



Universitatea POLITEHNICA București
Facultatea Automatică și Calculatoare
Departamentul Automatică și Informatică Industrială

LUCRARE DE LICENȚĂ

MyACS – Portal web dedicat Facultății de Automatică și Calculatoare

Coordonator

Sl. dr. ing. Adriana Olteanu

Absolvent

Vlad – Alexandru Nicolae

Cuprins

1. Introducere	2
Prezentarea Capitolelor.....	2
Prezentarea domeniului din care face parte lucrarea	3
Aplicații existente în domeniu	5
Funcționalitățile aplicației	6
Motivarea alegerii temei	7
2. Tehnologii software	8
HTML.....	8
CSS	9
JavaScript.....	10
Bootstrap	11
Ajax	11
NodeJS	11
Express	13
Baze de date	13
MongoDB	14
3. Utilizarea aplicației	16
Autentificare	16
Home Page	17
Cursuri si Evenimente	19
Calendar	20
Hartă	22
Securitate	23
Forum.....	23
4. Implementarea aplicației	25
MVC	25
Implementarea Bazei de Date	27
Sistemul de login	30
Securitatea aplicației	34
Rute	34
Filtrarea datelor	39
Calendar	40
Harta	43
5. Concluzii.....	44
Implementări viitoare	44
Opinie personală	45
6. Bibliografie.....	46

1. Introducere

Prezentarea Capitolelor

Prezenta lucrare conține aspecte teoretice și aspecte tehnice, prezentate detaliat, despre modalitatea de concepere și implementare a portalului web.

Lucrarea este structurată în 6 capitole, fiecare capitol fiind alcătuit la rândul lui din mai multe subcapitole.

Capitolul 1 – Introducere: reprezintă partea introductivă a lucrării, prezentând domeniul din care face parte lucrarea, a temei propriu-zise, motivul alegerii acesteia, o prezentare succintă a funcționalităților implementate și o comparație cu funcționalitățile oferite de aplicații asemănătoare.

Capitolul 2 – Tehnologii software: reprezintă o scurtă descriere a tehnologiilor utilizate pentru realizarea acestei aplicații. Cum funcționează Internetul, serverele web, aplicațiile web, paginile statice și dinamice, HTML, CSS, JS, Node.js, Express, cele mai importante noțiuni ale bazelor de date și informații despre MongoDB - reprezintă principalele puncte de interes al acestui capitol.

Capitolul 3 – Utilizarea aplicației va conține prezentarea generală a aplicației, metoda de utilizare a interfeței cu utilizatorul, modalitățile de stocare a informațiilor și a datelor și ce metode de securitate au fost utilizate.

Capitolul 4 - Implementarea aplicației. În acest capitol se vor regăsi informații legate de felul în care aplicația a fost realizată, cum a fost împărțită în secțiuni, cum comunică între ele, bucăți de cod mai speciale.

Capitolul 5 – Concluzii și dezvoltări ulterioare: alături de o opinie personală privitoare la aplicația realizată, se vor găsi informații suplimentare despre aplicații similare, performanțele acestora și posibile îmbunătățiri ce se vor putea adăuga în viitor.

Capitolul 6 – Bibliografie: va conține o listă amănunțită a documentelor utilizate în realizarea acestei lucrări.

Prezentarea domeniului din care face parte lucrarea

În prezent 70% din populația Europei locuiește în mediul urban și consumă 70% din energia produsă. În România deși acest procent este mai mic, doar 56.4% din populație este urbană, conform Institutului National de Statistică. Migrația rural – urbană a fost datorată dorinței populației de a găsi un loc de muncă, de a avea acces la educație, îngrijire medicală, cultură, cu alte cuvinte de a găsi un trai mai bun. De cele mai multe ori această dorință a fost îndeplinită, însă prin creșterea semnificativă a populației din interiorul orașelor au fost generate noi probleme. Printre acestea se pot menționa:

- Infrastructura neadecvată: Orașele nu au fost proiectate pentru a acomoda o populație atât de numeroasă;
- Aspectele de mediu: Consum mare de energie – 70%, și generare ridicată a emisiilor de CO₂ – 75;
- Stresul urban: Costuri din ce în ce mai mari, rată ridicată a șomajului și a criminalității, poluare, aglomerație;
- Competiția economică: Preturile crescute ale produselor și a serviciilor;

Desigur, urbanizarea a avut și efecte pozitive, dintre acestea, probabil cel mai important este: dezvoltarea incredibilă a tehnologiei. Apariția tehnologiei informaționale a adus numeroase schimbări în viața noastră de zi cu zi și a mediului înconjurător, acestea fiind cel mai vizibile în domenii precum comunicația, securitatea publică, servicii de utilități. În toate aceste domenii vom găsi echipamente electronice avansate, conectate la o rețea prin care vor fi transmise diferite informații preluate de la un număr mare de senzori. Existența acestor elemente conduce la evoluția orașelor, spre orașe inteligente (smart city în literatura din limba engleză).

Sub umbrela conceptului de oraș inteligent se vor găsi toate soluțiile tehnologice necesare pentru a diminua problemele ce sunt întâlnite în orașele contemporane. O altă componentă a acestui concept este reprezentată de metodele de



Fig 1. Componentele conceptului "Smart City"

implementare a acestor soluții, atât la nivel fizic: infrastructură, cât și la nivel mental: instruirea populației și formarea unei noi mentalități pentru atingerea scopului final de creștere a calității vieții și de eficientizare a consumului de resurse.

Conceptul de oraș inteligent este în realitate un concept extrem de dinamic, un proces ce se afla într-o stare de transformare continuă prin apariția noilor inovații tehnologice, urbanistice sau manageriale. Departamentul de inovații de muncă și aptitudini al Marii Britanii a punctat aceasta, spunând: „conceptul nu este static: nu există o definiție absolută a unui oraș inteligent, nici un punct final, ci mai degrabă un proces, sau o serie de pași prin care orașele devin mai locuibile, și astfel, mai apte să răspundă noilor provocări”.

O variantă des întâlnită de definire a orașului inteligent este realizată prin referirea specifică a principalelor arii de aplicare a tehnologiilor inteligente. Astfel, el poate fi descris ca un oraș dotat cu: energie inteligentă, mobilitate inteligentă, clădiri inteligente, tehnologie inteligentă, sistem de siguranță inteligent, infrastructură inteligentă, sistem de sănătate inteligent, cetățeni inteligenți și sistem de educație inteligent.

Probabil dintre toate acestea, cea mai importantă componentă o reprezintă educația. Prin educație vor fi formate noile mentalități pe care populația trebuie să le dețină, precum și dezvoltarea forței de muncă necesară pentru implementarea tuturor proiectelor inteligente. Din acest motiv, putem spune că universitățile joacă un rol foarte important în dezvoltarea orașelor inteligente. Se naște astfel un nou concept, cel de campus inteligent.

El va reprezenta o comunitate în interiorul altei comunități, toate ideologiile de eficientizare a consumului de resurse fiind împrumutate și implementate din cele ale orașului inteligent. Marile universități din lume promovează în acest moment metode de eficientizare a procesului de gestionare al clădirilor, noi politici de transport, utilizarea unei varietăți de senzori și echipamente de comunicare, inovarea infrastructurii urbane și eficientizarea experienței de învățare.

În anul 2016, Universitatea Politehnica București a semnat un acord de finanțare pentru implementarea unui proiect de micro – rețea inteligentă în tot campusul. Scopul acestui proiect este de a reduce costurile anuale ale consumului de energie electrică și a emisiilor de gaze cu efect de seră. Acest lucru se va datora utilizării unor tehnologii noi, cum ar fi: sisteme fotovoltaice pentru acoperișuri, sisteme de stocare a energiei, eficientizarea iluminatului energetic, motoare pe gaz.

Centrul de cercetare PRECIS reprezintă un alt exemplu de evoluție la nivel de infrastructură a Universității Politehnice București, adăugând 28 de noi săli de laborator, dotate cu echipamente moderne, Facultății de Automatică și Calculatoare. Acestea sunt doar câteva din realizările UPB, ce o vor aduce cu un pas mai aproape de obținerea primului campus inteligent din România.

Pe lângă dezvoltarea infrastructurii și oferirea de componente hardware moderne, un campus inteligent conține și aplicațiile software ce au fost concepute cu scopul de a inova metodele de învățare și de a facilita accesul rapid la informațiile de care studenții au nevoie. Scopul acestora este de a îmbunătăți experiența facultății pentru fiecare student.

Una dintre aplicațiile cel mai des întâlnite în marile universități din lume este reprezentată de un portal web, de unde studenții pot accesa informații diverse de la finanțe și situație școlară, până la cele mai bune locații de relaxare din vecinătatea campusului. Ex: AXESS – Universitatea Stanford, Passport – Universitatea York, Abert – Universitatea din New York.

Aplicații existente în domeniu

Așa cum am menționat și în paginile precedente, o formă a acestui tip de portal a fost deja implementată în majoritatea marilor universități din lume. Desigur, acestea au fost dezvoltate separat, astfel capacitățile acestora precum și platforma utilizată diferă. Mai jos voi descrie funcționalitățile unor astfel de portale.

