

UNIVERSITATEA „ALEXANDRU IOAN CUZA” DIN IAȘI
FACULTATEA DE INFORMATICĂ



LUCRARE DE LICENȚĂ

Coordonator științific

Lector Dr. Cosmin Vârlan

Absolvent

Soficu Corneliu-Dumitru

Sesiunea: Iulie 2018

UNIVERSITATEA „ALEXANDRU IOAN CUZA” DIN IAȘI
FACULTATEA DE INFORMATICĂ

APLICAȚIA DE EVALUARE A
PROFESORILOR DIN
FACULTATEA DE
INFORMATICĂ

Coordonator științific

Lector Dr. Cosmin Vârlan

Absolvent

Soficu Corneliu-Dumitru

Sesiunea: Iulie 2018

Avizat,
Îndrumător Lucrare de Licență

Titlul, Numele și prenumele _____

Data _____ Semnătura _____

**DECLARAȚIE privind originalitatea conținutului lucrării de
licență/diplomă/disertație/absolvire**

Subsemnatul(a)

domiciliul în

născut(ă) la data de, identificat prin CNP

absolvent(a) al(a) Universității „Alexandru Ioan Cuza” din Iași, Facultatea de

specializarea, promoția

declar pe propria răspundere, cunoscând consecințele falsului în declarații în sensul art. 326 din Noul Cod Penal și dispozițiile Legii Educației Naționale nr. 1/2011 art.143 al. 4 și 5 referitoare la plagiat, că lucrarea de licență cu titlul: _____

elaborată sub îndrumarea dl. / d-na _____,

pe care urmează să o susțină în fața comisiei este originală, îmi aparține și îmi asum conținutul său în întregime.

De asemenea, declar că sunt de acord ca lucrarea mea de licență/diplomă/disertație/absolvire să fie verificată prin orice modalitate legală pentru confirmarea originalității, consimțind inclusiv la introducerea conținutului său într-o bază de date în acest scop.

Am luat la cunoștință despre faptul că este interzisă comercializarea de lucrări științifice în vederea facilitării falsificării de către cumpărător a calității de autor al unei lucrări de licență, de diploma sau de disertație și în acest sens, declar pe proprie răspundere că lucrarea de față nu a fost copiată ci reprezintă rodul cercetării pe care am întreprins-o.

Data azi,

Semnătură student

DECLARAȚIE DE CONSIMȚĂMÂNT

Prin prezenta declar că sunt de acord ca Lucrarea de licență cu titlul „Aplicația de evaluare a profesorilor din facultatea de informatică”, codul sursă al programelor și celelalte conținuturi (grafice, multimedia, date de test etc.) care însoțesc această lucrare să fie utilizate în cadrul Facultății de Informatică.

De asemenea, sunt de acord ca Facultatea de Informatică de la Universitatea Alexandru Ioan Cuza Iași să utilizeze, modifice, reproducă și să distribuie în scopuri necomerciale programele-calculator, format executabil și sursă, realizate de mine în cadrul prezentei lucrări de licență.

Iași, 28.06.2018

Absolvent Corneliu-Dumitru Soficu,

Cuprins

| | |
|--|----|
| Introducere..... | 6 |
| 1 Tehnologii Folosite | 10 |
| 1.1 Limbajul PHP..... | 10 |
| 1.2 Framework-ul de dezvoltare web CodeIgniter | 11 |
| 1.3 Baza de date MySQL | 16 |
| 1.4 Tehnologii Front-end | 22 |
| 2 Prezentarea aplicației | 26 |
| 2.1 Modulul pentru studenți..... | 27 |
| 2.2 Modulul pentru administratori | 31 |
| 2.3 Modulul pentru profesori | 36 |
| 2.4 Modulul pentru decan | 36 |
| 3 Securitatea aplicației | 37 |
| 3.1 Autentificare..... | 37 |
| 3.2 Autorizarea accesului..... | 38 |
| Concluzii finale..... | 41 |
| Concluzii..... | 41 |
| Contribuții personale | 41 |
| Direcții de viitor..... | 41 |
| Bibliografie | 43 |

Introducere

În general, evaluarea profesorilor se referă la un proces formal folosit de o instituție de învățământ pentru a măsura performanța și eficacitatea profesorilor în actul de predare. În România acest proces este reglementat prin lege astfel încât rezultatele și performanțele activităților didactice și de cercetare ale personalului didactic și de cercetare dintr-o universitate sunt evaluate periodic, la intervale de maximum 5 ani. Această evaluare se face în conformitate cu o metodologie aprobată și aplicată de către senatul universitar (1).

Un sistem de evaluare din cadrul unei universități are ca rol principal îmbunătățirea calităților fiecărui profesor prin determinarea punctelor slabe și prin identificarea diferitelor oportunități de dezvoltare. Într-un sondaj internațional organizat de OECD¹ privind predarea și învățarea, profesorii au răspuns că *“Evaluările de performanță și feedbackul le crește satisfacția la locul de muncă și le crește semnificativ dezvoltarea lor ca profesori”* (2).

Scopul final al unui sistem de evaluare este de a îmbunătăți permanent actul didactic astfel încât studenții ce trec pragul universității să devină din ce în ce mai pregătiți pentru câmpul muncii. Exista mai multe modalități prin care activitatea didactica a unui profesor poate fi evaluată:

- Observații în cadrul unei ore de curs / seminar efectuate de către un alt profesor, de către o persoană din conducere sau de către o persoană externă;
- Auto-evaluare;
- Evaluări efectuate de către studenți.

Una din cele mai bune metode de evaluare a unui profesor este reprezentată de feedbackul oferit de către studenți. Aceștia au capacitatea de a evalua imparțial și

¹ The Organisation for Economic Co-operation and Development

anonim activitatea profesorului pe parcursul unui an universitar, oferind amănunte despre modul în care profesorul își desfășoară orele de curs / seminar. În acest mod studenții contribuie efectiv la îmbunătățirea condițiilor educaționale exprimându-și indignarea sau entuziasmul la sfârșitul unui semestru sau an universitar.

Astfel, se impune implementarea unui sistem de evaluare a profesorilor de către studenți și în cadrul Facultății de Informatică din Iași întocmai pentru a îmbunătăți condițiile educaționale conform argumentelor prezentate anterior.

Cu siguranță o modalitate de implementare este reprezentată de clasicul model al chestionarului, ce este distribuit studenților în format fizic, pe foi printate. Această modalitate aduce o serie de probleme ce pot influența rezultatul unei evaluări. O primă problemă ar fi cauzată de logistica organizării. Considerând numărul mare de studenți ce urmează o facultate, ar fi nevoie de o cantitate amplă de hârtie pentru a putea distribui fiecărui student câte o copie. Se mai poate adăuga și că timpul petrecut pentru a citi, a interpreta și a stoca rezultatele este foarte îndelungat.

Din fericire progresul tehnologic a cunoscut o uriașă creștere în ultimele decenii (3), după cum se poate vedea în Fig. 1, astfel încât tot mai multe persoane au

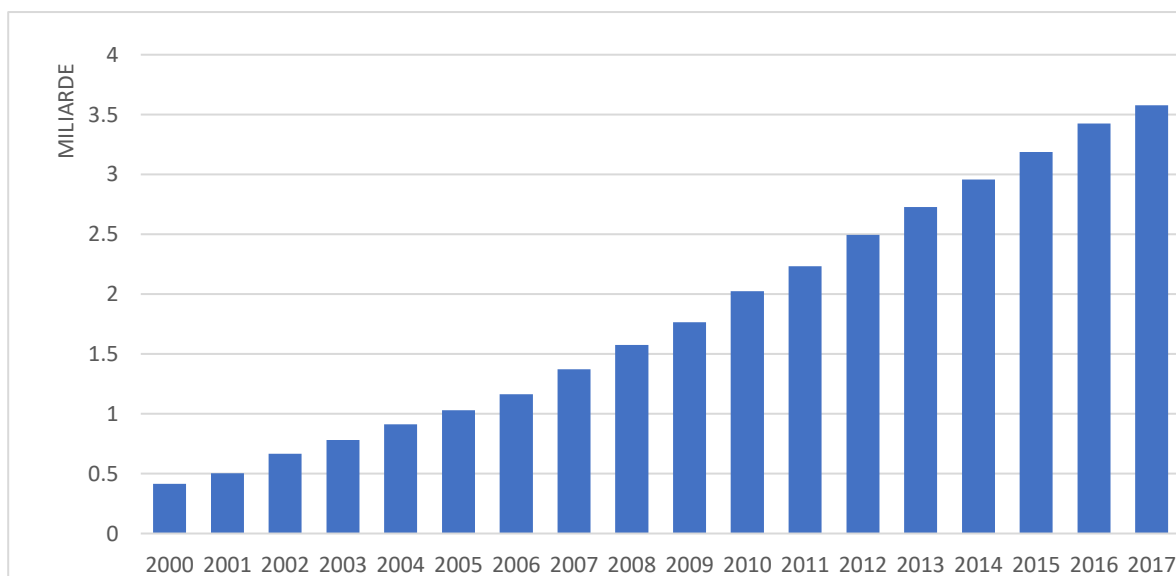


Fig. 1 Evoluția numărului de utilizatori ai internetului în ultimii 18 ani.

acces la internet. Totodată, îmbunătățirea vitezei de calcul și scăderea prețurilor pentru

dispozitivele capabile să se conecteze la internet, creează mediul perfect în care se poate dezvolta o aplicație prin care studenții să-și poată evalua profesorii de la facultate. O astfel de aplicație poate oferi anonimitate completă studenților, încurajându-i să-și expună, fără teamă și neinfluențați de nimeni, părerile. Aplicația poate administra într-un mod eficient și organizat într-o bază de date toate rezultatele evaluărilor, astfel încât să fie folosite pentru îmbunătățirea calității actului didactic.

Facultatea de Informatică dispunea până în acest an de o aplicație proprie prin care studenții să-și evalueze profesorii. Aceasta avea o funcționalitate de bază, fiind formată doar dintr-o simplă pagină cu întrebări. După ce studenții răspundeau la acele întrebări, datele erau stocate într-o bază de date. Luând în considerare migrarea utilizatorilor de internet dinspre calculatoare spre dispozitive mobile, fapt evidențiat și în figura Fig. 2, unde se observă că din anul 2017, majoritatea utilizatorilor de internet, folosesc telefonul mobil în detrimentul calculatorului, devine necesară îmbunătățirea aplicației de evaluare a profesorilor pentru acest tip de dispozitive.

Un alt neajuns ce poate fi identificat în cadrul aplicației curente de evaluare

este reprezentat de dificultatea accesării rezultatelor. Ar fi de mare folos ca profesorii evaluați să poată vedea ce cred studenții despre modul în care își susțin cursurile sau seminariile. Totodată, accesarea tuturor rezultatelor într-un mod rapid și sigur de către persoane cu funcție de conducere poate determina mai ușor luarea unor decizii privind

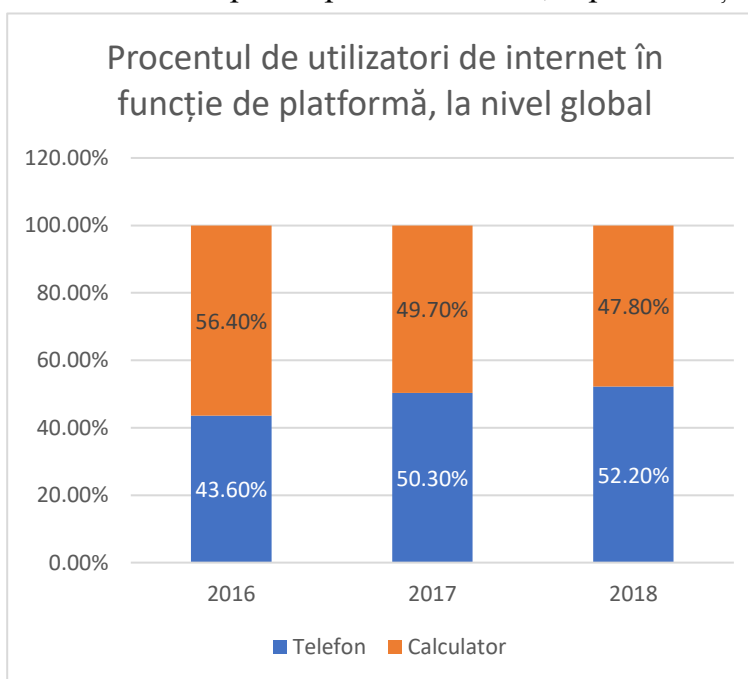


Fig. 2 Din anul 2017, majoritatea utilizatorilor folosesc telefonul mobil pentru a accesa internetul

activitatea anumitor profesori. De exemplu, decanul facultății ar putea ști foarte simplu, accesându-și contul în cadrul aplicației, care au fost profesorii cu cea mai bună evoluție sau profesorii care nu s-au descurcat atât de bine în timpul unui semestru.

