice tip folosi Grunt și plugin-urile oferite online, precum grunt-contrib-sass și grunt-contrib-uglify.

Pentru afișarea conținutului dinamic se va utiliza Javascript, folosind jQuery, preluarea conținutului dinamic al paginilor fiind realizate prin requesturi AJAX la server. Se are în vedere folosirea framework-ului jQuery-UI pentru unele elemente ale interfeței, precum și alte framework-uri în cazul în care este necesar.

Pentru partea responsive a site-ului se va folosi unul din framework-urile deja existente, Bootstrap.

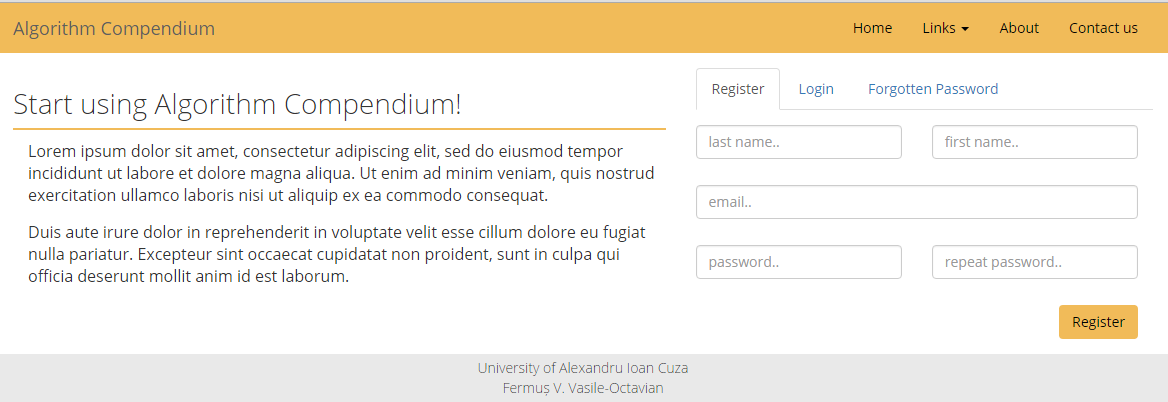
* 1. **Back-end**

Voi utiliza PHP pentru persistență, folosind baze de date MySQL, folosind în cea mai mare parte API-uri pentru fiecare request în parte. Conținutul datelor primit prin API-uri de la server va fi prezentat în format JSON, decodat de Javascript și folosit pentru afișarea datelor respective, precum și conținutul trimis la server. Folosind API-uri combinate cu requesturi AJAX vom avea și o fluiditate mai mare în utilizarea site-ului din cauză că nu mai este nevoie de “refresh” la fiecare request către server în parte, ci doar la schimbarea paginilor. Frameworkul pe care doresc să îl folosesc este Laravel 5, folosind versiunile de PHP și MySQL oferite de XAMPP.

1. Funcționalități
   1. **Sistem de înregistrare/logare**

Vom avea o pagină simplă de logare/înregistrare, cele două formulare fiind comutate printr-un singur buton folosind Javascript. Se va face un request la server pentru introducerea utilizatorului în baza de date, respectiv logarea sa. Securitatea va fi în mare parte întreținută de Laravel. Parola nu va fi salvată în baza de date, fiind folosit un hash.