AXESS (Universitatea Stanford) – Acest portal poate să fie accesat de angajații universității, studenți și chiar și de părinții acestora. În interiorul portalului, un student va găsi informații personale, date legate de metoda de finanțare utilizată și alte modalități de a obține ajutor financiar, va putea vizualiza programul cursurilor și a examenelor, se va putea menține informat despre cele mai noi evenimente din campus și multe altele. Pentru angajații Stanford, portalul le oferă informații despre programul acestora, salariu, fluturașul de plată și alte informații confidențiale.

CAL State LA(Universitate de Stat din California) – Portalul oferă acces atât studenților cât și profesorilor la informații despre cursurile ce se desfășoară în interiorul facultății, precum și date despre cele mai recente activități sportive ce se desfășoară atât în cadrul facultății cât și în apropierea acesteia.

NC State App – Universitatea de stat din Carolina de Nord se bazează în întregime pe o aplicație mobilă ce le oferă studenților, posibilitatea de a accesa emailul facultății, să își vizualizeze orarul și să își formeze un program zilnic personalizat. Pentru a se asigura ca informațiile referitoare la evenimentele ce se desfășoară în universitate sunt accesibile tuturor, cei de la NC State au implementat o rețea de socializare proprie. Aceasta le va permite studenților să încarce fotografii.

Platforma Moodle (Facultatea de Automatica si Calculatoare) – Platforma moodele reprezintă o alta platformă în care rolurile utilizatorilor sunt separate. Un profesor va putea adauga materiale pentru cursurile sale, va seta termene limită pentru predarea unei teme, va compune teste de verificare a cunostintelor și multe altele. Un student va putea accesa toate informațiile postate pe pagina unui curs, va încarca teme sau va putea stoca alte documente personale, dar va și avea posibilitatea de a încarca un feedback despre cursul pe care l-a urmat.

Tema proiectului

MyACS reprezintă o platforma web ce a fost concepută pentru a îmbunătății gama de servicii oferite de alte platforme dezvoltate pentru facultatea de Automatică și Calculatoare. Scopul acesteia este de a se asigura că atât studenții cât și profesorii vor avea acces la cele mai noi informații ce apar în cadrul facultății. Utilizând câteva dintre cele mai noi tehnologii web, MyACS va face disponibile aceste informații de oriunde în lume, oferind astfel un avantaj major, în comparație cu formatul fizic utilizat în prezent.

Funcționalitățile aplicației

Funcționalitățile aplicației web sunt:

- Existența unui sistem securizat de autentificare, sistem ce va putea fi utilizat de studenții, cadrele didactice sau alte persoane autorizate din interiorul Facultății de Automatică și Calculatoare;
- Personalizarea contului utilizatorului prin adăugarea de informații suplimentare și poze de profil;

- Secțiune pentru vizualizarea cursurilor obligatorii, facultative sau opționale ce se vor desfășura în sălile ACS;
- Vizualizarea evenimentelor ce prezintă un interes pentru membrii facultății;
- Calendar ce va putea fi utilizat pentru a tine evidența orarului, a materiilor opționale și a altor evenimente personale, pentru a permite o organizare mai bună a timpului;
- Harta interactivă a interiorului facultății cu denumirea sălilor. Aceasta va oferi și posibilitatea de a afla mai multe informații despre o anumită sală, precum și un traseu rudimentar pentru a putea găsi sala respectivă mai rapid;
- Secțiunea dedicată evenimentelor critice și a comportamentului ce trebuie urmat pentru fiecare caz;
- Forum dedicat studenților pentru a pune întrebări și primi răspunsuri despre facultate, cursuri și concursuri;

Motivarea alegerii temei

Probabil unul din principalele motive pentru care am ales această temă a fost apartenența la noțiunea de Smart Campus. Această zonă este în permanentă dezvoltare, încercând mereu să introducă idei noi și inovative.

Fiecare universitate din lume încearcă să dezvolte o aplicație proprie sau să caute o platformă capabilă să ofere informații studenților în mediul virtual. Consider că nu se va ridica niciodată problema unui număr prea mare de informații, astfel pot spune că fiecare funcționalitate este importantă, oricât de mică ar fi.

Un alt motiv pentru care am ales să dezvolt această aplicație se datorează dorinței Universității Politehnica București de a deveni primul campus inteligent din România. Alături de universitate, Facultatea de Automatică și Calculatoare se află într-o perioadă de tranziție, încercând să mărească cantitatea de informație ce este distribuită în mod virtual. Digitalizarea informației oferă avantajul unei comunicări mult mai eficiente, rapide și ușor de direcționat. Mi-am dorit ca prin prezenta lucrare să contribui și eu la acest proces.

2. Tehnologii software

Apariția calculatorului a reprezentat un pas important în evoluția tehnologică. Acest dispozitiv, creat pentru a ușura viețile oamenilor, a devenit indispensabil în orice mediu de lucru și este din ce în ce mai mult utilizat și în mediul educațional. Calculatoarele moderne sunt capabile să ruleze aplicații complexe, să efectueze operații Dar pe lângă toate aceste funcționalități, calculatorul reprezintă o unealtă utilizată în comunicare, acest lucru fiind realizat prin Internet.

Prin Internet înțelegem ansamblul rețelilor de calculatoare interconectate, ce utilizează protocolul TCP/IP, din întreaga lume. Internetul reprezintă o rețea ce este formată la randul ei din alte rețele publice, private, academice, guvernamentale ce are un scop global și este obținută prin conectarea unui număr mare de dispozitive. Internetul transportă date informaționale dintr-o gamă largă de domenii și servicii, cum ar fi transportul de documente hypertext și aplicații ale World Wide Web (prescurtare WWW).

Construcția World Wide Web a fost realizată utilizând protocolul de transfer hypertext (Hypertext Transfer Protocol – prescurtare HTTP). Acest protocol este un protocol mic și rapid ce se potrivește foarte bine cu sistemele informatice multimedia. El oferă o tehnică de comunicare prin care paginile se pot transmite de la un calculator aflat la distanță către propriul calculator.

De cele mai multe ori, calculatorul aflat la distanță va rula un software de tip server web. Un server web reprezintă un program ce furnizează pagini web la cerere.

HTML

HTML (HyperText Markup Language) este un limbaj de calculator conceput pentru a permite formarea de site-uri web. Acesta este un limbaj foarte accesibil, ușor de înțeles și foarte puternic pentru ceea ce este capabil să realizeze. Sub influența organizației W3C, organizație ce are scopul de a proiecta și întreține limbajul, HTML evoluează pentru a putea respecta cerințele ce se găsesc într-o permanentă stare de creștere a audienței internetului.

Analizând fiecare termen ce intră în componența denumirii, putem extrage următoarele informații:

- HyperText: metodă pe care o utilizăm pentru a accesa diferite porțiuni ale internetului – apăsând pe un text special numit hyperlink putem accesa orice locație de pe internet, fără a fi nevoiți să respectăm o anumită ordine de a realiza un lucru;
- Markup: reprezintă ceea ce se va întâmpla cu textul ce se găsește în interiorul unor taguri html;
- Language: conține cuvinte cod și sintaxe ca orice alt limbaj;

HTML este format dintr-o serie de bucăți mici de cod ce sunt scrise în interiorul unui document text, acestea mai sunt cunoscute și sub numele de taguri HTML. Ulterior, salvând textul sub formatul unui fișier HTML, vom putea vizualiza conținutul sau utilizând un browser web. Browser-ul va citi fișierul și va traduce textul într-o formă vizibilă, așa cum a intenționat autorul.

Tagurile reprezintă ceea ce separa un text normal de cod HTML. Acestea sunt identificate foarte ușor, fiind formate din litere sau cuvinte ce se vor găsi în interiorul unor paranteze unghiulare ,<...>’.

CSS

CSS (Cascading Style Sheet) reprezintă un limbaj al cărui scop este de a simplifica procesul, de a face un site web prezentabil.

Utilizând CSS, un programator poate să controleze felul în care o pagina web va arăta, acesta având posibilitatea să aleagă ce culoare va avea un anumit text, ce font va fi utilizat, ce spațiere se va folosi între diferite elemente, ce modificări vor apărea în funcție de dispozitivul pe care îl utilizăm pentru a accesa o anumită pagină web și multe altele.

CSS oferă control asupra felului în care un document HTML va fi prezentat autorului sau publicului.