Influențat de cele prezentate anterior, îmi propun prin intermediul acestei lucrări, implementarea unei astfel de aplicații pentru evaluarea profesorilor. Pornind de la modelul aplicației curente de evaluare, noua aplicație este o aplicație web realizată în PHP folosind framework-ul web CodeIgniter, având o bază de date MySQL și este axată pe un design prietenos cu utilizatorul, putând fi cu ușurință folosită și pe platforme mobile. Aplicația este disponibilă pentru studenți, pentru profesori, pentru decan și este întreținută de către un grup de administratori.

În capitolul 1 sunt prezentate tehnologiile și conceptele utilizate pentru dezvoltarea acestei aplicații web.

Capitolul 2 descrie funcționalitățile și modalitățile de utilizare ale aplicației în funcție de tipul utilizatorilor.

Capitolul 3 prezintă modalitățile implementate pentru a asigura securitatea în aplicația de evaluare a profesorilor.

1 Tehnologii Folosite

1.1 Limbajul PHP

În acest moment, PHP are cea mai mare cotă de piață (Fig. 3) dintre toate limbajele de tip Backend, fiind folosit de peste 80% dintre toate site-urile web de pe internet (4). Unele dintre cele mai populare site-uri web ce folosesc PHP sunt facebook.com ce a înregistrează peste 1.45 miliarde de utilizatori activi într-o singură zi (5) sau wikipedia.org cu un număr mediu de 18 miliarde de articole citite de utilizatori pe lună (6). Putând fi rulat pe toate tipurile de sisteme operare majore, PHP asigură o foarte mare flexibilitate iar orientarea limbajului către programarea orientată obiect face dezvoltarea

aplicațiilor mult mai ușoară. Ca atare limbajul PHP, apărut în anul 1994 rămâne alegerea preferată a dezvoltatorilor web când vine vorba de implementarea părții de backend al unui site web, dovedindu-și ani la rând eficiența.

Întrucât server-ul Facultății de Informatică este capabil să ruleze scripturi PHP, aplicația de evaluare a profesorilor poate fi dezvoltată și apoi lansată pe server-ul facultății, putând rula mulți ani de acum încolo. Versiunea de PHP disponibilă pe serverul „profs” al Facultății de Informatică este 7.2, versiune ce va primi actualizări de securitate și îmbunătățiri până în anul 2021 (7).

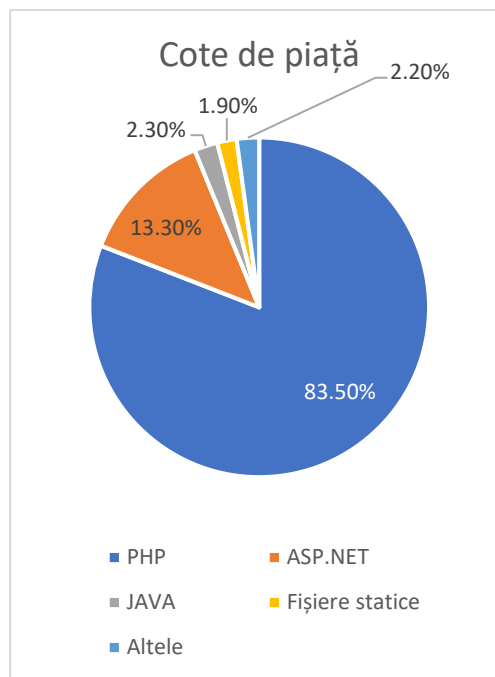


Fig. 3 Cotele de piață ale fiecărui limbaj folosit pe parte de backend

1.2 Framework-ul de dezvoltare web CodeIgniter

Folosindu-se de limbajul de scripting PHP, framework-ul web CodeIgniter este open-source și este utilizat pentru dezvoltarea de site-uri web dinamice. Fiind administrat de *British Columbia Institute of Technology* și ajuns deja la a treia versiune, CodeIgniter este bazat pe șablonul de dezvoltare model-view-controller(MVC), putând fi modificat pentru a folosi șablonul hierarchical-model-view-controller(HMVC).

Crescând în popularitate în ultimii ani, Rasmus Lerdorf, creatorul limbajului PHP a declarat că i-a plăcut acest framework „*întrucât este mai rapid, mai ușor și cel mai puțin asemănător unui framework*” (8).

În general CodeIgniter este folosit pentru dezvoltarea rapidă de software, având un impact redus asupra memoriei și având o dimensiune mică. Aceste argumente fac framework-ul să aibă o performanță foarte bună și să se claseze în top-ul framework-urilor dezvoltate în PHP. Conform unei serii de teste realizate pe cele mai folosite framework-uri de PHP, al căror rezultat a fost publicat pe platforma Medium de către Will Bowman (9), CodeIgniter este framework-ul cu cea mai bună performanță. Testele au constatat în afișarea unui mesaj simplu „Hello World”, folosind setările minimale ale fiecărui framework: fără bază de date, fără debugging și fără un sistem de templating. Mediul folosit pentru rularea testelor a fost format dintr-o versiune PHP 7.0.7, utilizând Apache 2.4.18 și sistemul de operare Ubuntu 16.04 x64 2gb.

Rezultatele obținute în urma testelor efectuate pe cele mai populare framework-uri PHP:

| Framework | Număr de cereri pe secundă | Memorie maximă(MB) |
|------------------|-----------------------------------|---------------------------|
| PHP simplu | 5241 | 0,34 |
| CodeIgniter | 1696 | 0,38 |
| Symfony | 330 | 2,13 |
| Laravel | 180 | 2,29 |
| Zend Framework | 273 | 1,77 |

Așadar, analizând rezultatele, se poate observa că framework-ul cu cea mai bună performanță în funcție de numărul de cereri pe secundă și în funcție de impactul asupra memoriei este reprezentat de CodeIgniter.

De menționat și că acest framework are nevoie de configurări minimale pentru a putea funcționa, astfel încât lansarea acestuia pe orice server web ce poate rula scripturi PHP se poate face fără probleme, într-un timp scurt.

Luând în calcul toate aceste aspecte pozitive legate de CodeIgniter, putem regăsi o multitudine de companii ce au adoptat CodeIgniter ca soluție pentru realizarea site-urilor lor. Printre cele mai populare regăsim:

- Casio Computers: Este o companie multinațională specializată în producția și vânzarea de electronice fondată în Japonia, care a folosit CodeIgniter pentru dezvoltarea website-ului de prezentare a produselor lor.
- Nissan: Gigantul nipon din industria auto a recurs la CodeIgniter pentru site-ul lor de prezentare și vânzare de mașini (Fig. 4);

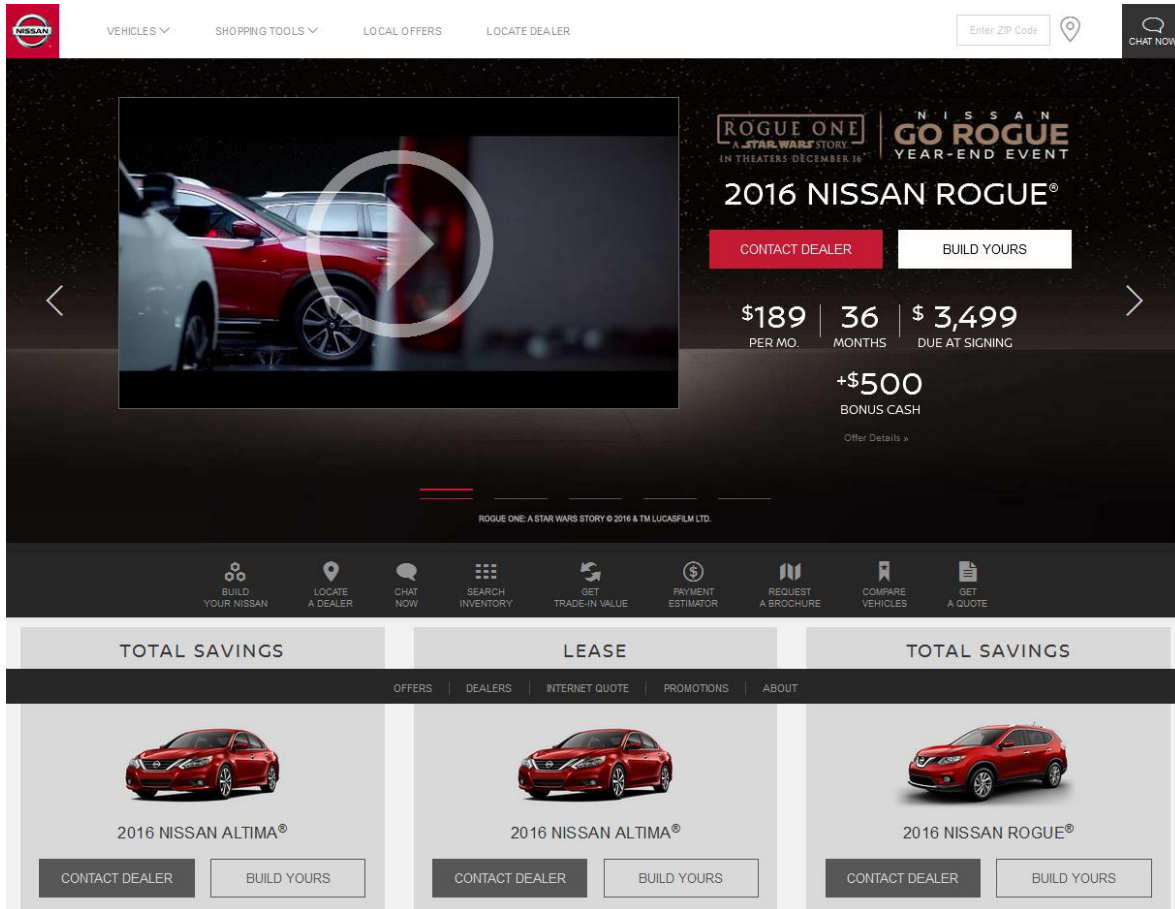


Fig. 4 Website-ul pentru prezentarea și comercializarea de automobile al companiei Nissan a fost realizat în CodeIgniter

Cu toate acestea, CodeIgniter prezintă și o serie de dezavantaje, precum:

- Lipsa unui ORM: Framework-ul în sine nu conține această calitate, motiv pentru care, multe interogări în baza de date trebuie scrise manual.
- Anumite părți nu sunt orientate obiect, cauzând o oarecare lipsă de lizibilitate și înțelegere a codului.
- Uneori trebuie dezvoltate manual unele componente de bază, întrucât nu sunt disponibile în mod implicit, de exemplu: modulul de autorizare al accesului.
- Lipsa unui nivel de tip Business Logic în cadrul aplicației

La fel ca oricare alt framework popular pentru dezvoltarea aplicațiilor web, CodeIgniter separă logica aplicației în modele, view-uri și controllere:

- Modelul reprezintă structura de date folosită pentru a actualiza informații din baza de date, prin efectuarea operațiilor de tip CRUD.
- View-ul în CodeIgniter este puțin diferit față de alte framework-uri similare de PHP, întrucât nu vine cu un sistem de templateing precum Blade, Mustache sau Volt. Acest fapt se datorează concepției minimaliste orientată spre performanță, întrucât un motor de templateizare penalizează la capitolul eficiență prin nevoia de interpretare a sintaxei speciale în sintaxă PHP.
- Controller-ul are ca scop legarea modelului de view, utilizând cât mai puțină logică.