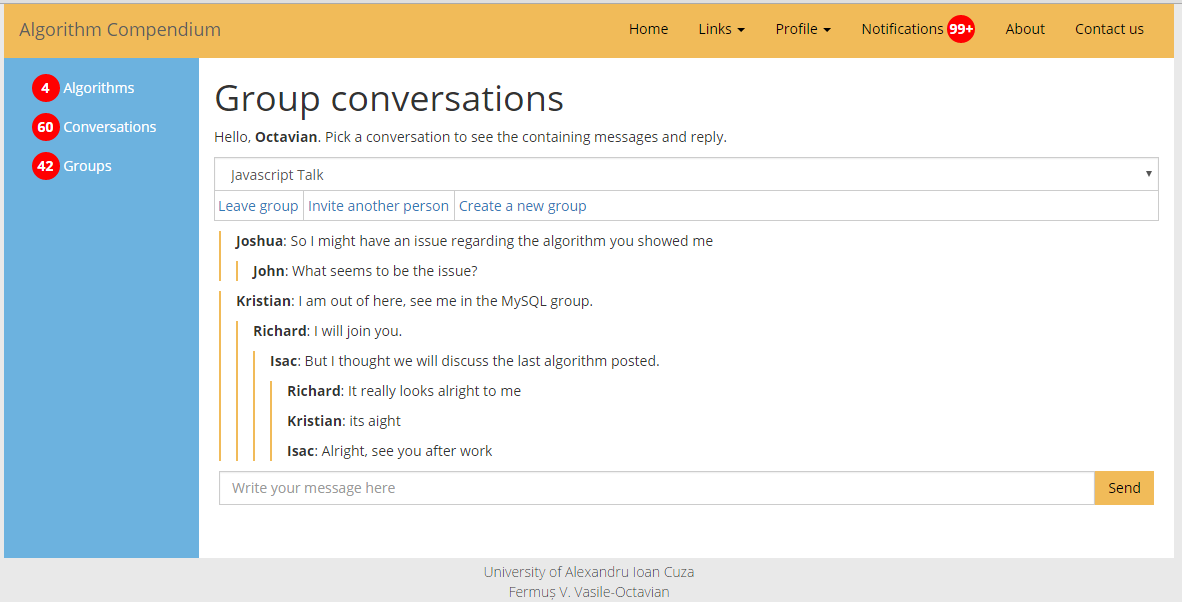
Ca și funcționalități specifice, utilizatorul va avea opțiunea de a-și închide contul pe o perioadă nedeterminată (activitatea pe site va fi reținută), schimbarea parolei, fie prin “Forgotten Password”, fie prin modificarea acesteia de la panoul de control al utilizatorului, unde va avea și datele furnizate la profil modificabile.



* 1. **Sistem de mesagerie**

Utilizatorii vor putea comunica prin mesaje, asemănător unei pagini de grup pe Facebook. 2 sau mai mulți utilizatori vor putea avea o pagină comună unde se vor putea posta mesaje.

În acest mod utilizatorii aplicației vor putea discuta în privat algoritmi sau orice alte subiecte. Aceste mesaje for fi șterse permanent atunci când ultimul utilizator prezent va dori ștergerea lor sau atunci când nici un utilizator nu mai este prezent în conversație. O persoană care folosește aplicația va primi notificări atunci când un nou mesaj va fi primit și o legătură către conversația respectivă.



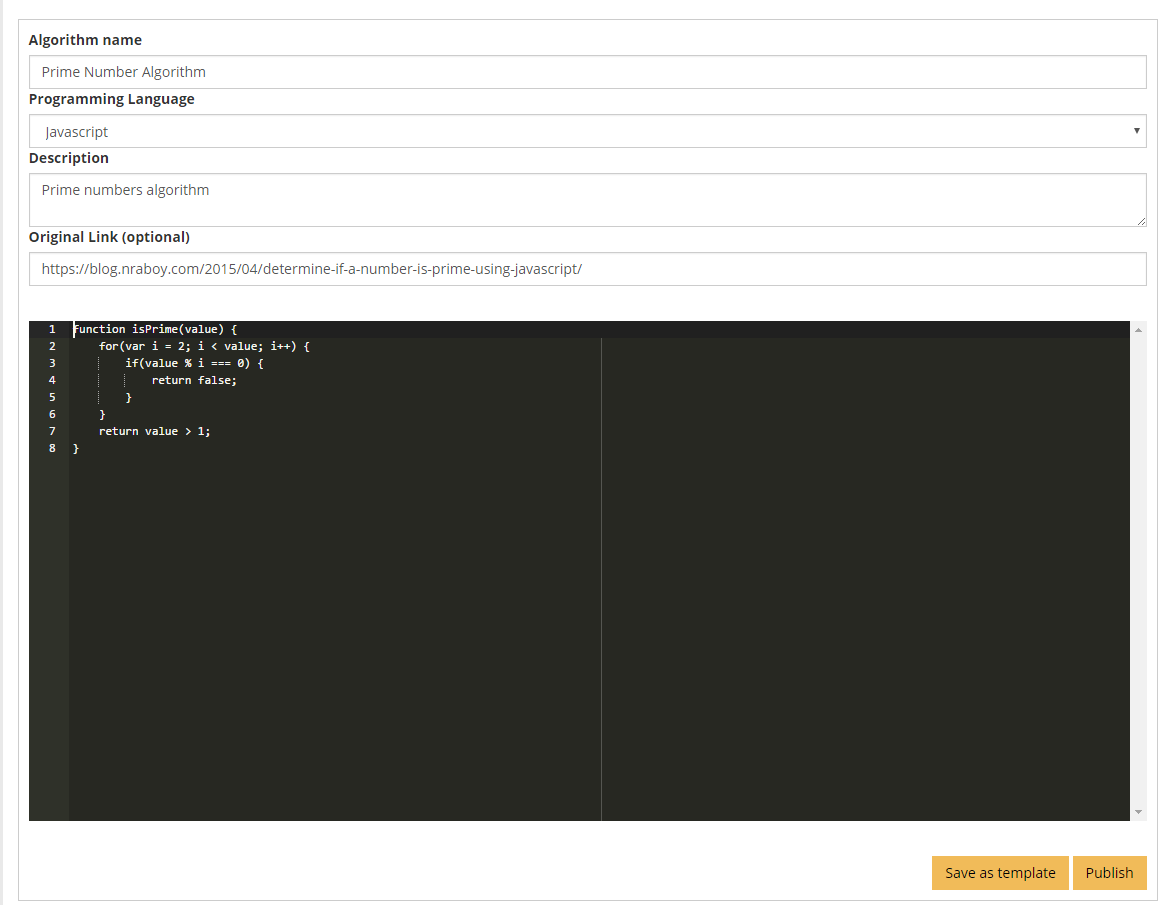
* 1. **Sistem de postare a algoritmilor**