Utilizarea acestui limbaj oferă o serie de avantaje, printre acestea putând fi enunțate:

- Paginile se vor încarca mai rapid: Dacă vom utiliza CSS atunci nu va mai trebui să scriem attribute separate pentru fiecare tag html;

- Salvăm timp: Un fișier CSS poate fi scris o singură dată și reutilizat în diferite pagini html;
- Un număr mai mare de stiluri decât HTML: CSS utilizeze o gamă mai largă de attribute decât cele ce sunt găsite în limbajul HTML, oferind astfel posibilitatea de a oferi o prezentare mai bună unei pagini web;
- Compatibilitatea unui site web cu diferite dispozitive: Prin utilizarea limbajului CSS, conținutul unui fișier html va putea fi optimizat în așa fel încât acesta să arate la fel de bine, indiferent de platforma pe care o utilizăm pentru a accesa acea informație;

JavaScript

JavaScript este un limbaj dinamic de programare, care împreună cu HTML și CSS reprezintă cele trei tehnologii ce se găsesc în nucleul World Wide Web. Implementarea de scripturi în partea clientului permit ca acestea să interacționeze cu utilizatorul și să formeze pagini dinamice. Este un limbaj de programare interpretat, cu capabilitatea de a fi orientat spre obiecte.

Cea mai utilizată formă de JavaScript este cea din partea clientului. Un script poate fi inclus sau menționat într-un document HTML, ca mai apoi acesta să fie interpretat de către browser-ul web. Acest lucru înseamnă că o pagină web nu trebuie să fie statică, ci că aceasta poate conține programe ce vor interacționa cu utilizatorul, vor controla browser-ul sau vor crea conținut HTML în mod dinamic.

Cele mai importante avantaje ce sunt oferite de utilizarea limbajului JavaScript sunt:

- Mai puține interacțiuni cu serverul: Putem valida acțiunea utilizatorului înainte de a trimite o cerere către server. Realizând acest lucru, reducem traficul ce se realizează pe server, astfel micșorând încărcarea serverului;
- Oferirea imediată a unui răspuns pentru vizitatori: Un utilizator nu va trebui să aștepte ca pagina web să se reîncarce pentru a vedea dacă au uitat să completeze un anumit lucru într-un formular;

- Creșterea interactivității: Se pot realiza interfețe ce vor interacționa în momentul în care un utilizator realizează o anumită acțiune;
- Interfețe mai bogate: În urma utilizării limbajului javascript, apare posibilitatea de a include elemente noi într-o pagina web, de exemplu putem realiza operațiuni de drag&drop, oferind astfel o nouă experiență utilizatorilor;

Bootstrap

Bootstrap este o puternică arhitectură front – end, creată pentru a veni în ajutorul programatorilor atunci când dezvoltă aplicații web. Acesta conține o serie de schițe decorative predefinite, prin utilizarea de HTML și CSS, pentru componente des întâlnite în interfețele grafice – de exemplu: formulare, butoane, tabele de navigație, tabele și multe altele.

Utilizarea tehnologiei bootstrap într-un site web oferă desigur o serie de avantaje. Probabil cel mai important, este posibilitatea de a realiza site-uri ce modifică aspectul elementelor în funcție de dimensiunile ecranului dispozitivului de pe care sunt accesate.

Ajax

Ajax nu este un limbaj de programare. El poate fi descris mai mult ca fiind un concept. Ajax constă într-un script ce este executat în partea clientului ce oferă posibilitatea de a realiza comunicări cu serverul / baza de date, fără a necesita reîncărcarea paginii.

NodeJS

Node.js este o platformă dedicată pentru partea de server, ce a fost construită utilizând motorul javascript dezvoltat de Google Chrome, V8. Din anul 2009, anul în care limbajul a fost dezvoltat de Ryan Dahl, și până în prezent, definiția oferită de documentația oficială este:

„Node.js este o platformă construită pe baza motorului de rulare javascript a celor de la Google, dezvoltată pentru realizarea rapidă și scalabilă a aplicațiilor web. Node.js utilizează un model orientat spre evenimente non blocante de intrare și ieșire ce face ca aplicațiile să nu consume un număr ridicat de resurse și să fie foarte eficient în realizarea de aplicații ce operează în timp real și rulează pe diferite dispozitive.”

Node.js este un limbaj ,open source’, mediul de dezvoltare utilizat rulând pe diferite platforme. Aplicațiile Node.js sunt scrise în JavaScript și acestea pot rula pe sistemul de operare X a celor de la Apple, Windows sau Linux.

Principalele caracteristici specifice limbajului Node.js sunt:

- Orientat spre evenimente și asincron: Toate interfețele de programare ale aplicației, din biblioteca node.js, sunt asincrone, deci non blocante. În esența acest lucru înseamnă că un server ce utilizează aplicații dezvoltate utilizând node.js nu va aștepta niciodată ca interfața de programare să returneze date. Serverul se va ocupa de următoarea interfață de programare imediat ce aceasta este apelată. Mecanismul de notificări pentru evenimentele din node.js, va ajuta serverul să primească un răspuns de la apelurile anterioare ale interfeței de programare;
- Foarte rapid: Deoarece limbajul a fost construit bazându-se pe motorul javascript V8, dezvoltat de Google Chrome, biblioteca node.js este foarte rapidă în execuția codului;
- Un singur fir de execuție, dar foarte scalabil: Node.js utilizează un model bazat pe un singur fir de execuție cu evenimente ce rulează într-o buclă de execuție. Mecanismul de evenimente ajută serverul să răspundă într-o manieră non blocantă și face ca serverul să fie foarte scalabil, acest lucru fiind în opoziție cu serverele tradiționale ce formau fire de execuție limitate pentru realizarea apelurilor. Node.js utilizează un program cu un singur fir de execuție, program ce poate să ofere servicii unui număr mult mai ridicat de cereri decât serverele tradiționale, cum ar fi un server apache http;
- Nu există buffering: Aplicațiile node.js nu utilizează un buffer pentru date, acestea fiind afișate în blocuri;
- Licența: Node.js a fost lansat sub o licența MIT;

- Biblioteca bogata: Node.js dispune de un număr mare de module javascript ce simplifică procesul de dezvoltare al aplicațiilor web ce utilizează node.js

Express

Express.js este un framework de aplicații web pentru Node.js, ce a fost lansat ca un software gratuit, open-source, sub o licența MIT. A fost dezvoltat pentru crearea de aplicații web și interfețe de programare ale aplicațiilor. De asemenea, express.js a fost numit frameworkul standard al serverelor pentru Node.js.

Acesta este utilizat deoarece node.js nu a fost proiectat pentru a putea construi site-uri web. Framework-ul express este capabil să ofere o serie de funcții ce sunt necesare pentru a realiza acest lucru.

Scriș în javascript, express acționează doar ca un strat subțire ce intră în componența nucleului unei aplicații web. Express nu a fost creat doar pentru anumite tehnologii, acesta neavând o opinie cu privire la ce tehnologii dorim să îi atașăm. Această libertate, precum și faptul că este utilizat alături de viteza de dezvoltare și mediul pur javascript a lui node, a transformat expres într-un candidat puternic pentru dezvoltarea rapidă de aplicații și prototipuri;

Utilizând express, putem construi aplicații web și mobile cu o singură pagină, cu mai multe pagini sau hibride, funcții back-end pentru aplicațiile web dar și interfețe programabile ale aplicațiilor;

Express utilizează două motoare de șablonare: Jade și EJS ce facilitează transmiterea de date într-o structura web;

Express suportă o arhitectura de tip Model-Vizualizare-Control (eng: „Model-View-Controller” sau cum este mai cunoscut, model MVC) ce reprezintă o metodă ajutătoare de a dezvolta site-uri web într-o manieră orientată spre obiect.

Baze de date

O bază de date este o colecție de date structurată, ce poate fi stocată în format electronic pe un sistem de calcul, oferind facilități de acces și stocare în timp util.

Un sistem de gestiune al unei baze de date (SGBD) reprezintă un program software ce permite definirea structurii bazei de date, implementarea limbajelor de implementare, definirea și manipularea altor obiecte, specificarea facilităților privind stocarea datelor, asigurarea de drepturi și roluri de acces. Un SGBD izolează utilizatorul de mecanismele interne de stocare și acces.

O bază de date trebuie să asigure următoarele proprietăți:

- Abstractizarea datelor : datele trebuie prezentate într-o forma abstracta si structurata. Pentru a mări eficiența bazei de date, se formează structuri complexe de date înainte de memorarea acestora. Structura bazei de date este determinată de o combinație de scheme, relații, restricții ce trebuie să fie satisfăcute și legături ce se efectuează între date;
- Integritatea datelor: se referă la corectitudinea datelor conținute de baza de date și la modalitatea de manipulare a acestora astfel încât, acestea să satisfacă restricțiile impuse inițial;
- Independența datelor: modalitatea de organizare a datelor trebuie să fie transparența pentru utilizatori, iar modificările din baza de date nu trebuie să necesite rescrierea acestora pentru a funcționa în mod optim;
- Securitatea datelor: se realizează prin protejarea bazei de date de accese neautorizate. Acest lucru se realizează prin gestionarea accesului la informații și tipurile de operații pe care fiecare utilizator le poate efectua;
- Partajarea datelor: determină utilizarea bazei de date simultan de mai multe aplicații; se asigură gestiunea tranzacțiilor și a prelucrărilor concurente;

MongoDB

MongoDB este o bază de date gratuită, open-source, ce poate fi utilizată de pe o varietate de platforme și este orientată spre documente. Aceasta este clasificată drept o bază de date NoSQL.