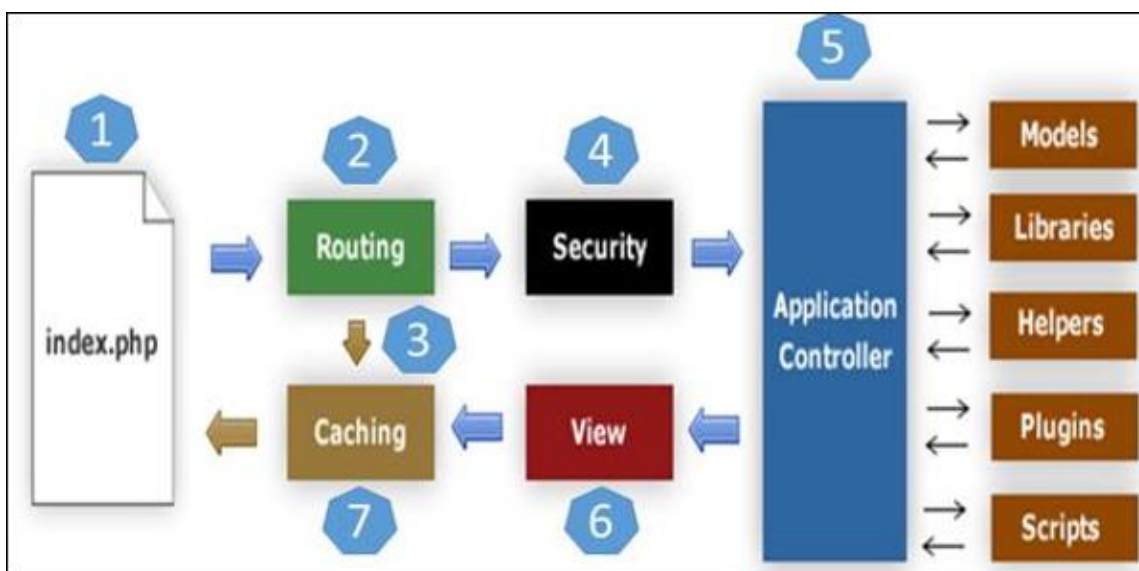


Fig. 5 Arhitectura unei aplicații CodeIgniter

Deși, CodeIgniter este împărțit pe mai multe niveluri, conform unei arhitecturi de tip „multitier”, acestuia îi lipsește nivelul de Business Logic, unde de obicei toată logica unei aplicații este implementată. Totuși, CodeIgniter pune la dispoziție varianta creării de helpere și librării ce pot fi folosite în cadrul controllerelor. Această caracteristică este reprezentată și în Fig. 5 (10).

În cadrul aplicației de evaluare a profesorilor din Facultatea de Informatică, am adoptat o extensie a MVC în cadrul CodeIgniter numită hierarchical-model-view-controller (HMVC). Acest șablon de dezvoltare permite organizarea aplicației sub forma unor module, fiecare modul fiind la rândul lui organizat în MVC. Cel mai mare beneficiu practic al acestei organizări este reprezentat de separarea structurii de conținut într-un fel de widget-uri. Un exemplu ar putea fi comentariile, rating-urile sau afișarea conținutului unui coș de cumpărături pentru un website de comerț electronic. Acestea reprezintă în esență o bucată din conținut ce trebuie afișată pe mai multe pagini sau chiar, în alte locuri, în funcție de contextul cererii principale HTTP.

Framework-urile obișnuite MVC în general nu oferă o soluție clară pentru acest tip de structură, așa că programele ajung deseori să duplicate și să schimbe layout-uri folosind helpere speciale, creându-și propriile lor structuri de widget-uri sau librării. Neajunsul este că responsabilitatea generării unei bucăți particulare de conținut sau încărcarea datelor necesare se „scurge” în zone multiple sau se duplică.

HMVC, are ca scop rezolvarea acestei probleme. Structura este similară cu cea tradițională MVC, de exemplu: dacă este nevoie de a încărca date despre comentarii și a le afișa în format HTML, s-ar trimite o cerere către un controller de comentarii cu anumiți parametri. Cererea interacționează cu modelul, încarcă view-ul și afișează conținutul. Diferența față de tradiționalul MVC este că în loc de a afișa comentariile într-o pagină separată, ele sunt afișate în linie, sub articolul pe care îl citește utilizatorul. Din cauza aceasta, HMVC reușește să crească modularitatea codului, ajută la reutilizare și menține o mai bună separare a preocupărilor (*eng.* separation of concerns) (11). Acest lucru este evidențiat în figura Fig. 6.

În cadrul aplicației de evaluare a profesorilor, am implementat 9 module, fiecare având propriile controllere, modele și view-uri, acestea fiind pentru: evaluări, profesori, setări, statistici, înregistrare, date, plângeri, rapoarte și pentru generarea de perioade de evaluare.

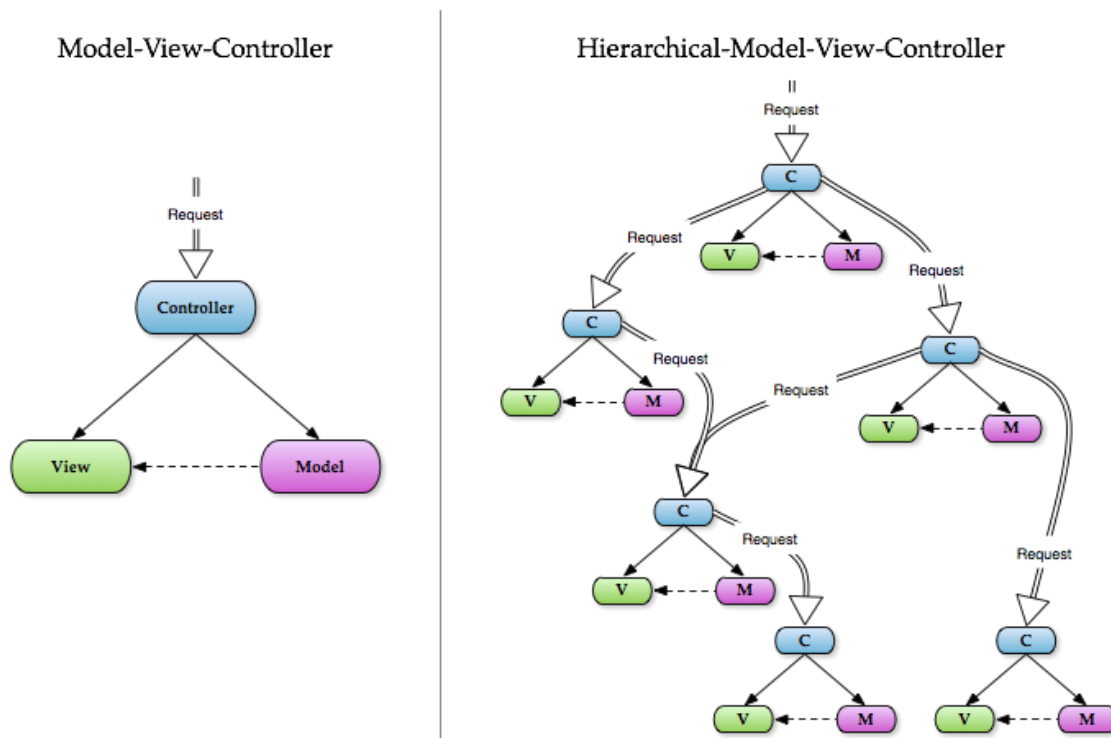


Fig. 6 Comparație între model-view-controller și hierarchical-model-view-controller

Considerând tot ce a fost menționat mai sus, pot spune că framework-ul CodeIgniter se pretează cu succes la cerințele impuse de implementarea aplicației de evaluare a profesorilor din Facultatea de Informatică Iași, iar organizarea acesteia după șablonul HMVC aduce beneficii în timpul dezvoltării dar și în viitor, asigurând o mai bună claritate asupra codului și o mentenanță mai ușoară.

1.3 Baza de date MySQL

Motivul folosirii acestei baze de date în cadrul aplicației de evaluare a profesorilor este reprezentat de succesul și fiabilitatea dovedită de la lansarea acesteia în 1995 până în zilele noastre. Fiind folosită de către agenții guvernamentale precum NASA, de către corporații ca Facebook, Netflix sau Spotify, MySQL reprezintă o soluție foarte bună pentru orice tip de aplicație ce are nevoie de o bază de date relațională. Luând în considerare și faptul că Facultatea de Informatică a adoptat MySQL ca și bază de date

proprie, pot spune că cea mai bună variantă pentru baza de date în cadrul aplicației de evaluare a profesorilor este MySQL.

Pentru realizarea acestei aplicații am recurs la o structura tabelara formată din 14 tabele pentru a deservi diferitelor informații ce trebuie stocate. Anumite tabele au asociate chei străine ce sunt folosite pentru a constrânge consistența datelor.

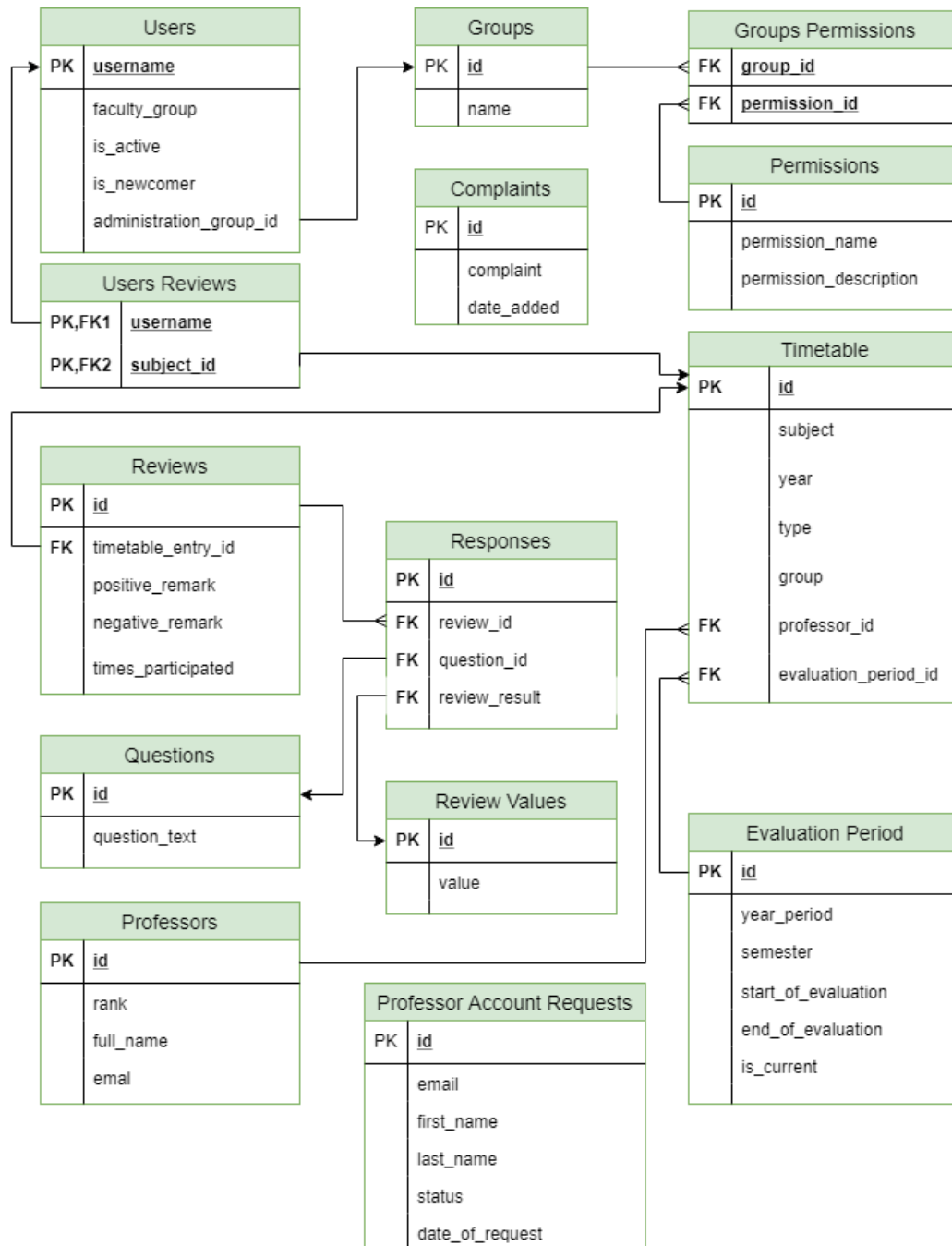


Fig. 7 Diagrama Entity-Relationship a bazei de date din aplicația de evaluare a profesorilor