Orice utilizator poate posta un articol care va conține implementarea unei probleme, precum și descrierea sa și adnotări la orice linie de cod la care este necesară această funcție.

Articolul trebuie încadrat într-o listă aparținând unui anumit limbaj de programare și va fi publicat doar atunci când creatorul permite asta. Altfel, acesta va fi păstrat ca template.

Dacă algoritmul este preluat de pe alt site se poate da un link către pagina originală. Un utilizator poate de asemenea șterge un algoritm postat de acesta, dar și de un moderator, dacă se observă nereguli precum scrierea a orice altceva decât un algoritm sau atunci când se primesc plângeri de tip copyright.

Bineînțeles, utilizatorii pot atrage atenția unui moderator atunci când un articol este văzut ca fiind incorect/ofensiv, urmând să fie luate măsuri în diferite moduri.

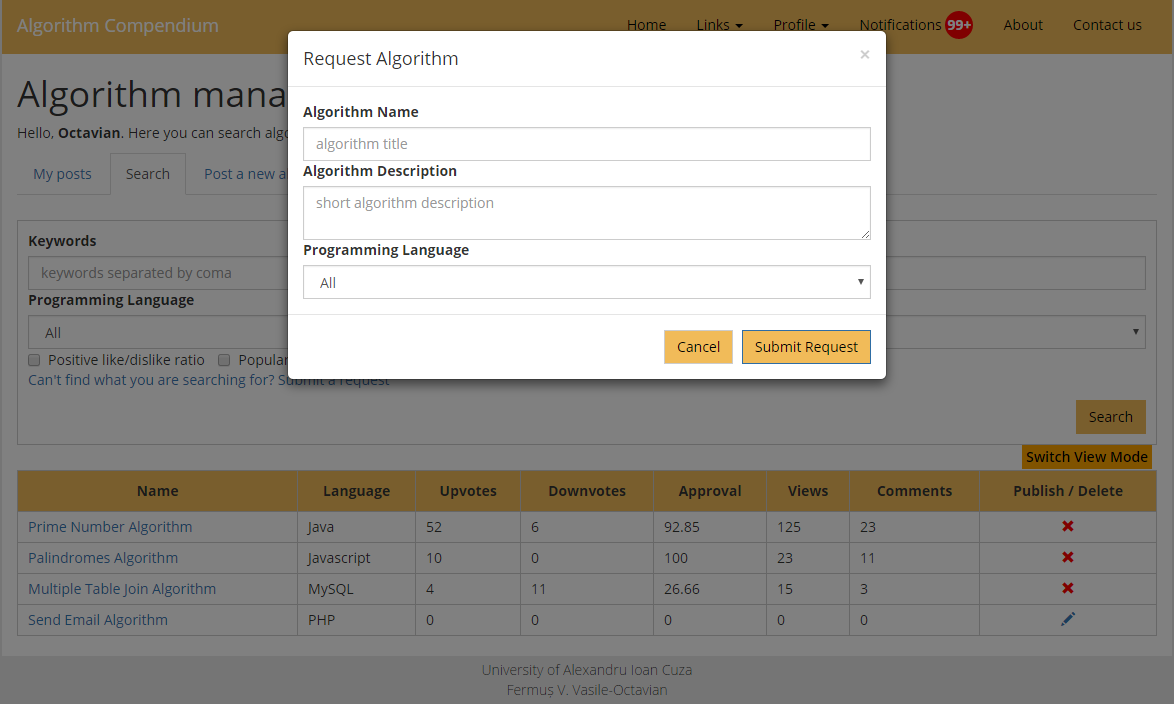


* 1. **Sistem de cereri de algoritmi**