O bază de date NoSQL nu mai utilizează structura clasică tabelară a bazelor de date relaționale, în schimb aceasta permite structurarea datelor mult mai logic. Acestea au apărut datorită vastității datelor moderne care de cele mai multe ori sunt nestructurate.

MongoDB a adaptat o structură asemănătoare fișierelor de tip JSON (fișiere ce au scheme dinamice) , formând astfel un nou tip de fișiere, denumite BSON. Acesta schemă face ca integrarea datelor în anumite tipuri de aplicații să fie mai rapidă. Mongo a fost construită pentru a fi scalabilă, accesibilă și să ofere performanță pornind de la utilizarea unui singur server până la infrastructuri mult mai complexe.

Principalele operațiuni ce sunt posibile într-o bază de date Mongo sunt:

- Realizarea de interogări ad hoc – se pot realiza interogări după câmp, expresii regulate sau interogări de interval;
- Indexarea: orice câmp dintr-un document BSON poate fi indexat;
- Replicarea: valabilitatea imediată a datelor este datorată unor seturi ce sunt formate de două sau mai multe copii ale datelor originale;
- Balansarea încărcăturii (Load Balancing) – „Sharding” reprezintă metoda utilizată de MongoDB pentru a-i permite să fie scalabilă orizontal. Acest lucru înseamnă că datele vor fi distribuite și împărțite în blocuri ce vor fi stocate în diferite „shard”-uri ce se vor găsi pe diferite servere. Pentru a determina cum au fost distribuite datele se vor utiliza chei pentru sharduri;
- Agregarea – Pentru procesarea în serie a datelor și realizarea operațiilor de agregare se va utiliza MapReduce;
- Stocarea fișierelor – Mongo poate fi utilizată ca un site de fișiere ce utilizează toate funcțiile de mai sus prin funcționarea într-o manieră distribuită prin intermediul procesului de sharding;

3. Utilizarea aplicației

Pentru a începe utilizarea aplicației, un client va trebui să acceseze portul 3000 al serverului local, prin comanda : „http://localhost:3000”. În urma acestei comenzi, va fi trimis fișierul predefinit primei pagini, login.ejs, ce va fi afișat în browser.

Autentificare

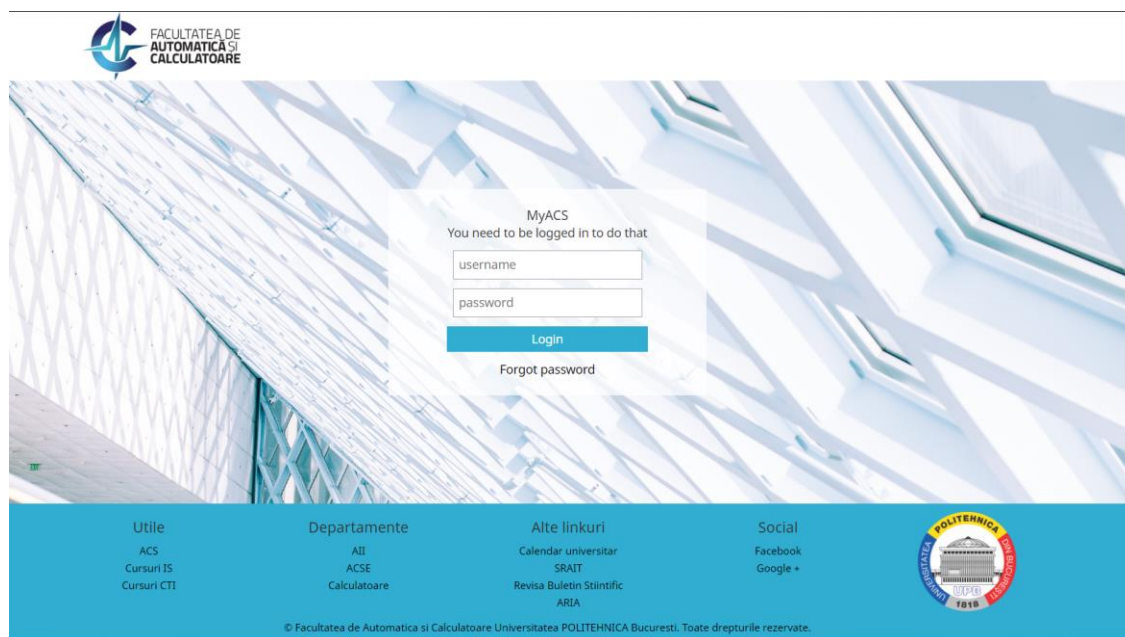


Fig. 2 – Pagina de autentificare

Așa cum se poate observa, prima pagină reprezintă pagina de autentificare a aplicației. Aceasta a fost concepută pentru a fi ușor de utilizat de către profesori și studenți.

În colțul din stânga sus se poate observa sigla Facultății de Automatică și Calculatoare, care împreună cu sigla Universității Politehnice București, din colțul dreapta jos, vor anunța instituțiile pentru care a fost realizată această platformă și oferă o oficializare a paginii. Tot în partea de jos a paginii se vor regăsi și alte linkuri utile. Acestea au ca scop reducerea timpului necesar navigării între diferitele platforme ACS.

Zona centrală cuprinde un formular necesar pentru realizarea operațiunii de autentificare a utilizatorului. Această acțiune se va finaliza prin introducerea numelui

utilizatorului - conform deciziei de proiectare, acesta va fi reprezentat de adresa de email oferită de UPB, a parolei și a apăsării butonului Login.

În situația în care un utilizator nu își mai amintește parola, acesta va putea apela la linkul „Ați uitat parola?”. Realizând acest lucru, se va deschide o nouă fereastră, ce va conține formularul atașat în partea dreapta a acestui paragraf. Aici utilizatorul va putea introduce adresa de email, iar prin apăsarea butonului „Reset Password” va fi lansată în execuție operațiunea de resetare parolă, care va consta în trimiterea unui email, în interiorul căruia se va găsi un link. Prin accesarea linkului se va deschide o nouă pagină, ce va permite introducerea unei parole noi, iar pentru a garanta faptul ca parola nu a fost introdusă greșit, am adăugat și opțiunea de a confirma parola dorită.

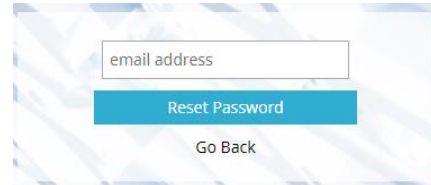
A screenshot of a password reset form. It features a text input field labeled 'email address', a blue button labeled 'Reset Password', and a smaller blue button labeled 'Go Back' below it. The background is a light blue geometric pattern.

Fig. 3 – Formular resetare parolă

Home Page

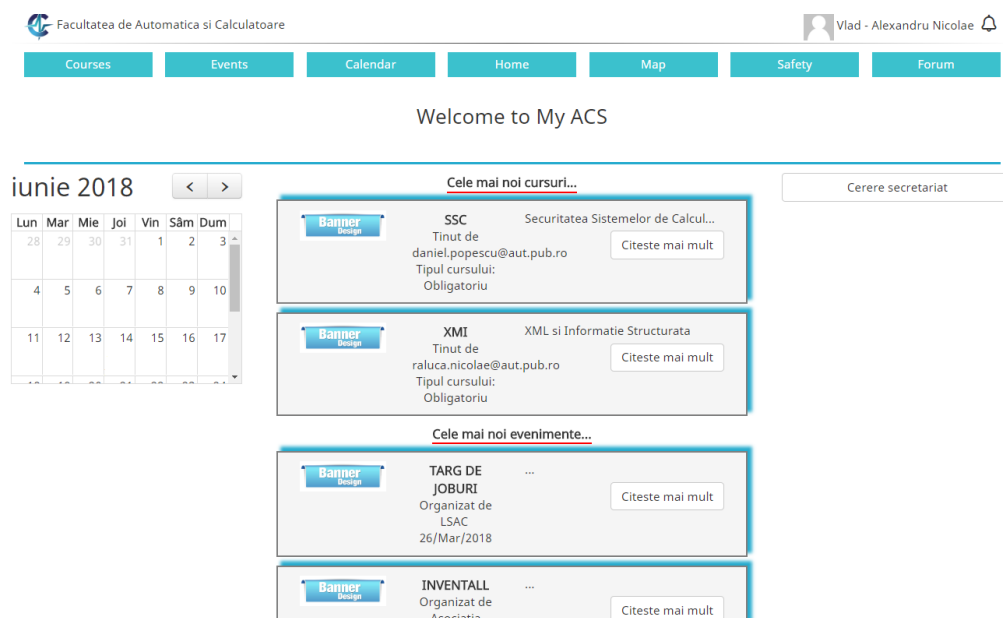
A screenshot of a web application's home page. At the top, there is a header with the logo of 'Facultatea de Automatica și Calculatoare' and a user profile for 'Vlad - Alexandru Nicolae'. Below the header is a navigation bar with buttons for 'Courses', 'Events', 'Calendar', 'Home', 'Map', 'Safety', and 'Forum'. The main content area starts with 'Welcome to My ACS'. On the left, there is a calendar for June 2018. The central part of the page is titled 'Cele mai noi cursuri...' and lists two courses: 'SSC' (Securitatea Sistemelor de Calcul...) and 'XMI' (XML si Informatie Structurata), each with a 'Citeste mai mult' button. Below this, another section titled 'Cele mai noi evenimente...' lists two events: 'TARG DE JOBURI' and 'INVENTALL', also with 'Citeste mai mult' buttons. A 'Cerere secretariat' button is located in the top right corner of the main content area.