Din Fig. 7 se poate observa că nu există nici-o asociere între un student și evaluarea pe care acesta a făcut-o, astfel fiind asigurată confidențialitatea evaluărilor. Există doar o asociere între un student și o materie, în tabela „Users Reviews”, aceasta fiind făcută pentru a preveni un student să evalueze aceeași materie de mai multe ori.

Pentru fiecare cheie străină din baza de date am optat să restricționez acțiunea de ștergere. Am ales această variantă pentru a mă asigura că dacă cineva încearcă să elimine o intrare dintr-o tabela, toate celelalte intrări asociate prin cheie străină acesteia, să nu fie șterse. În schimb am adoptat o strategie de „soft delete” pentru a păstra datele în baza de date dar a nu le afișa în interfață.

Tabelele din baza de date au următoarele atribuții:

- *Evaluation Period*: Stochează date privind perioadele de evaluare ce s-au desfășurat s-au se desfășoară în facultate. Se rețin anul universitar, semestrul, data de început și data de sfârșit ale fiecărei perioade de evaluare. Câmpul *is_current* este utilizat pentru a specifica care este perioada de evaluare în curs de desfășurare. Poate exista o singură perioadă activă, constrângere impusă de un trigger pe *on insert* și *on update*. Dacă există o intrare cu valoarea câmpului *is_current* 1 (TRUE), atunci aplicația va fi deschisă studenților pentru evaluare.
- *Timetable*: Reprezintă un tabel în care se stochează toate materiile din orar. Datele din acest tabel sunt vizualizate de către studenți pentru a efectua evaluările. Fiecare intrare din acest tabel are asociat un id al unei perioade de evaluare din tabelul *Evaluation Period*.
- *Professors*: Este o tabelă în care se stochează numele și gradul didactic al profesorilor găsiți în orar. În plus, este folosită pentru a stoca datele de autentificare ale profesorilor, când doresc să-și vadă rezultatele.
- *Reviews*: Este folosită pentru a stoca evaluările făcute de studenți. Fiecare intrare are un id asociat unei intrări în tabela *Timetable* pentru a identifica pentru ce s-a făcut evaluarea. Câmpurile *positive_review* și *negative_review* specifică

comentariile pozitive și negative ale profesorului evaluat iar `times_participated` specifică numărul de seminarii/cursuri la care a participat studentul.

- *Responses*: Este folosită pentru a stoca răspunsurile studenților la întrebările din tabela *Questions*. Această tabelă este în relație de one-to-many cu tabela *Reviews*, astfel încât o evaluare are asociată mai multe răspunsuri.
- *Questions*: Specifică întrebările la care trebuie să răspundă studenții.
- *Review Values*: Conține răspunsurile posibile la întrebările disponibile. Este de preferat ca aceste valori să nu fie schimbate, pentru că ele sunt folosite pentru afișarea statisticilor precum media pe mai mulți ani.
- *Users*: Reprezintă tabela studenților, unde se stochează numele de cont de pe server-ul fenrir, grupa din care fac parte studenții, grupă folosită pentru a identifica materiile ce trebuie revizuite de către aceștia. Prin intermediul câmpului `is_newcomer` se verifică dacă studentul are selectată o grupă, astfel încât să fie forțat să-și selecteze una, înainte să înceapă să evalueze.
- *Users Reviews*: Reprezintă un istoric cu materiile revizuite de către utilizatori. Este folosită pentru a identifica dacă un utilizator a revizuit sau nu o materie. Nu se stochează evaluările efectuate de către aceștia, fiind păstrată astfel anonimitatea.
- *Groups*: În aplicația de evaluare a profesorilor sunt definite mai multe grupuri administrative iar tabela *Groups* conține informații despre fiecare. Este de preferat ca id-urile acestor grupuri să nu fie modificate întrucât sunt folosite în interiorul aplicației.
- *Permissions*: Existând mai multe tipuri de utilizatori, fiecare poate efectua doar anumite acțiuni. Aceste acțiuni sunt definite de permisiuni și pot fi asociate mai multor tipuri de utilizatori.
- *Groups Permissions*: Este o tabelă intermediară ce rezolvă relația many-to-many dintre grupuri și permisiuni, astfel încât fiecare grup poate avea mai multe permisiuni, iar fiecare permisiune poate fi asociată la mai multe grupuri.

- *Complaints*: Întrucât această aplicație are implementată o funcționalitate prin care utilizatorii să declare anumite inconveniențe întâmpinate în folosire, este nevoie de o tabelă în care să fie stocate aceste plângeri.
- *Professor Account Requests*: Reprezintă o tabelă în care sunt stocate cererile pentru conturi de profesor ale utilizatorilor. O astfel de cerere poate avea mai multe stări: În așteptare, Aprobata sau Respinsa. Un administrator trebuie să verifice cererile primite în aplicație și să le valideze dacă sunt valide.

Deși se observă că există un număr destul de mare de tabele, acest lucru nu ajunge să afecteze performanța prin creșterea timpului interogărilor. Folosind aplicația de evaluare a profesorilor, se fac maxim 3 join-uri, dar acestea se fac prin intermediul câmpurilor de tip cheie străină care au un index creat automat. Astfel join-urile se execută cu cea mai mare rapiditate. Exemplu de interogare cu 2 join-uri: Obținerea tuturor evaluărilor care sunt efectuate în timpul unei perioade de evaluare active. Pornind de la Evaluările din tabelă *Reviews*, se face un join cu tabelă *Timetable*, pentru a obține informațiile despre materiile evaluare și apoi un join cu tabelă *Evaluation Period*, pentru a verifica care este perioada de evaluare activă.

În general, interogările sunt destul de clare și de ușoare dar există și interogări mai complicate, de exemplu: pentru obținerea tuturor materiilor pe care un student trebuie să le evalueze împreună cu cele deja evaluate din perioada activă. Pentru realizarea acestei interogări, am folosit o sub-interogare, care, pentru fiecare curs și seminar din anul și grupa studentului, verifică dacă nu se găsește în tabelă de evaluări efectuate de către student. Un exemplu de interogare pentru a obține toate materiile evaluate și neevaluate de către studentul cu contul „corneliu.soficu”, aflat în grupa 3A5, poate fi observat în Fig. 8.

```

SELECT *,
    (SELECT IF(Count(subject_id) > 0, 1, NULL)
     FROM   user_reviews a
           JOIN timetable b
             ON b.id_subject = a.subject_id
           JOIN evaluation_period c
             ON c.id = b.evaluation_period_id
     WHERE  a.username = 'corneliu.soficu'
           AND c.is_current = 1
           AND b.subject = `timetable`.subject
           AND b.type = `timetable`.type
           AND b.professor_id = `timetable`.professor_id) AS
already_reviewed
FROM   `timetable`
JOIN   `professors`
      ON `timetable`.`professor_id` = `professors`.`id_professor`
JOIN   `evaluation_period`
      ON `timetable`.`evaluation_period_id` = `evaluation_period`.`id`
WHERE  `is_current` = 1
      AND `group` IN( 'I3', 'I3A', 'I3A5' )
ORDER BY `subject` ASC,
         `type` ASC,
         `full_name` ASC

```

Fig. 8 Interogare pentru a obține toate materiile pe care studentul le-a evaluat deja, împreună cu materiile neevaluate

Pentru a verifica dacă un student a evaluat deja o materie, trebuie găsite acele materii din tabela *Users Review* care au același nume, același profesor și același tip: Curs, Seminar, Laborator, etc.. Dacă există astfel de materii, ele vor fi marcate cu 1 și dacă nu, ele vor fi marcate cu NULL, conform sub-interogării. În final vor rezulta toate materiile disponibile pentru evaluare în funcție de grupa din care face parte studentul, marcate cu 1 dacă au fost deja evaluate și cu NULL în caz contrar. Întrucât, se face o căutare după 3 câmpuri, am creat un index peste coloanele: *subject*, *type* și *professor_id* pentru a eficientiza interogarea.

Pentru a veni în ajutorul dezvoltării de interogări în baza de date, framework-ul CodeIgniter oferă o librărie numită *QueryBuilder* pentru a crea mai sigur și mai ușor interogările. Librăria *QueryBuilder*, după cum spune și numele, folosește șablonul de proiectare Builder pentru a „construi” o interogare. Concret, *QueryBuilder* ajută prin simplificarea scrierii interogării și a obținerii de rezultate. În plus, această librărie

previne atacurile de tip SQL Injection prin fixarea variabilelor (eng. *Bind Variable*) în mod automat (Fig. 9).

```
$query = $this->db->get('mytable');

foreach ($query->result() as $row)
{
    echo $row->title;
}
```

Fig. 9 Exemplu în care se folosește QueryBuilder pentru a extrage toate datele din tabela „mytable”

1.4 Tehnologii Front-end

Pentru a realiza interfața grafică, am recurs la tehnologiile clasice pentru dezvoltarea de pagini web: HTML, CSS, JavaScript.

Dezvoltarea paginilor HTML a fost realizată cu ajutorul librăriei *Bootstrap 4*, care ajută în principal prin oferirea unui set de instrumente pentru realizarea de pagini web responsive și orientate către utilizatorii mobile.

Principala metodă prin care au fost construite paginile aplicației de evaluare a profesorilor a fost prin utilizarea modulului de layout CSS Flexbox. Întregul framework Bootstrap 4 este construit pe baza modulului Flexbox întrucât reprezintă o nouă metodă de aranjare în pagină a diferitelor elemente. Flexbox este o alternativă mai ușoară la clasicele metode de layout precum block, inline, table sau positioned din cauză că elementele pot fi așezate în diferite direcții, ordinea lor de afișare poate fi rearanjată, aliniamentul poate fi modificat și pot fi potrivite dinamic în cadrul containerului din care fac parte.

Un dezavantaj al Flexbox este că multe browsere nu suportă această modalitate de afișare, în special browser-ele cu versiuni mai vechi pentru telefoane.

Bootstrap oferă un ajutor când vine vorba de implementarea elementelor Flexbox, având deja implementate diferite clase CSS des folosite, reducând astfel nevoia de a rescrie de multe ori aceleași bucăți de cod (eng. *Boilerplate Code*).

Pentru a realiza un design orientat către utilizatorii de telefoane am recurs la funcționalitățile oferite de Bootstrap, iar pentru cazurile mai dificile am recurs la *Media Queries* pentru a schimba orientarea, mărimea sau stilul anumitor elemente în funcție de dimensiunea ecranului.