Utilizatorii aplicației pot face cereri de algoritmi pe o pagină specializată pentru aceasta. Vor avea la dispoziție o listă cu cererile existente și pot vota acele cereri pentru a primi o atenție mai mare de la persoanele ce folosesc site-ul.

Similar, dacă nu există o astfel de cerere, se poate crea pe loc, adăugându-se în scurt timp la lista curentă, cu aprobarea administratorilor pentru a se evita cererile ambigue.

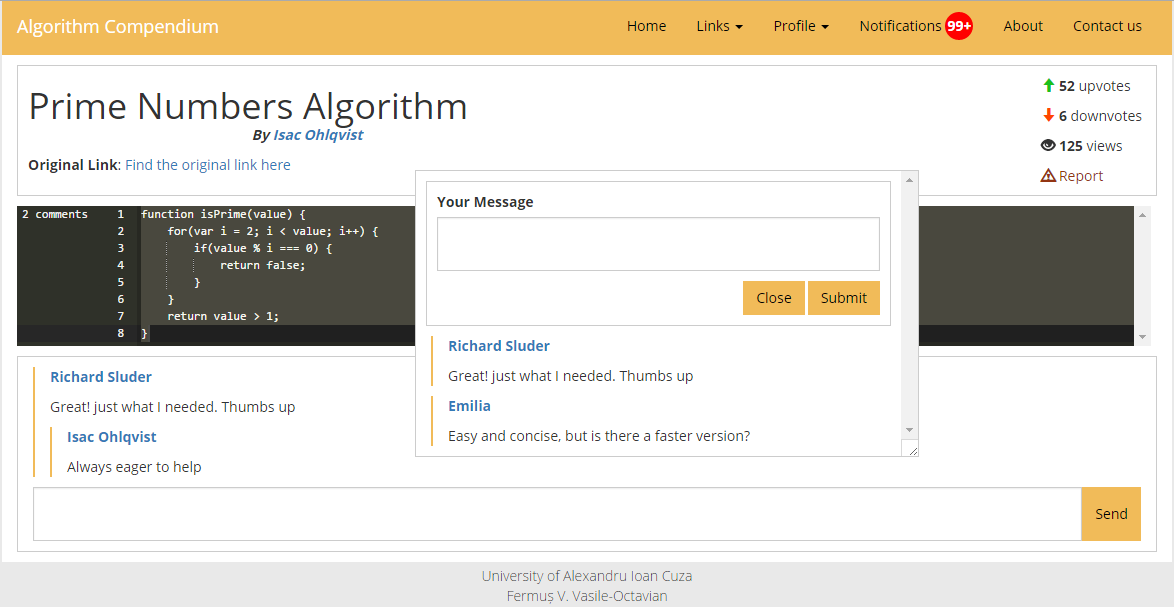
Utilizatorii pot vota negativ cereri; Dacă o cerere are prea multe voturi negative va fi marcată pentru vizualizare, luându-se măsuri dacă este nevoie.



* 1. **Sistem de comentare/notare/adnotare a codului**

Utilizatorii pot bineînțeles comenta asupra algoritmilor postați, asemănător unei pagini de blog. Mai mult decât atât, ei pot face postări de dimensiuni mici în dreptul unei linii de cod, aceasta fiind vizibilă prin popup. Cei care fac astfel de comentarii pot alege dacă acel mesaj este vizibil public sau doar celui ce a scris codul original.