Fig. 4 – Pagina Home

După ce procesul de autentificare a fost finalizat, va fi deschisa pagina intitulată ‚homepage’. Analizand reprezentarea vizuala de mai sus, putem observa ca antetul a fost oarecum schimbat. În coltul stanga sus se va putea gasi în continuare sigla facultăți, puțin restilizată, dar diversitatea antetului este datorată colțului dreapta sus. Aici, utilizatorii vor putea vizualiza imaginea de profil aleasă (sau așa cum este reprezentat și mai sus o imagine standard, oferită tuturor utilizatorilor), numele acestora și un clopotel utilizat pentru notificări. Navigând deasupra imaginii de profil sau a numelui, va apărea un submeniu din care vom putea accesa paginile: editare profil, vizualizare situație școlară sau să ne delogăm de la platforma MyACS.

Sub antet se va găsi meniul principal. De aici se vor putea accesa toate celelalte pagini din interiorul MyACS. Acesta este format din 7 elemente, fiecare element având un nume reprezentativ. Pentru un utilizator de tip student acestea vor fi: Courses, Events, Calendar, Home, Map, Safety și Forum. Pentru un utilizator de tip profesor, formatul meniului va fi schimbat, acesta având posibilitatea de a vedea informații despre studenții ce îi urmează unul sau mai multe cursuri. Și pentru un utilizator de tip secretariat vor apărea niște schimbări, mai exact, i se va oferi acces către pagina de vizualizare a studenților, de modificare a situației lor școlare, de adăugare a unor noi studenți în aplicație sau de vizualizare a cererilor realizate de aceștia.

Sub aceste informații se va putea găsi conținutul propriu zis al paginii Home. Această pagină este formată din câteva din funcționalitățile principale ale celorlalte pagini ale platformei. Acestea sunt:

- Un calendar lunar;
- Ultimele 3 postări din pagina de cursuri și pagina evenimentelor (modalitatea de afișare a acestora va fi explicată mai detaliat în cele ce urmează);
- Acces către pagina de trimitere a unor cereri către secretariat sau decanat;

Cursuri si Evenimente

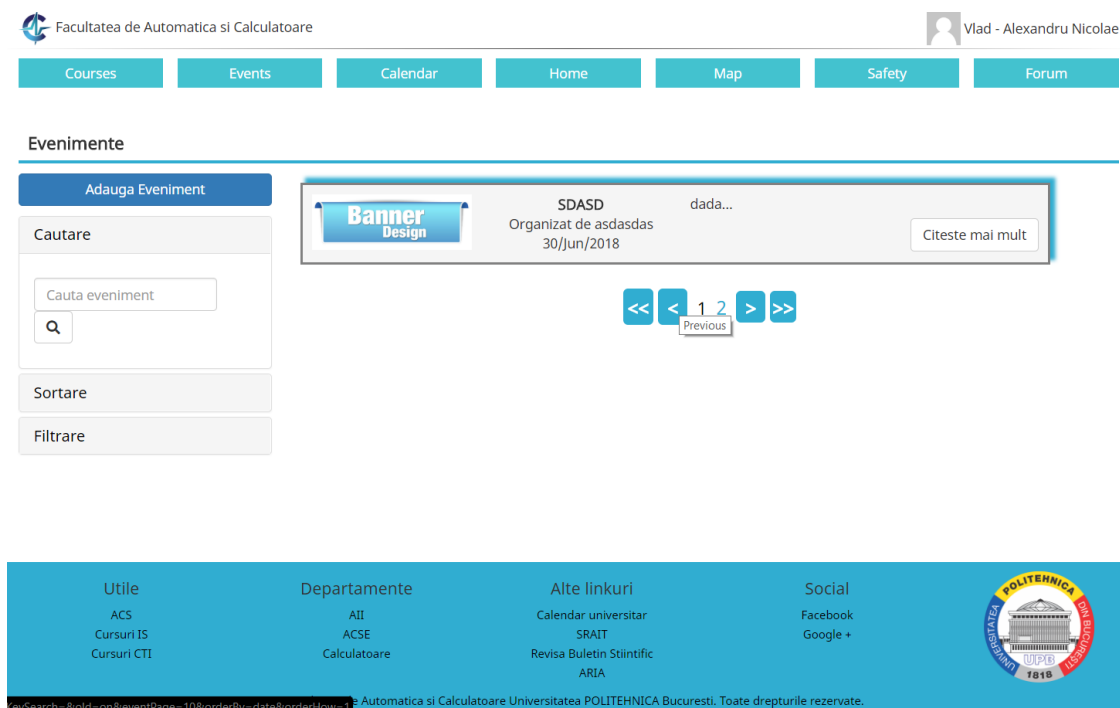


Fig. 5 – Pagina Events

Paginile cursuri și evenimente au o structură asemănătoare. În partea stânga a conținutului central se găsește o zonă de acțiuni. De aici, persoanele autorizate, vor putea adăuga cursuri/evenimente noi. Pe lângă această posibilitate, toți utilizatorii vor putea face căutări după un nume specific al unui curs/eveniment, vor putea alege metoda preferată de afișare a datelor (sortate după nume, data la care au fost postate pentru prima dată respectiv profesor/organizator ascendent sau descendent) și vor filtra rezultatele astfel încât să fie respectate anumite criterii. În imaginea alăturată se pot observa posibilitățile de filtrare pentru pagina Evenimente. Se vor putea realiza filtrări bazate pe existența unui cuvânt specific în interiorul datelor salvate în interiorul BD. O altă posibilitate este de a afișa doar evenimentele postate de utilizatorul curent oferind astfel un nivel de ușurință în găsirea și editarea evenimentelor proprii.

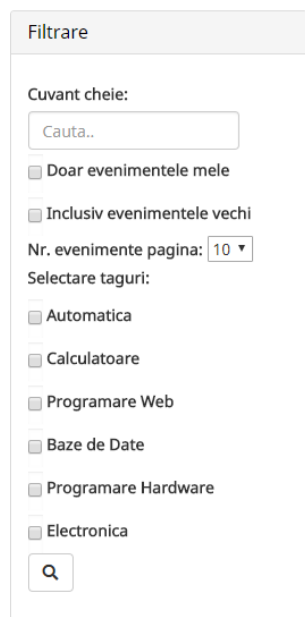


Fig. 6 – Modalitati de filtrare

Alte posibilități de filtrare includ: afișarea evenimentelor vechi, mai exact a evenimentelor

ce s-au desfășurat deja, schimbarea numărului de evenimente disponibile pe pagina (se va putea selecta una din valorile : 10, 20 sau 40) și vizualizarea selectivă după existența tagurilor reprezentative.

Partea dreaptă reprezintă zona de afișare efectivă. Pentru fiecare curs sau eveniment a fost formată o zonă specifică ce va conține următoarele:

- o imagine (acesta poate sa fie un logo al companiei organizatoare, o reprezentare grafica specifică pentru un anumit curs sau o imagine standard în situația în care nu s-a introdus altceva)
- date despre titlul cursului / evenimentului, numele profesorului ce va susține cursul respectiv sau a organizatorului evenimentului și data susținerii;
- o descriere succintă a cursului și a evenimentului
- posibilitatea de a citi toate informațiile oferite despre cursul / evenimentul prezentat

Așa cum am menționat mai sus, numărul de postări afișate pe o singura pagină este limitat, navigarea fiind realizată printr-o zonă de paginație, ce va permite selectarea rapidă a primei sau a ultimei pagini, trecerea de la o pagină la alta consecutiv sau în salturi prin selectarea unei pagini dorite.

Calendar

Calendar

LunăSăptămânăZi

iunie 2018

<>Az

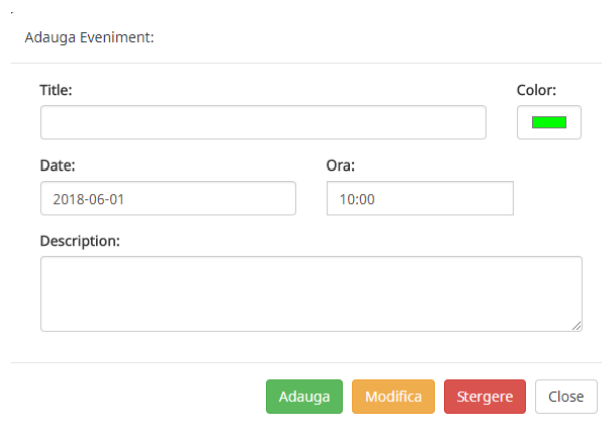
Lun	Mar	Mie	Joi	Vin	Sâm	Dum
28	29	30	31	1	2	
4	5 11 SSC	6	7	8	9	10
11	12	13	14 1 10	15 10 Test2	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1

Fig. 7 – Pagina Calendar

Pagina calendar este alcătuită în întregime din reprezentarea vizuală a unui calendar. Din partea stângă va putea fi selectată modalitatea de afișare a datelor, schimbând calendarul lunar într-o agendă săptămânală sau zilnică. În partea dreaptă se găsesc trei butoane necesare pentru a putea schimba lunile, săptămânile respectiv zilele.