Limbajul JavaScript a fost folosit intensiv pentru toate paginile aplicației. Principalul motiv pentru utilizarea acestuia este de a crea o experiență mai bună de folosință pentru utilizatori, reducând numărul de încărcări ale paginilor prin folosirea de apeluri XMLHttpRequest. În plus am inclus și librăria jQuery care ajută în dezvoltarea de cod JavaScript prin simplificarea iterării și manipulării documentelor HTML, a animațiilor și a gestionării evenimentelor.

În mare parte, am folosit apeluri de tip AJAX, care se ocupă cu schimbarea de informații între server și client în fundal, astfel încât să modific bucăți de conținut în funcție de acțiunile utilizatorilor. Un astfel de exemplu este reprezentat de funcționalitatea de a căuta o materie. Un utilizator poate să tasteze materia dorită și după aproape fiecare tastare, va fi făcută o cerere către server în urma căreia vor fi afișate rezultatele primite. Tot ce a fost găsit anterior va fi șters, iar noile recomandări vor fi afișate în locul celor vechi.

O problemă care apare cu acest tip de implementare a funcționalității de căutare prin intermediul apelurilor AJAX este că server-ul va fi invadat de un număr foarte mare de cereri pe secundă. Întrucât serverul Apache al facultății de informatică are activată o protecție împotriva unui număr mare de cereri pe secundă de la același client, este necesară elaborarea unui mecanism de temperare a cererilor de tip AJAX.

Un astfel de mecanism pe care l-am aplicat cu succes în aplicația de evaluare a profesorilor este împrumutat din domeniul ingineriei electronice și se numește *debouncing*. Acțiunea de *bouncing* reprezintă tendința a două contacte metalice într-un dispozitiv electronic de a genera semnale electrice multiple atunci când contactele lor se deschid sau se închid. Un exemplu foarte bun este reprezentat de tastatura folosită la calculatoare. În momentul când apăsăm o tastă, ne așteptăm ca un singur

contact să fie făcut. În realitate există un contact inițial urmate de alte două contacte. Soluția pentru această problemă este orice tip de dispozitiv hardware sau un software care să asigure că doar un singur contact este făcut atunci când un contact electric este deschis sau închis.

În cadrul aplicației de evaluare a profesorilor am utilizat conceptul de *debouncing* pentru a trimite o singură cerere de căutare a unei materii către server odată la fiecare 500 ms., dacă există una sau mai multe taste apășate. În JavaScript, acest procedeu poate fi realizat prin apeluri de tip *setTimeout* și *clearTimeout*, astfel încât cererea de căutare să fie făcută după 500ms atunci când o tastă este apășată, iar dacă o nouă tastă este apășată, atunci cererea precedentă va fi ștearsă, iar cea nouă va fi executată după 500 ms. Implementarea acestei modalități de căutare este disponibilă în Fig. 10.

```
function requestReviewInformation() {

    var searchURL = BASE_URL + "evaluation/get_reviews_for_query?q=";
    var inputMessage = $(this).val();

    clearTimeout(debounceTimeout);
    debounceTimeout = setTimeout(function() {
        executeAjaxRequest(searchURL + inputMessage, "GET", fillReviewBoxes);
    }, 500);
}
```

Fig. 10 Modalitate de a efectua o căutare a unei materii după un cuvânt prin AJAX, utilizând *debouncing*

Cu ajutorul tehnologiilor de Front-end și Back-end am reușit implementarea unui sistem de modificare a informațiilor importante din baza de date prin intermediul unor tabele. Astfel, acest sistem reprezintă o interfață pentru administratori asupra bazei de date prin care pot adăuga sau modifica în mod sigur, rapid și eficient diferite date ale aplicației. Am folosit plugin-ul pentru jQuery numit *DataTables* pentru a crea o infrastructură prin care pot manipula și afișa tabelele din baza de date. *DataTables* oferă suport pentru acțiuni precum editare, sortare sau paginare, efectuate ușor prin AJAX. Această cerere AJAX este trimisă către server prin metoda GET, iar apoi, pe parte de back-end, în funcție de existența și valoarea parametrilor trimiși, se generează

o interogare în baza de date iar rezultatul ei este mai apoi trimis ca răspuns la client printr-un JSON. În final răspunsul este interpretat de către API-ul *DataTables* și afișat în tabel.

Întrucât mulți oameni preferă vizualizarea datelor sub forma unor grafice în detrimentul altor modalități de reprezentare statistică, am recurs la o librărie de JavaScript care poate crea o sumedenie de grafice în format SVG, pe baza unor date specificate. Librăria *Chartist* este una din librăriile open-source preferate de către

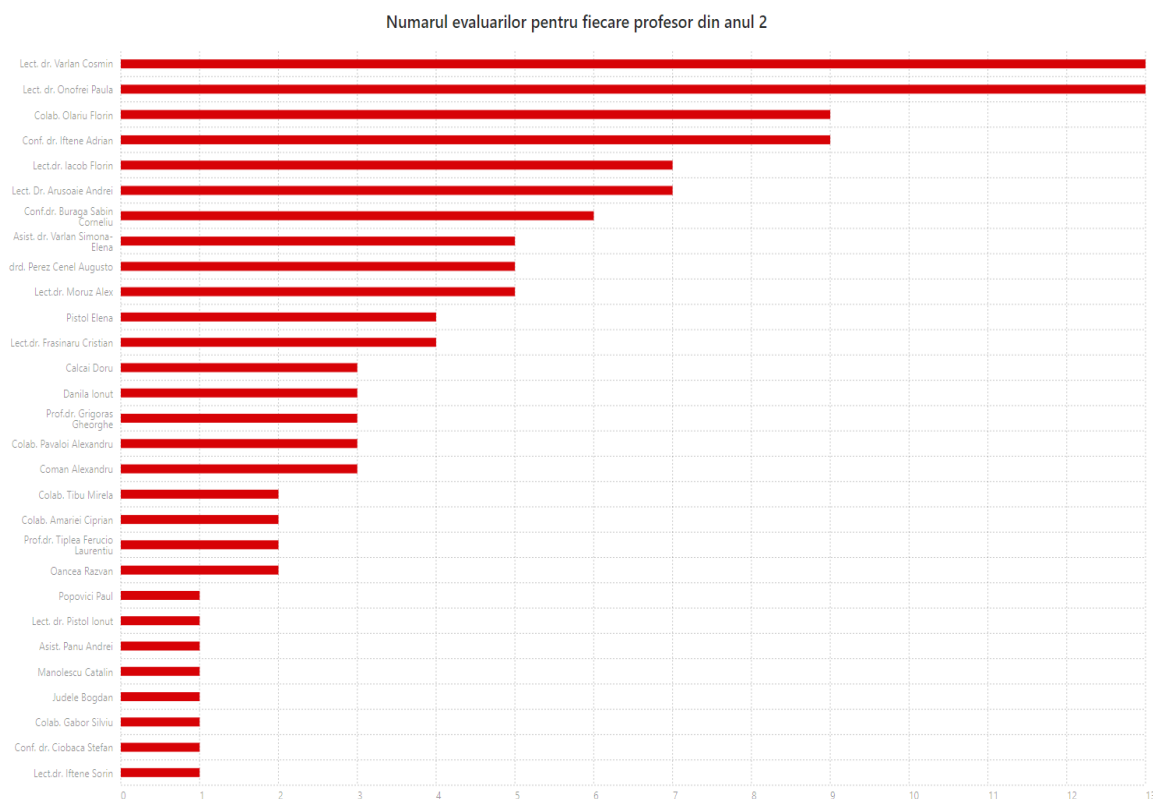


Fig. 11 Exemplu unde se folosește librăria *Chartist* pentru a afișa date privind numărul evaluărilor pentru fiecare profesor dintr-un an

dezvoltatorii Front-end datorită simplității în folosire, a design-ului graficelor și a compatibilității cu toate tipurile de ecrane. Am utilizat această librărie în pagina de statistici, vizibilă de către studenți pentru a afișa statistici legate de perioada curentă de evaluare. Un exemplu în care am utilizat librăria *Chartist* este pentru a afișa numărul evaluărilor pentru toți profesorii dintr-un anumit an, așa cum se poate vedea în Fig. 11.

2 Prezentarea aplicației

Această lucrare are ca obiectiv crearea unei aplicații web pentru evaluarea profesorilor din facultatea de informatică, ce poate fi lansată pe server-ul facultății și mai apoi deschisă pentru evaluări la sfârșitul fiecărui semestru. Implementarea acestui obiectiv este realizată prin utilizarea diferitelor tehnologii și concepte prezentate în capitolele anterioare. Astfel, aplicația este compusă din doua părți, partea de front-end și partea de back-end. Pe partea de front-end am utilizat HTML, CSS și JavaScript pentru crearea unei interfețe grafice prietenoase cu utilizatorul, ușor de folosit și compatibilă cu toate tipurile de dispozitive ce au browsere de internet moderne. Partea de back-end este realizată în PHP folosind framework-ul CodeIgniter și este menită să gestioneze toată logica aplicației, să administreze diferite informații din baza de date și să creeze toate paginile HTML.

Aplicația este menită să funcționeze fără o intervenție externă foarte mare din partea administratorilor astfel încât toate aspectele ce țin de evaluarea profesorilor precum adăugarea de date, de perioade de evaluare sau de interpretare a rezultatelor să poată fi făcută din cadrul aplicației.

Aplicația de evaluare a profesorilor este gândită pentru a fi folosită de 4 tipuri de utilizatori:

- Studenți;
- Profesori;
- Decan;
- Administratori.

În continuare voi prezenta funcționalitățile acestei aplicații pentru fiecare tip de utilizator.

2.1 Modulul pentru studenți

Studenții au posibilitatea sa-și evalueze profesorii pe care i-au avut în timpul ultimului semestru. Pentru a putea avea acces în aplicație, ei trebuie sa-și furnizeze contul de fenrir și parola, create la începutul primului an universitar, autentificarea fiind făcută prin intermediul protocolului LDAP.

Dacă nu există nici-o perioadă de evaluare activată de către un administrator, atunci studenții vor fi întâmpinați de un mesaj de eroare care le va indica acest lucru, spunându-le să revină la sfârșitul semestrului curent sau următor.

Odată autentificați în aplicație, ei trebuie sa-și selecteze grupa din care fac parte pentru a găsi toate materiile specifice acesteia (Fig. 12). Grupa se selectează o singură dată, la prima autentificare, acesta fiind stocată în baza de date pentru ca studentul să nu mai fie nevoit din nou să o selecteze.