Fiecare articol va putea fi notat cu upvote/downvote sau salvat într-o listă de algoritmi favoriți pentru ca orice utilizator să aibă acces facil la acesta. Dacă creatorul original al codului șterge algoritmul, fiecare utilizator ce a salvat acel algoritm va primi o notificare, precum și opțiunea de a șterge acel link, urmând ca după o perioadă de timp să fie șters automat din baza de date. Această notare va fi folosită și în sistemul de căutare a algoritmilor.

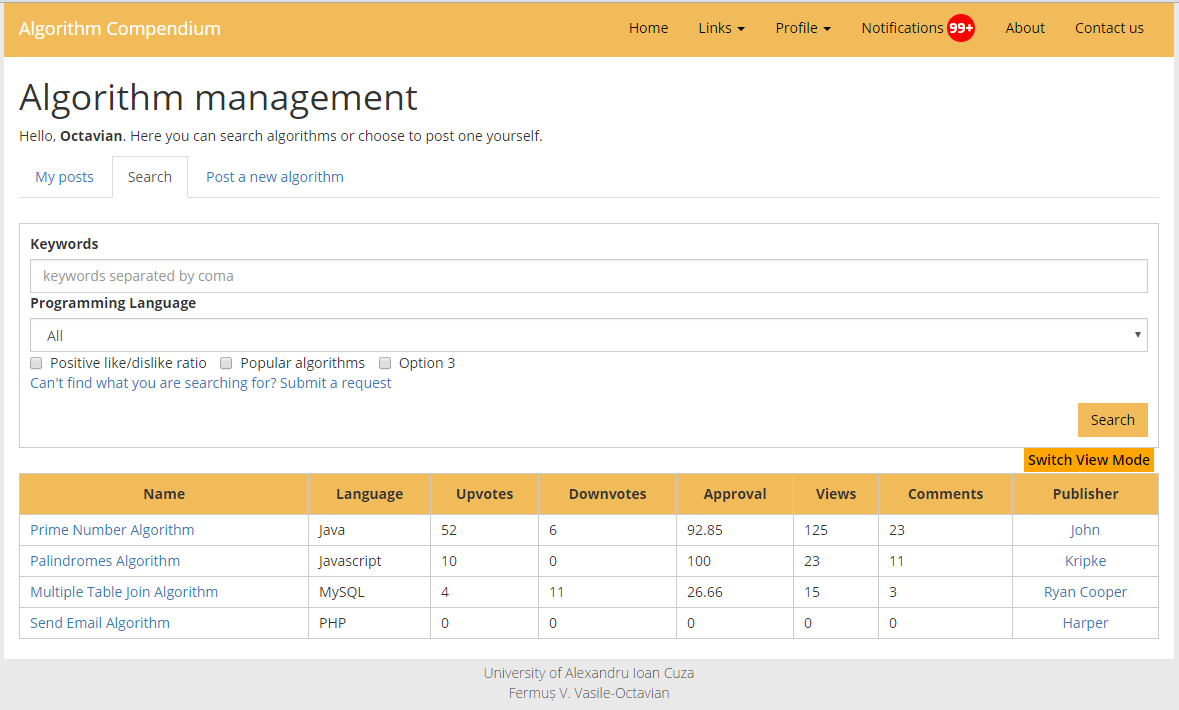


* 1. **Cautare dupa popularitate/nume/limbaj de programare**

Notarea despre care am scris anterior va fi folosit în pagina de căutare de algoritmi, cei cu o popularitate mai mare fiind mai ușor de găsit, iar cei cu note mici primind mai multă atenție.

De fiecare dată când un articol nou este postat, are o popularitate mai mare pentru o anumită perioadă scurtă de timp, pentru a putea fi văzut de utilizatori, după care va scădea/crește treptat, după notele date în timp.

Căutarea se va face pe bază de limbaj, numărul de utilizatori ce au vizualizat/notat pagina, precum și nota obținută de algoritm. Pentru a diferenția algoritmi cu același nume, numele utilizatorului ce a postat algoritmul va fi adăugată în listă, precum data postării și alte informații utile.