Scopul acestei pagini este de a permite vizualizarea orarului și a altor evenimente personale adăugate de către un utilizator. Alături de această funcționalitate a fost adăugată și posibilitatea de a face modificări foarte rapid, doar prin selecția evenimentului și mutarea acestuia într-o altă zi sau într-un alt interval orar. Desigur, acest lucru poate fi realizat doar pentru evenimentele personale. Orarul nu va putea fi modificat decât de personalul autorizat, iar o astfel de modificare va genera o notificare prin email pentru toți participanții la cursul respectiv.

Pentru a adăuga un nou eveniment, tot ceea ce trebuie să facă un utilizator este să selecteze ziua dorită. În acel moment se va deschide o fereastră intitulată „adaugă eveniment”. De aici, se vor putea stabili titlul evenimentului, culoarea dorită pentru afișare, ora la care evenimentul se va desfășura, precum și o



Adauga Eveniment:

Title:

Color:

Date: Ora:

Description:

Adauga Modifica Stergere Close

descriere a acestuia. După apăsarea butonului

Fig. 8 – Adăugare Eveniment

„Adaugă”, noul eveniment va fi creat. Selectarea ulterioară a evenimentului va permite modificarea informațiilor cu unele mai actualizate sau ștergerea definitivă a acestuia.

Hartă

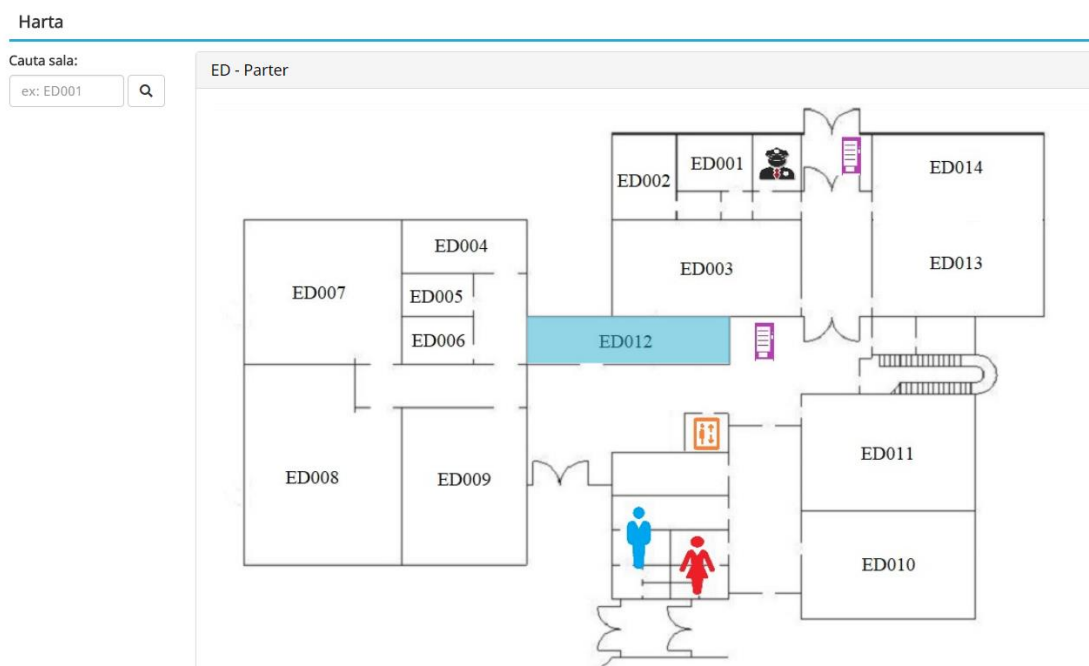


Fig. 9 – Pagina Map

Pagina Hartă este formată din două secțiuni. În partea stânga se va găsi un câmp de căutare, câmp ce poate fi utilizat pentru a afla mai multe informații despre o sală specifică. Tot în partea stânga, sub acest câmp vor fi afișate și informațiile suplimentare despre o anumită sală în momentul în care aceasta va fi selectată pe hartă. Aceste constau în denumirea sălii, rolul acesteia (pentru ce este utilizată), informații despre numărul de locuri sau echipamentul găsit în interiorul sălii.

În partea dreaptă se va găsi o secțiune în interiorul căreia sunt afișate planurile fiecărui etaj din interiorul facultății. Utilizatorul va putea comuta între acestea cu ușurință, doar prin selectarea numelui planului dorit. De asemenea acesta va avea posibilitatea de a utiliza planul respectiv drept hartă. În momentul în care acesta se va afla deasupra unei săli, acesta își va schimba culoarea, ușurând astfel procesul de identificare a sălii. Efectuarea unui click va genera automat afișarea informațiilor suplimentare.

În urma căutării unei săli specifice sau a apăsării butonului „Află mai mult”, buton generat în momentul afișării secțiunii de informații suplimentare, va fi deschisă o nouă pagină. În interiorul acesteia, utilizatorul va putea vizualiza toate informațiile deținute în baza de date despre sala respectivă, indicații vizuale pentru a putea găsi cu ușurință sala și detalii despre evenimentele ce se desfășoară în ziua curentă în sala respectivă.

Securitate

Facultatea de Automatica si Calculatoare

Vlad - Alexandru Nicolae

Courses Events Calendar Home Map Safety Forum

Siguranta

Cutremur
Incendii
Accident Nuclear sau chimic
Accidente Medicale
Altercatii si Violenta fizica

Date Generale

Reguli de comportare

- Păstrarea calmului, să nu se intre în panică și să liniștiți pe ceilalți colegi de grupa;
- Prevenirea tendințelor de a părăsi sala în care va aflați: putem fi surprinși de faza puternică a mișcării seismice în holuri, scări etc. Nu se folosește în nici un caz ascensorul;
- Dacă suntem în interiorul unei săli – rămânem acolo, departe de ferestre, care se pot sparge, să se stea înspre centrul locuinței (clădirii), lângă un perete. Protejarea se face sub o grindă, toc de ușă solid, birou, masă sau bancă din clasă suficient de rezistente spre a ne feri de căderea unor lămpi, obiecte mobile suprapuse, tencuieli ornamentale etc.;
- Dacă suntem surprinși în afara unei clădiri rămânem departe de aceasta, ne ferim de tencuieli, cărămizi, coșuri, parapete, cornișe, geamuri care de obicei se pot prăbuși în campus;
- Nu trebuie să: fugim la uși, sărim pe fereastră, aleargăm pe scări, utilizăm liftul, alergăm prin campus. Deplasarea se face cu calm spre un loc deschis și sigur;
- Dacă a trecut șocul puternic al seismului, se închid imediat sursele de foc cât se poate de repede, iar dacă a luat foc ceva se intervine imediat;
- Dacă seismul ne surprinde în autoturism, ne oprim cât se poate de repede într-un loc deschis, se evită clădirile prea aproape de stradă, dincolo de poduri, pasaje, linii electrice aeriene și ne ferim de firele de curent electric căzute;
- Dacă suntem într-un mijloc de transport în comun sau în tren, stăm pe locul nostru până se termina mișcarea seismică. Conducătorul trebuie să oprească și să deschidă ușile, dar nu este indicat să vă împulziți la coborâre sau să spargeți ferestrele. În metrou păstrați-vă calmul și ascultați recomandările personalului trenului, dacă acesta s-a oprit între stații în tunel, nu părăsiți vagoanele;
- Dacă vă aflați într-un loc public cu sală mare (de exemplu: teatru, cinematografe, stadioare, săli de sport etc.) nu...

Fig. 10 – Pagina Safety

Pagina de securitate reprezintă o pagina informativă, dedicată tuturor persoanelor din facultatea de Automatică și Calculatoare. De aici, se pot afla informații despre cutremure, incendii sau alte situații de criza ce pot afecta în orice moment siguranța persoanelor.

În partea stângă se găsește zona de navigare, iar în partea dreaptă va fi afișată zona de informare propriu zisă.

Forum

Titlul acestei secțiuni este destul de descriptiv. Pagina alcătuiește un forum dedicat studenților din facultate, oferindu-le posibilitatea de a pune și de a afla răspunsuri la diferite întrebări.