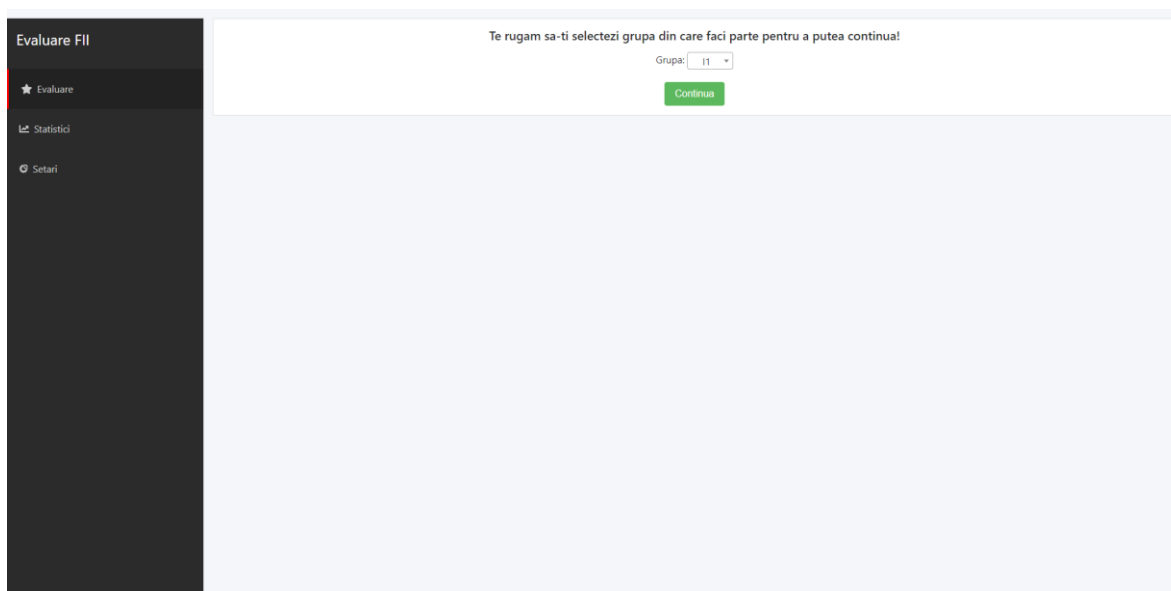


Fig. 12 Opțiunea pentru selectarea grupei la prima autentificare în aplicație a studenților

După setarea grupei, studenții vor avea access la tot conținutul creat pentru ei, precum opțiunea de a evalua profesorii, de a căuta o materie sau de a vizualiza statisticile curente.

Pentru a evalua o materie, studenților le este prezentată o pagină cu toate materiile specifice grupei din care fac parte. Fiecare materie este dispusă sub forma unei mici „cărți”² în care sunt trecute: numele materiei, tipul materiei și numele profesorului care predă materia respectivă.

Dacă un student a evaluat deja o materie, cartea acesteia în pagina de materii disponibile va fi bifată, el nemaiputând să o evalueze din nou (Fig. 13). Pentru restul materiilor disponibile pentru evaluare, dacă studentul selectează una din ele, îi va fi

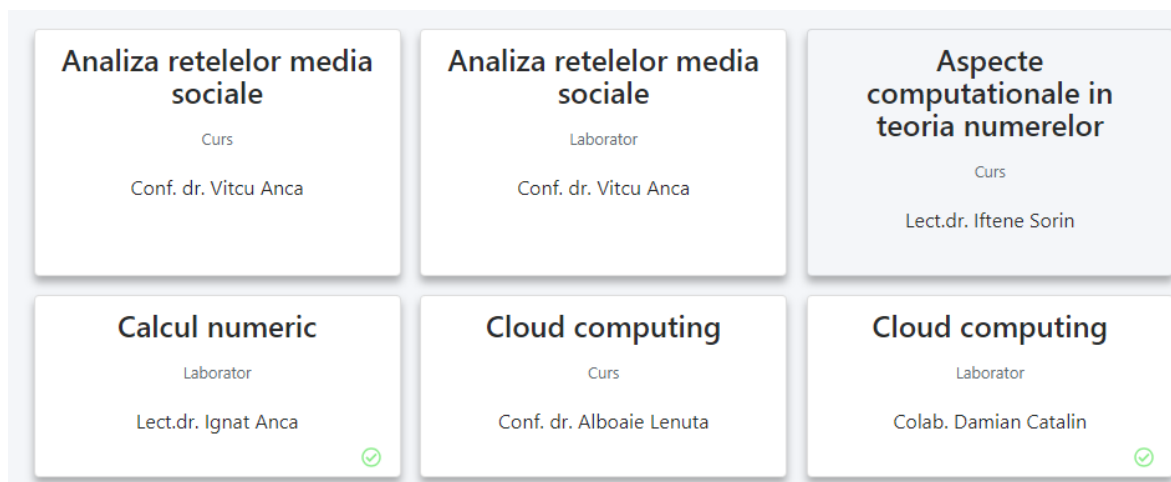


Fig. 13 O parte din pagina unde sunt afișate evaluările ce trebuie efectuate împreună cu cele deja efectuate

prezentat un modal³, în care sunt afișate toate întrebările la care trebuie să răspundă împreună cu toate variantele de răspuns (Fig. 14). Acest modal este disponibil pe mai multe pagini, astfel încât experiența de utilizare să devină mai simplă. Pe lângă setul de întrebări, în modal se regăsește și o secțiune pentru comentarii pozitive și comentarii negative alături de opțiunea de selectare a numărului de cursuri/seminarii la care a participat studentul pe parcursul semestrului. Odată completată evaluarea, răspunsul este trimis către server, verificat și dacă nu există erori, un mesaj de succes va fi afișat. După aceasta, cu ajutorul JavaScript, se marchează printr-o bifă cartea cu materia evaluată.

² O „carte” (eng. Card) este un tip de container pentru conținut, disponibil prin framework-ul Bootstrap ce aranjează diferite elemente într-un chenar de dimensiuni variabile, similar unei cărți de vizită

³ Un modal reprezintă o pagină ce va apărea peste conținutul curent, similar ca o alertă

Evalueaza

Materie: Analiza rețetelor media sociale
 Tip: Laborator
 Profesor: Conf. dr. Vitcu Anca

Anonimitatea va este garantata 100%. Va rugam sa evaluati cu incredere!

20%

1 2 3 4 5

1. Cursul/seminarul este structurat logic, clar si concis.

Nu ma pot pronunta Niciodata Destul de rar Destul de des Intotdeauna

2. Cursurile/seminarile aduc elemente noi pe baza unei bibliografii relevante.

Nu ma pot pronunta Niciodata Destul de rar Destul de des Intotdeauna

3. Cursurile/seminarile contin si rezultate ale cercetarii stiintifice din domeniu.

Nu ma pot pronunta Niciodata Destul de rar Destul de des Intotdeauna

4. Dupa ce am studiat aceasta materie interesul meu pentru domeniu a crescut.

Nu ma pot pronunta Niciodata Destul de rar Destul de des Intotdeauna

5. Profesorul enunta din inceput obiectivele didactice si urmareste atingerea lor.

Nu ma pot pronunta Niciodata Destul de rar Destul de des Intotdeauna

Fig. 14 Modalul ce conține formularul de evaluare. Se observă că este format din 5 pagini și sunt afișate primele 5 întrebări

Pentru a putea să găsească o materie mai ușor, am oferit studenților opțiunea de căutare. Studenții pot introduce numele unei materii, a unui profesor sau a unui tip de materie (curs/seminar). Pe măsură ce scriu, rezultatele vor apărea imediat, nemaifiind nevoie să apese pe vreun buton sau să părăsească pagina curentă. Acest lucru este realizat prin AJAX, existând un listener în JavaScript pentru evenimentul de tastare. Am aplicat inclusiv tehnica de debouncing pentru această funcționalitate, o descriere mai detaliată găsindu-se la capitolul Tehnologii Front-end.

Studenții pot să vadă o pagină cu statistici (Fig. 15) unde sunt prezentate numărul de evaluări efectuate de către ei în perioada curentă de evaluare, împreună cu numărul total de evaluări, numărul total de profesori și numărul total de materii tot din perioada curentă de evaluare.

Pe această pagină am pus la dispoziție și un grafic de tip bar chart ce afișează numărul de evaluări pentru fiecare profesor din anul în care se află studentul. Datele pentru acest grafic sunt preluate în mod asincron prin AJAX în momentul încărcării paginii.

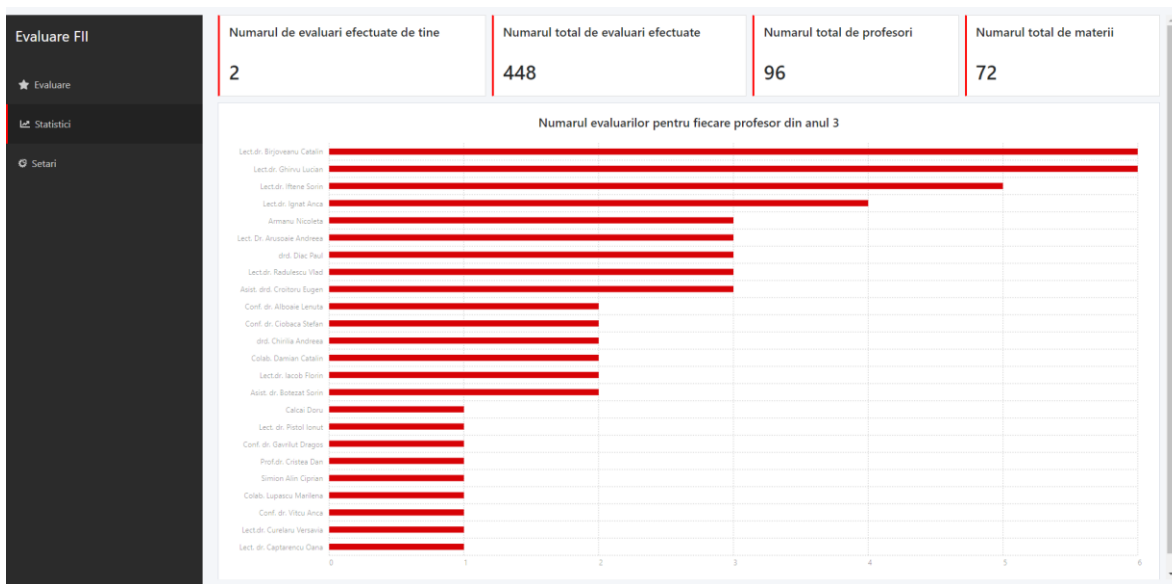


Fig. 15 Pagina de statistici, vizibilă de către studenți

Pentru studenți există și o pagină de setări, unde pot să-și schimbe grupa. Am introdus această funcționalitate din cauză că pot exista situații în care un student și-a schimbat grupa pe parcursul semestrului sau situații în care a rămas restant la o materie și trebuie să o repete. Astfel, studenții au libertatea să-și aleagă orice grupă de la licență sau de la master. Schimbarea grupei se face tot printr-o cerere de tip AJAX, eliminând astfel nevoia de a încărca din nou pagina.

Pagina de setări cuprinde și opțiunea prin care studenții să trimită mesaje administratorilor privind anumite probleme sau dificultăți întâmpinate în cadrul aplicației. Am introdus varianta aceasta pentru a ușura modalitatea de comunicare cu administratorii, fără a mai fi nevoie să compună un email și să-l trimită la o anumită adresă. Pagina de setări este vizibilă în Fig. 16.

Fig. 16 Pagina de setări pentru studenți

2.2 Modulul pentru administratori

Administratorii au un rol foarte important în buna funcționare a acestei aplicații, ei controlând majoritatea proceselor și a datelor. Pentru a deveni administrator, un cont de utilizator trebuie promovat în această funcție de către un alt administrator.

Generarea perioadelor de evaluare se poate face de către un administrator din pagina numită „generatori” (Fig. 17). O perioadă de evaluare are ca scop, generarea și inserarea tuturor datelor în baza de date astfel încât studenții să poată evalua profesorii pe care i-au avut în timpul semestrului. În urma generării perioadei de evaluare, sunt inserate în baza de date, informații precum: structura orarului facultății în momentul generării sau numele complet însoțit de gradul didactic al fiecărui profesor. O perioadă de evaluare poate fi generată doar dacă nu există o altă perioadă de evaluare activă, în care studenții pot evalua. Acest lucru este specificat în tabela *Evaluation Period* din baza de date print câmpul *is_active*, iar constrângerea este verificată printr-un trigger. Mai multe detalii legate de tabela *Evaluation Period* se găsesc la capitolul: 16Baza de date MySQL.

În urma generării unei perioade de evaluare, profesorii sunt inserați în baza de date, doar dacă numele lor complet nu există deja. Dacă există, atunci gradul lor

didactic, stocat în tabelă, va fi actualizat în caz că este diferit. Această funcționalitate este introdusă pentru cazurile în care unui profesor i se schimbă gradul didactic de la un semestru la altul.