* 1. **Multi-language (engleză-română)**

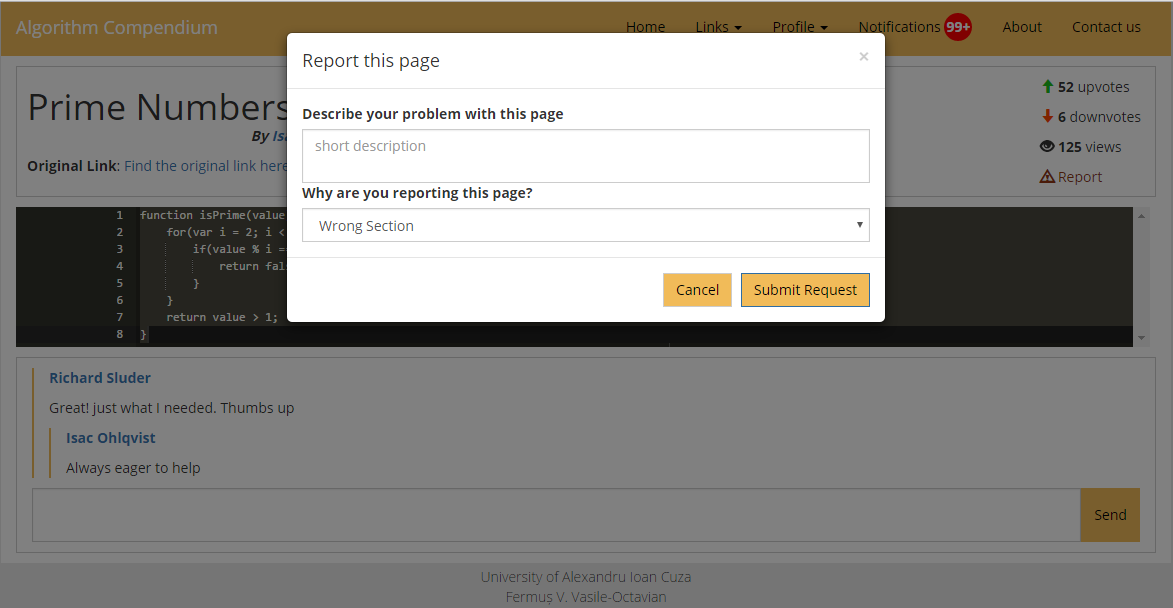
Pe plan secundar s-ar dori implementarea site-ului în engleză și română, fiind nevoie de support masiv din partea php pentru orice conținut ce nu tine de cel scris de un utilizator al site-ului. Stocarea textelor meniurilor, logo-ului, paginilor de logare, căutare sau postare a algoritmilor se va face în MySQL și transmisă paginii HTML la deschiderea paginii.

Folosind Javascript, schimbarea limbii se poate face dinamic, fără să fie nevoie de Refresh în cadrul paginii, însă această funcționalitate este una secundară, pornindu-se cu limba engleză ca limbă principală și translatarea făcându-se dacă timpul o permite.

* 1. **Sistem de report**

Orice neregulă sau conduită rea a utilizatorilor poate fi reportată printr-un simplu click de buton. Neregulile pot fi legate de numele unui utilizator, comentarii postate sau comportament neadecvat, precum și o postare de algoritm ce prezintă nereguli clar intenționate. Măsurile luate pot varia, de la atenționări, oprirea utilizatorului de la a posta până la banarea contului pe diferite perioade de timp.

Toate aceste măsuri vor fi publice pe pagina utilizatorului și algoritmii postați, alții putând vedea reputația sa atunci când caută un algoritm sau compară algoritmi a diferiți utilizatori. În acest mod se dorește menținerea unei ordini între cei ce folosesc site-ul.



1. Concluzii

Deși implementări similare există (stackOverflow, w3schools), ideea este puțin diferită de cea a aplicației pe care doresc să o creez pentru această lucrare, căutarea fiind gândită ca fiind mai facilă decât stackOverflow, și mai riguroasă decât pe w3schools.

De asemenea, se dorește eliminarea ambiguității, precum și suport mai bun pentru înțelegerea algoritmilor, cu ajutorul posibilei comunități. De asemenea, comunicarea între grupuri de utilizatori este oferită ca posibilitate, oferind șansa de “cursuri” sau discuții asupra diferitelor teme prezentate.