Inițial va fi deschisă zona de afișare a principalelor subiecte de comunicare. Așa cum se poate observa și în figura alăturată, un subiect va construi un bloc de date. Acest bloc de date va conține o siglă semnificativă, numele subiectului, numărul postărilor din interiorul acestui subiect, data la care a fost creat precum și

Courses Events Calendar Home Map Safety Forum

Forum

	Software Gratuit	2 Postare()	27/May/2018 creat de nicolae.vlad1994@gmail.com
Companii parteneri ale Universitatii Politehnice Bucuresti			
	ERASMUS	0 Postare()	12/Jun/2018 creat de nicolae.vlad1994@gmail.com
Informatii despre programul ERASMUS			

Adauga Topic

Fig. 11 – Vizualizare Topice

numele utilizatorului ce a adăugat acel subiect și nu în ultimul rând o scurtă descriere a tipurilor de postări ce vor găsi în interiorul subiectului. În partea de jos se poate observa și butonul de adăugare a unui nou topic de discuție.

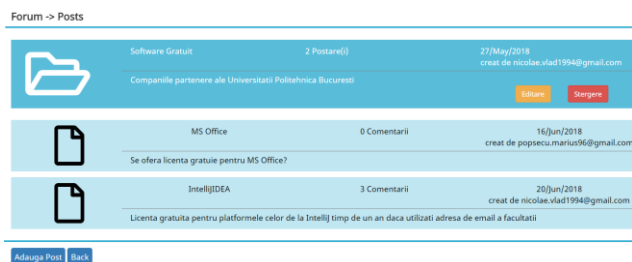


Fig. 12 – Vizualizare postări

Prin apăsarea pe un topic, utilizatorul va fi redirectat către zona de postări. Aceasta este destul de asemănătoare cu zona topic, principala diferență fiind: în prima parte a pagini se vor putea observa aceleași informații ce au fost

prezentate în pagina anterioară și butoanele 'Editare' și 'Ștergere', butoane ale căror acțiuni sunt deja cunoscute. Mai jos, se vor putea vedea blocurile de date ce vor conține informații despre postări și butoanele de adăugare a unei noi postări și de returnare la pagina anterioară.

Ultima secțiune a forumului va fi reprezentată de vizualizarea răspunsurilor din interiorul unei postări. În partea superioară se poate observa zona de descriere a postării respective, împreună cu butoanele de modificare a conținutului acesteia sau ștergere definitivă. Sub această zonă, sunt afișate comentariile studenților cu privire la tema discutată. Acestea vor fi alcătuite din informații despre cel ce a postat comentariul respectiv, informația ce s-a dorit să fie transmisă, momentul postării comentariului respectiv și butoanele ce vor oferi control asupra comentariului respectiv. Sub zona de afișare a comentariilor va fi afișat formularul ce va permite introducerea unui nou răspuns și butonul de returnare la pagina anterioară.

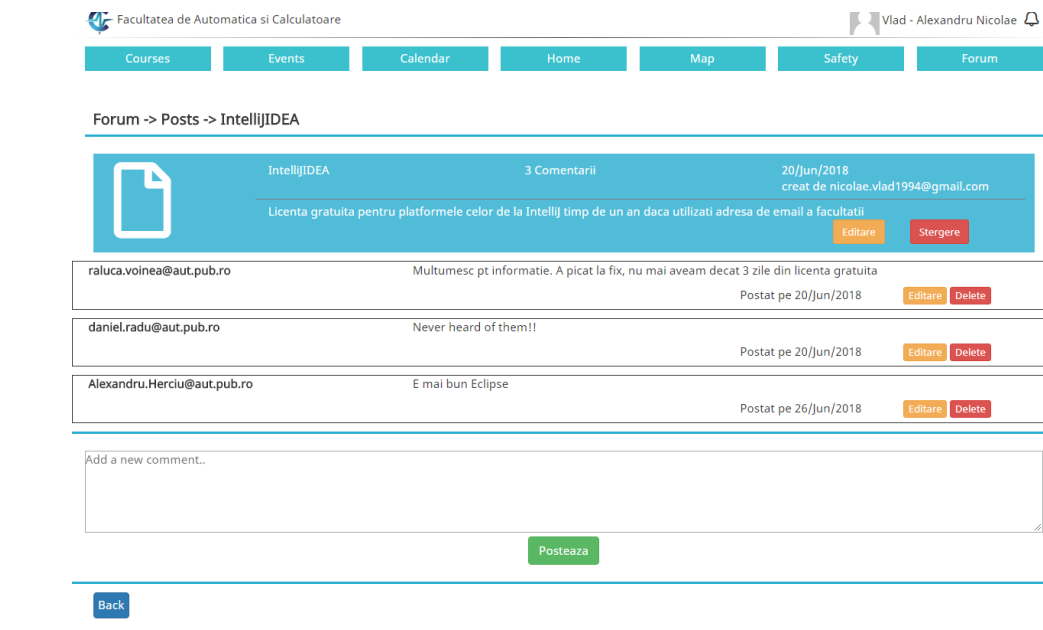


Fig. 13 – Vizualizare conținut postare

Implementarea aplicației

Primul pas pentru realizarea acestei etape a fost alegerea unui mediu integrat de dezvoltare (Integrated development environment, cel mai adesea cunoscut sub prescurtarea de IDE). Un IDE reprezintă o aplicație software a cărui scop este de a ajuta programatorii în scrierea programelor. Un mediu de dezvoltare va combina toți pașii necesari creării unui program, editare cod sursa, compilare, depanare, testare, într-un singur soft, ce oferă în același timp și o interfață prietenoasă.

Pentru dezvoltarea aplicației MyACS am ales să utilizez mediul de dezvoltare produs de compania ceha JetBrains, WebStorm. WebStorm este mediu foarte puternic, perfect echipat pentru dezvoltarea de aplicații complexe pentru partea clientului utilizând JavaScript, dar și pentru aplicațiile din partea serverelor, utilizând Node.js. WebStorm oferă suport pentru HTML, CSS, JavaScript și alternativele lor mai moderne, și utilizează unelte de dezvoltare cum ar fi: editor inteligent pentru oferirea de asistență pentru JavaScript, Node.js și ECMAScript6 dar și un debugger integrat pentru aplicațiile realizate cu Node.js.

MVC

Următorul pas a fost alegerea unui model arhitectural de dezvoltare. În această privință am ales MVC (Model View Control, prescurtare engleza MVC), acesta reprezentând un model arhitectural ce constă în separarea aplicației în trei componente logice principale: modelul, vizualizarea și controlorul. Fiecare dintre aceste componente au fost formate pentru a se ocupa de aspecte specifice ale dezvoltării aplicației. Am ales să utilizez un astfel de model deoarece, MVC reprezintă standardul cel mai utilizat în dezvoltarea aplicațiilor web, oferind scalabilitate aplicațiilor și posibilitatea ca modificările realizate în fișierele din zona de vizualizare să nu aducă schimbări într-o altă zonă. Astfel se garantează faptul că restul aplicației va rula corespunzător.

Componentele modelului MVC sunt:

- Model: Această componentă conține toate informațiile ce conțin logica datelor cu care utilizatorul va lucra. Acestea pot fi date ce sunt transferate între Controlor și

Vizualizator sau orice alte date, cum ar fi un obiect Utilizator va prelua informațiile despre utilizatori din baza de date, le va manipula, actualiza și salva în baza de date sau le va afișa în aplicație;

- Vizualizare: Aici vor fi stocate toate datele ce vor fi utilizate pentru logica interfeței cu utilizatorul. De exemplu, vizualizarea informațiilor despre Utilizatori vor conține zone text, imagini, check box-uri, etc. – toate aceste reprezentând elemente cu care utilizatorii vor interacționa;
- Controlor: Această componentă acționează ca o interfață de legătură între componentele menționate mai sus. Rolul acesteia este a procesa cererile efectuate , de a manipula datele utilizând componenta model și de a interacționa cu componenta vizualizare pentru a oferi un răspuns final. De exemplu, controlorul Utilizator va primi o serie de date de intrare de pe pagina de adăugare utilizatori, acesta va utiliza acele date pentru a crea un nou utilizator ce va fi salvat în baza de date, respectând modelul Utilizator din componenta model, urmând ca în final să încarce o noua pagina din Vizualizator.

În imaginea alăturată se poate observa structura întregului proiect. Analizând fiecare document în parte, vom putea distinge:

- Middleware: Conține funcții suplimentare ce sunt rulate în controlor. Acestea vor fi apelate imediat ce a fost realizată o cerere, înainte ca operațiile standard, pentru realizarea răspunsului, să fie efectuate.
- Models: Componenta Model;
- Node_modules: Conține toate elementele ce au fost instalate prin utilizarea funcției specifice Node.Js, 'npm'.