Am inclus posibilitatea generării unei perioade de evaluare prin mai multe modalități, necesitatea unei redundanțe în cazul de față fiind justificată de anumite cazuri excepționale sau modificări ce pot apărea pe parcurs.

Prima modalitate este cea mai ușoară și mai rapidă de folosit, funcționând pe baza unui scraper⁴, conceput să analizeze paginile HTML ce conțin orarul. Aceste pagini HTML sunt indicate prin mai multe URL-uri într-un fișier de configurare și specifică locația orarelor pentru fiecare an de la licență (I1, I2, I3) și pentru fiecare an de la master (MIS1, MIS2, MLC1, MLC2, etc.). La început, datele introduse în formularul de generare precum: anul universitar, semestrul, metoda de generare sau intervalul de evaluare sunt validate și apoi adăugate în tabela *Evaluation Period*, id-ul rezultat fiind folosit ulterior la inserarea materiilor. Întrucât, informațiile din orar sunt așezate în format tabelar, „scraper-ul” va extrage foarte ușor fiecare rând din tabel și îl va asocia fiecărui profesor, într-un dicționar. Această separare se face cu ajutorul unor expresii regulate ce sunt disponibile în fișierul de configurare, astfel încât să fie modificate cu ușurință. Pentru fiecare profesor, se va separa gradul didactic sau gradele didactice de nume și vor fi introduse într-un câmp nou. Acest lucru este implementat pentru a asigura schimbarea gradelor didactice în viitor. Dacă există o materie predată de mai mulți profesori, atunci materia respectivă va fi asociată la ambii profesori în dicționar. În final, dicționarul este iterat, iar fiecare profesor este adăugat sau modificat în baza de date. După adăugarea profesorilor, fiecare materie asociată unui profesor în dicționar va fi la rândul ei adăugată în baza de date, fiindu-i asociată id perioadei de evaluare anterior inserate și un id al unui profesor. Dacă formatul orarului facultății

⁴ Scraping reprezintă o metodă prin care este extras conținut dintr-o pagină de internet pentru a servi anumitor scopuri.

Fig. 17 Pagina de generare a unei perioade de evaluare, disponibilă pentru administratori

este modificat atunci această metodă de generare devine mai greu de folosit și se impune folosirea celorlalte maniere de generare.

A doua modalitate este gândită pentru a asigura compatibilitatea cu fosta aplicație de evaluare a profesorilor, input-ul în acest caz fiind un fișier text cu rezultatul scraper-ului anterior. Formatul acestui fișier este reprezentat de mai mulți vectori din PHP asociați unui an și conțin numele fiecărei materii împreună cu profesorul ce predă materia respectivă, despărțite prin diez (#).

A treia modalitate este creată pentru a introduce date pe baza unui fișier CSV. În acest mod nu mai există constrângeri privind formatul orarului sau compatibilitatea cu fosta aplicație, datele introduse trebuind să fie dispuse în modul următor: grad didactic profesor, nume complet profesor, nume materie, tip materie, grupa/an. De precizat că trebuie introduse toate materiile pentru toți anii de la licență și master într-un singur CSV și încărcat în aplicație.

Administratorii au posibilitatea să modifice diferite date din aplicație, dintr-o pagină specială (Fig. 18). În această pagină se găsește conținutul pentru majoritatea tabelor din baza de date, scopul fiind de a altera într-o manieră rapidă informații importante precum, textul unei întrebări sau al unei variante de răspuns, numele unui

| Nr. Crt. | An Universitar | Semestru | Grad didactic profesor | Nume complet | Materie | An | Tip | An / Grupa | Actiuni |
|----------|----------------|----------|------------------------|--------------------|--|----|-----------|------------|---------|
| 1 | 2017 - 2018 | 2 | dir. | Perez Cene Augusto | Limba engleza IV | 2 | Seminar | I2B1 | |
| 2 | 2017 - 2018 | 2 | Lect.dr. | Iftene Sorin | Introducere in criptografie | 2 | Laborator | I2B5 | |
| 3 | 2017 - 2018 | 2 | Lect. dr. | Benchea Razvan | Object-Oriented Programming | 1 | Laborator | I1E2 | |
| 4 | 2017 - 2018 | 2 | Conf. dr. | Zalinescu Adrian | Probabilitati si statistica | 1 | Laborator | I1B1 | |
| 5 | 2017 - 2018 | 2 | Lect. dr. | Patrut Bogdan | Sisteme de operare | 1 | Laborator | I1B3 | |
| 6 | 2017 - 2018 | 2 | Conf. dr. | Ciobaca Stefan | Programare functionala | 2 | Laborator | I2B4 | |
| 7 | 2017 - 2018 | 2 | | Coman Alexandru | Metodologii de lucru in mediul Open Source | 3 | Curs | I3 | |
| 8 | 2017 - 2018 | 2 | Lect. Dr. | Arusoaie Andrei | Programare functionala | 2 | Laborator | I2A6 | |
| 9 | 2017 - 2018 | 2 | Colab. | Tibu Mirela | Didactica informatiilor | 2 | Seminar | I2A | |
| 10 | 2017 - 2018 | 2 | Lect.dr. | Birjoveanu Catalin | Smart Card-uri si Aplicatii | 3 | Laborator | I3B5 | |

Fig. 18 Pagina de administrare a datelor

profesor, sau al unei materii. Tot din această pagină se poate opri sau porni o perioadă de evaluare.

Pentru a putea vizualiza mesajele cu problemele trimise de studenți am creat o pagină separată (Fig. 19) cu toate mesajele primite. Acestea sunt ordonate după cele mai noi introduse și sunt afișate câte 5 pe pagină. Pentru a prezerva anonimatul studenților am ales să nu stochez și numele de cont al studentului care a introdus mesajul.

| Message Text | Date |
|--|-------------|
| search containerul are dimensiuni puse aiurea... | 18-Jun-2018 |
| butonul de "inainte" nu dispare pe ultima pagina d... | 17-Jun-2018 |
| variantele de raspuns se potrivesc foarte putin cu... | 17-Jun-2018 |
| variantele de raspuns se potrivesc foarte putin cu intrebarile | |
| seminare*... | 17-Jun-2018 |
| uneori apar doua bife in loc de una duap evaluare ... | 17-Jun-2018 |
| daca alegi o grupa generala I1 nu sunt afisate toa... | 13-Jun-2018 |
| Nu pot sa caut materii din anii anteriori.Daca eu ... | 13-Jun-2018 |
| Nu merge un buton... | 12-Jun-2018 |

Fig. 19 Pagina cu mesaje primite de la studenți

Administratorii mai au atribuția de a autoriza cadrele didactice să folosească aplicația. Aceștia trebuie să analizeze cererile făcute de profesori pentru un cont și să le valideze sau să le respingă. Pentru aceasta, am creat o pagina specială (Fig. 20) în care pot fi vizualizate toate cererile existente în funcție de tipul lor: aprobate, respinse sau în așteptare. O astfel de cerere este compusă din email-ul profesorului împreună cu numele acestuia. Rolul administratorului este de a analiza cererea respectivă și să confirme sau să infirme autenticitatea ei. Odată confirmată cererea, profesorului i se va trimite prin email-ul furnizat datele de autentificare în aplicație, acestea fiind formate din email și o parolă auto-generată. Tot acest mecanism fiind implementat pentru a se asigura că un profesor nu poate avea acces la evaluările altui profesor. Un profesor poate să facă o cerere pentru o adresă de email, doar dacă nu există deja o altă cerere pentru aceeași adresă de email care să se afle în stare de așteptare sau aprobată.

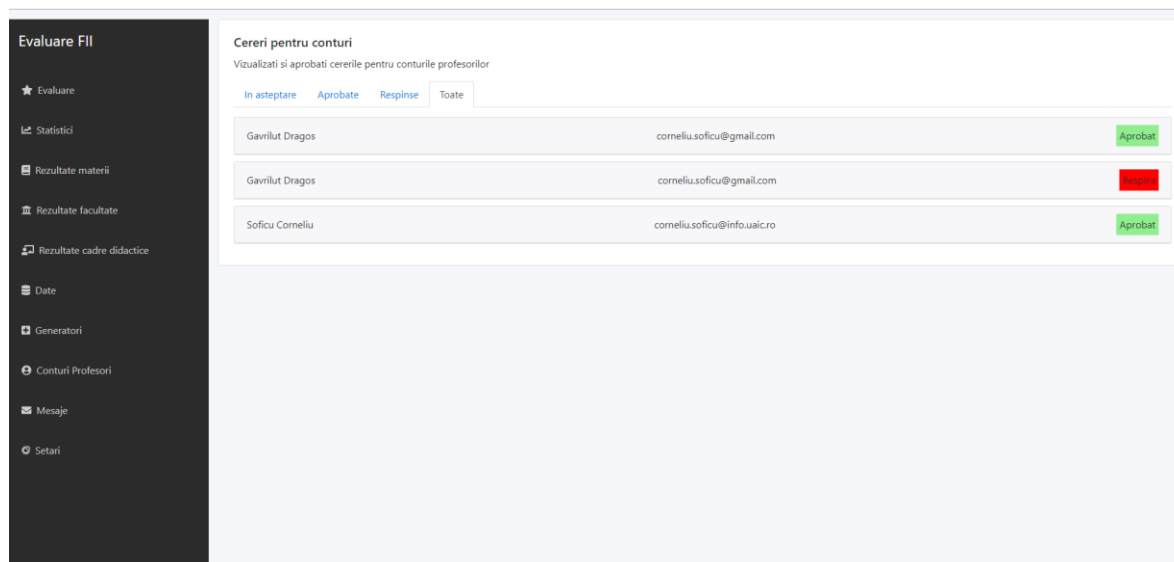


Fig. 20 Pagina de gestionare a cererilor pentru conturi de profesori

2.3 Modulul pentru profesori

Odată autentificați în aplicație, profesorii au opțiunea de a vedea rezultatele evaluărilor împreună cu evoluția lor pe parcursul mai multor perioade de evaluare (Fig. 21). Acest lucru se poate realiza foarte ușor doar prin selectarea unei perioade de evaluare. Pe aceeași pagină există și un grafic cu evoluția anumitor parametri precum, nota medie sau numărul de evaluări primite și pot fi văzute rezultate individuale în funcție de perioada de evaluare selectată.



Fig. 21 Pagina cu rezultatele evaluărilor pentru un profesor

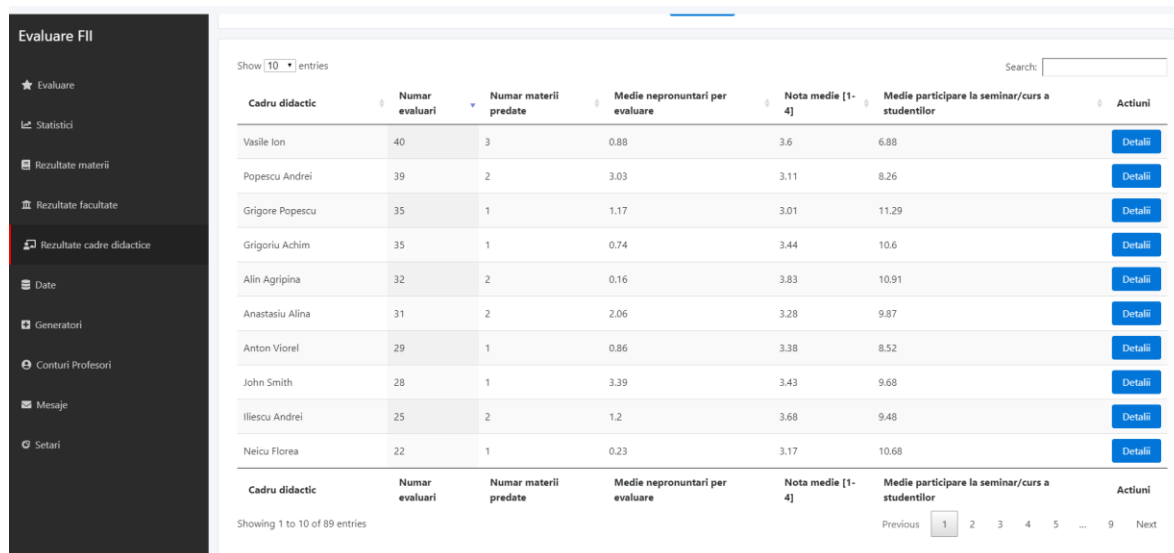
Profesorii mai au acces și la o pagină pentru a vedea rezultatele obținute pentru materiile predate în semestrul respectiv. Această pagină este similară cu cea precedentă, fiind organizată după fiecare întrebare la care au răspuns studenții.

2.4 Modulul pentru decan

Decanul are acces la toate evaluările profesorilor, putând vedea în acest mod, toate evaluările primite de fiecare profesor pentru fiecare materie. În plus el are acces la un tabel (Fig. 22) în care sunt centralizate rezultate generale pentru toți profesorii în funcție de anumiți parametri, precum: numărul de materii predate, numărul de

evaluări primite sau nota medie, toate fiind centralizate pentru o anumită perioadă de evaluare.

Tabelul respectiv a fost realizat cu ajutorul plugin-ului jQuery *DataTables* și oferă suport pentru paginare, sortare și căutare, decanul putând în acest mod să facă o comparație între rezultatele profesorilor.



| Cadrul didactic | Numar evaluari | Numar materii predate | Medie nepronuntari per evaluare | Nota medie [1-4] | Medie participare la seminar/curs a studentilor | Actiuni |
|-----------------|----------------|-----------------------|---------------------------------|------------------|---|-------------------------|
| Vasile Ion | 40 | 3 | 0.88 | 3.6 | 6.88 | Detalii |
| Popescu Andrei | 39 | 2 | 3.03 | 3.11 | 8.26 | Detalii |
| Grigore Popescu | 35 | 1 | 1.17 | 3.01 | 11.29 | Detalii |
| Grigoriu Achim | 35 | 1 | 0.74 | 3.44 | 10.6 | Detalii |
| Alin Agripina | 32 | 2 | 0.16 | 3.83 | 10.91 | Detalii |
| Anastasie Alina | 31 | 2 | 2.06 | 3.28 | 9.87 | Detalii |
| Anton Viorel | 29 | 1 | 0.86 | 3.38 | 8.52 | Detalii |
| John Smith | 28 | 1 | 3.39 | 3.43 | 9.68 | Detalii |
| Iliescu Andrei | 25 | 2 | 1.2 | 3.68 | 9.48 | Detalii |
| Neicu Florea | 22 | 1 | 0.23 | 3.17 | 10.68 | Detalii |

Fig. 22 Tabel cu rezultatele profesorilor, vizibil de către decan

3 Securitatea aplicației

Fiind o aplicație web cu acces pentru toată lumea și cunoscând că natura internetului expune o serie de vulnerabilități pe care persoane rău intenționate le pot exploata, devine obligatorie implementarea unor mecanisme de prevenție împotriva celor mai cunoscute tipuri de atacuri și vulnerabilități.

3.1 Autentificare

Autentificarea în aplicația de evaluare a profesorilor se poate face în două moduri în funcție de tipul utilizatorului.

Studentii se pot autentifica în aplicație cu ajutorul contului de student creat în primul an de facultate. Autentificarea propriu zisă se face prin protocolul LDAP,

numele gazdei serverului (eng. *hostname*) fiind *students.info.uaic.ro*. Dacă studenții se autentifică pentru prima dată, în baza de date, în tabela *Users*, va fi stocat numele de cont, iar grupul administrativ „Studenți” va fi selectat implicit. Tot la prima autentificare, în tabela de utilizatori se va stoca valoarea 1 în câmpul *is_newcomer* pentru a indica că studentul este nou venit și trebuie să-și selecteze o grupă pentru a putea efectua evaluări.

Pentru ca un profesor sau decanul facultății să se autentifice în aplicație, trebuie să completeze un formular reprezentând o cerere de creare a unui cont în care să furnizeze adresa de email împreună cu numele complet. Această cerere va fi validată manual de către un administrator, și dacă este validă, va atribui contul de profesor nou creat unei intrări în tabela *Professors*, după care se va trimite pe mail-ul furnizat parola de acces autogenerată. După ce a primit mail-ul, profesorul se poate autentifica în aplicație și apoi să-și vadă evaluările. Acest mecanism a fost conceput pentru a garanta că un profesor nu poate accesa contul altui profesor și să-i vadă evaluările. Dacă un profesor dorește să-și schimbe parola autogenerată, acest lucru este posibil, existând un formular special creat.

Datele de autentificare ale profesorilor sunt stocate în baza de date sub forma de email și parola, aceasta din urma fiind modificată (eng. *hashed*) prin intermediul algoritmului *bcrypt*. Modificarea parolei printr-un algoritm de *hashing* este necesară pentru a păstra în siguranța parola de autentificare, ea neputând fi reconstituită în forma inițială (eng. *plaintext*) conform proprietăților funcțiilor de *hashing*.

3.2 Autorizarea accesului

Aplicația de evaluare a profesorilor este gândită pentru a oferi acces mai multor tipuri de utilizatori, ca atare trebuie aplicată o politică de control al accesului pentru a gestiona într-o manieră sigură și clara acțiunile pe care le pot efectua aceștia în cadrul aplicației.

Ca atare, am recurs la modelul de control al accesului bazat pe roluri (Eng. Role-based access control) pentru a administra autorizarea în aplicație.

În cadrul acestei aplicații, rolurile sunt create în funcție de tipul utilizatorului, iar pentru fiecare rol, sunt atribuite mai multe permisiuni ce oferă capabilități de a realiza diferite operațiuni. Aplicația de evaluare a profesorilor respectă cele trei reguli principale ale modelului de control al accesului bazat de roluri:

1. Atribuirea rolurilor. Un subiect poate exercita o permisiune doar dacă utilizatorul a selectat un rol sau dacă i-a fost atribuit un rol. Utilizatorii aplicației de evaluare a profesorilor primesc rolul implicit de student, atunci când se autentifică prima dată în aplicație, iar schimbarea rolului se poate face doar de către un administrator.
2. Autorizarea rolurilor. Rolul activ al unui subiect trebuie autorizat pentru subiect. Împreună cu regula nr.1, această regulă asigură că utilizatorii pot avea rolurile doar pentru care sunt autorizați. Întrucât un utilizator primește un rol automat, la prima autentificare, sau îi este atribuit un rol de către un administrator, se asigură în acest mod că utilizatorul va avea un rol doar pentru categoria de utilizator din care face parte.
3. Autorizarea permisiunilor. Un subiect poate exercita o permisiune doar dacă permisiunea este autorizată pentru rolul activ al subiectului. Împreună cu regulile 1 și 2, această regulă asigură că utilizatorii pot exercita doar permisiunile pentru care sunt autorizați. Pentru a accesa diferite componente ale aplicației, un utilizator trebuie să posede anumite permisiuni. Lipsa acestora va determina redirectarea către o pagină de eroare cu accesul interzis.

După ce un utilizator se autentifică cu succes, se obțin permisiunile acestuia din baza de date în funcție de grupul din care face parte și se stochează în sesiune, pentru a putea fi verificate pe parcursul utilizării aplicației.

Pentru controller-ele fiecărui modul al aplicației de evaluare a profesorilor am creat un middleware ce se ocupă cu autorizarea accesului. În funcție de modul, dacă un utilizator încearcă să acceseze un anumit controller și nu are permisiunile necesare, el va fi redirectat către o pagină de eroare unde va fi specificat că nu are acces pentru resursa respectivă.

Rolurile din modelul de control al accesului sunt disponibile în baza de date, în tabela *Groups* iar permisiunile în tabela *Permissions*. Asocierile dintre roluri și permisiuni se fac în tabela *Groups Permissions* iar asocierea dintre un utilizator și un rol se face în tabela *Users* și în tabela *Professors* prin intermediul câmpului *group_id*.

Un exemplu ce conține câteva permisiuni din aplicație.

| Permisiune | Descriere | Roluri |
|-----------------------------|---|--|
| can_review | Permisiune pentru a evalua un profesor | Studenti |
| can_view_student_statistics | Permisiune pentru a vedea statistici făcute pentru studenți | Studenti |
| can_change_settings | Permisiune pentru a modifica diferite setări privind contul | Studenti, Profesori, Administratori, Decan |
| can_see_own_results | Permisiune pentru a vedea propriile rezultatele ale evaluărilor | Profesori |

Concluzii finale

Concluzii

Lucrarea „Aplicația de evaluare a profesorilor facultății de informatică” adoptă cele mai noi tehnologii din domeniul dezvoltării de aplicații web și își propune să ofere atât studenților cât și profesorilor un instrument care să asigure îmbunătățirea actului didactic și a experienței educaționale din facultate.

Contribuții personale

Am reușit prin tehnicile de dezvoltare back-end și front-end să creez o aplicație web care facilitează evaluarea profesorilor din facultate de către studenți, dezvoltând o interfață grafică prietenoasă cu utilizatorul și adaptată la majoritatea tipurilor de ecrane.

Aplicația este completată de modulul creat pentru profesori și cel pentru decan, prin care interpretarea rezultatelor devine mai ușoară, iar partea de administrare a aplicației ajută la buna funcționare pe termen lung. Folosindu-mă de infrastructura facultății de informatică am asigurat lansarea aplicației reușind în acest mod să dezvolt o platformă ce poate fi folosită de către multe generații de studenți și profesori de acum încolo.

Direcții de viitor

Cu siguranță, această aplicație poate fi îmbunătățită existând mai multe direcții pentru viitor:

- Extinderea folosirii de la nivelul facultății la nivelul universității, astfel încât să fie utilizată de toți studenții și toți profesorii universității;

- Îmbunătățirea componentei de front-end printr-un design mai ușor de folosit, mai atrăgător și adaptat pentru toate tipurile de dispozitive;
- Implementarea unei modalități prin care studenții să poată vedea evaluările anumitor profesori, astfel încât să fie mai bine informații asupra unor materii ce urmează să le parcurgă.

Bibliografie

1. *Art. 303 Legea 1/2011 Evaluarea calității cadrelor didactice.*
2. *Teaching and Learning International Survey (TALIS).* s.l. : OECD, 2013.
3. **Intenert Growth Statistics. *Internet World Stats.* [Interactiv]**
<https://www.internetworldstats.com/emarketing.htm>.
4. **Usage statistics and market share of PHP for websites. *W3Techs.* [Interactiv]**
Iunie 2018. <https://w3techs.com/technologies/details/pl-php/all/all>.
5. **Newsroom Facebook: Company Info. *Facebook Newsroom Website.***
[Interactiv] 31 Martie 2018. <https://newsroom.fb.com/company-info/>.
6. **30 Amazing Wikipedia Statistics and Facts. *Expanded Dramblings.* [Interactiv]**
Mai 2018. <https://expandeddramblings.com/index.php/wikipedia-statistics/>.
7. **PHP Versions. *PHP.* [Interactiv] <http://php.net/supported-versions.php>.**
8. **Rasmus, Lerdorf. *frOSCon.* Sankt Augustin, Germania, 2008.**
9. ***PHP MVC Framework Showdown: 7.1 Performance.* Bowman, Will. s.l. : Medium, 2017.**
10. **CodeIgniter - Application Architecture. *Tutorialspoint.* [Interactiv]**
www.tutorialspoint.com/codeigniter/codeigniter_application_architecture.htm.
11. **Hierarchical model–view–controller. *Wikipedia.* [Interactiv]**
https://en.wikipedia.org/wiki/Hierarchical_model%E2%80%93view%E2%80%93controller.