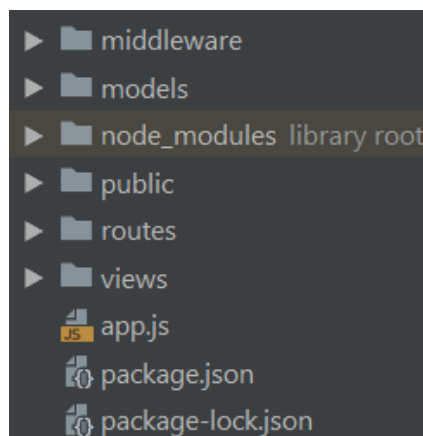


Fig. 14 – Vizualizare conținut proiect

- Public: Aici am salvat foile de stiluri, scripturile javascript precum și alte informații vizuale utilizate în aplicație;
- Routes: Componenta Controlor;
- Views: Componenta Vizualizator;

- App.js: Componenta principală a aplicației, aceasta făcând legătura dintre toate celelalte elemente.
- Package.json: Acest document reprezintă nucleul ecosistemului Node.js. Fișierul este utilizat ca și un manifest în care sunt menționate pachetele și modulele utilizate de aplicația realizată în Node.

Implementarea Bazei de Date

Realizând o analiză a datelor ce urmează să fie afișate în platforma MyACS, am identificat o serie de colecții, unele individuale, iar altele cu relații de tip unu la mai mulți. Mai jos am atașat o reprezentare schematică a acestor colecții. Se poate observa cu ușurință importanța componentei „Utilizator”, aceasta aflându-se într-o relație cu aproape fiecare colecție.

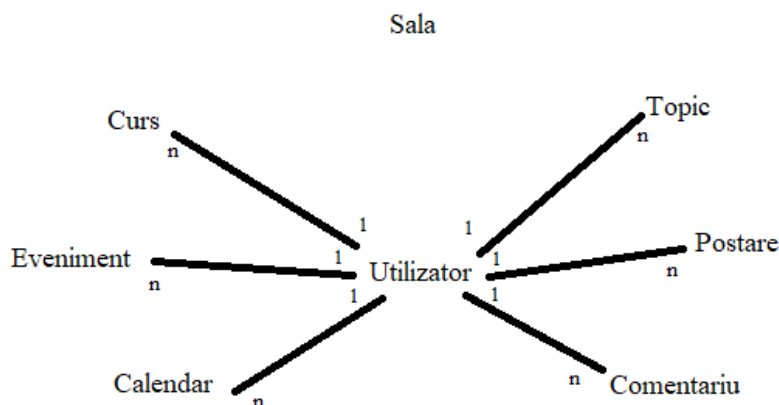


Fig. 15 – Vizualizare schematică a colecțiilor

Pentru a realiza legătura dintre aplicație și baza de date MongoDB am utilizat pachetul Mongoose. Acesta oferă o soluție simplă, bazată pe o schemă pentru a ne permite să modelăm datele din aplicație. Pachetul include casting încorporate, validare, formulare de interogări și multe alte unelte ce ne permit să construim logica aplicației;

De fiecare dată când vom utiliza mongoose, vom porni de la formularea unei scheme. Fiecare schemă va reprezenta modelul unei colecții din MongoDB și va defini

forma documentului din interiorul acelei colecții. Pornind de la schema de mai sus, am separat fiecare colecție și am identificat principalele informații ce vor urma a fi stocate în ele astfel:

- Utilizator: deoarece în interiorul aplicației MyACS există o separare a rolurilor deținute de utilizatori, aceasta colecție este de departe cea mai complexă. Alături de proprietățile standard, ce sunt generate automat de pachetul Mongoose: `_,v'` și `_,id'`, am mai adăugat:

- Username, password, resetPasswordToken și resetPasswordExpire – toate aceste informații fiind necesare pentru realizarea sistemului de autentificare;
- Type și o serie de vectori ce conțin numeroase alte proprietăți: `personal_info`, `documents`, `home` și `family` – aceste date au fost utilizate pentru a îmbogăți colecția de date despre fiecare utilizator, date ce vor putea fi utilizate ulterior în alte documente sau în realizarea de sondaje.

- Vectorii `student_info` și `employee_info` vor conține informații specifice, ce sunt diferite în funcție de tipul de utilizator ce a fost ales în momentul formării unui nou cont. O

particularitate a lui MongoDB este aceea ca un utilizator nu trebuie să conțină toate câmpurile ce sunt introduse în interiorul schemei. Astfel un student nu va avea niciodată informații despre data de angajare, poziția ocupată în facultate sau departamentul pentru care lucrează; respectiv, un angajat nu va avea informații legate de numărul matricol, volumul matricol, metoda de finanțare,

etc. Acest lucru mi-a permis utilizarea unei singuri colecții pentru utilizatori, evitând situațiile în care fiecare tip de utilizator avea o colecție

```

    _id: ObjectId("5aba595af58aa5272c138b6d2")
    > personal_info: Object
    > documents: Object
    > family: Object
    > student_info: Object
      username: "nico15e-v1ad1994@gmail.com"
      type: "Student"
      salt: "79d3cfed37c27f993422da345925370d83795f5811d29610a0b42e0f806d5a65"
      hash: "912b7226b0b0ff293d1a484ec70ec38996284bfcfc0b5e2e09c862cda34e138651886c..."
      _v: 121
    > home: Object
    > events: Array
    > courses: Array
    > grades: Object
    > calendar: Array
      resetPasswordExpires: 2018-06-06 20:10:31.250
      resetPasswordToken: "78955d7eb3f56233f552a2cc4f9812744f7967f"
    > topics: Array
    > posts: Array
    > comments: Array

    _id: ObjectId("5b2e65fccc002104ce790a1")
    > personal_info: Object
    > documents: Object
    > home: Object
    > family: Object
    > employee_info: Object
    > grades: Object
    > events: Array
    > calendar: Array
    > courses: Array
    > topics: Array
    > posts: Array
    > comments: Array
      username: "test@test.com"
      type: "Professor"
      salt: "5f01a150a310e3472a07e165e7b9fd048e78828b3cea700ada4b0ba475cf3cf2"
      hash: "ec94528072a827aa46d5fa62e77a2ed9e9e82b376f9c90fe081bc8507ca5cf48c0b..."
      _v: 0

```

Fig. 16 – Vizualizare unei secțiuni din colecția utilizatori folosind aplicația MongoDBCompass

corespunzătoare și multe câmpuri se repetau sau o singura colecție ce urma să conțină elemente nule. În partea stângă se poate observa clar acest lucru (pentru vizualizarea datelor din interiorul bazei de date am utilizat aplicația MongoDBCompass).

- Events, calendar, courses, grades, topics, posts și comments sunt vectorii ale căror elemente vor fi reprezentate de obiecte. În interiorul acestor obiecte, se vor salva id-urile unice al unui element corespunzător dintr-o colecție cu același nume.
- Topic, Postare și Comentariu: acestea reprezintă colecțiile utilizate pentru realizarea forumului studențesc. În interiorul lor se vor găsi date standard despre data postării materialului respectiv, numele și id-ul unic al celui ce a realizat aceasta operațiune precum și o descriere a conținutului;
- Curs: în interiorul acestei colecții au fost salvate informații despre deținătorul cursului respectiv și alte date informative cum ar fi: tipul cursului, costul, descrierea, cuvinte reprezentative, etc;
- Calendar: utilizat pentru salvarea evenimentelor personale sau a celor preluate automat din orar și vizualizarea acestora;
- Evenimente: o colecție asemănătoare cu cea realizată pentru cursuri, de această dată destinată evenimentelor ce urmează să se desfășoare în facultate. Se vor găsi date despre autor, organizator, data desfășurării evenimentului, descrierea evenimentului și cuvinte ce pot fi utilizate cel mai ușor pentru a rezuma activitatea ce urmează să fie efectuată;
- Sala: ultima colecție realizată în interiorul proiectului, aceasta va conține informații descriptive despre sălile din interiorul facultății.

Următorul pas a fost scrierea efectivă a unei scheme pentru fiecare colecție în parte. O schemă se poate scrie sub forma unui obiect javascript, în interiorul căruia vom specifica numele proprietății pe care dorim să o atașăm și al tipului de date ce se dorește a fi introdusă, de exemplu: string, integer, date, array sau un alt obiect. Acest lucru poate fi observat și în exemplul de mai jos.

```

var eventSchema = new mongoose.Schema({
  author: {
    id: {
      type: mongoose.Schema.Types.ObjectId,
      ref: "User"
    },
    username: String
  },
  name: String,
  date: Date,
  created: Date,
  organizer: String,
  description: String,
  tags: []
});

```

Sistemul de login

Pentru realizarea sistemului de login am utilizat pachetul passport. Acest pachet a fost proiectat pentru a se ocupa de cererile de autentificare și poate oferi diferite strategii de autentificare, incluzând cele bazate pe rețele sociale.

Strategia pe care am decis să o utilizez este una clasică, bazată pe email oferit de facultate urmat de o parola inițială alcătuită din cuvântul ,autcalc' urmat de ultimele 6 cifre din codul numeric personal. Pentru a realiza acest lucru am început prin instalarea pachetului și cererea tuturor dependențelor în interiorul fișierului app.js

```

Var passport = require("passport"),
    LocalStrategy = require("passport-local").Strategy,
    passportLocalMongoose = require("passport-local-mongoose");
app.use(require("express-session")({
  secret: "Sic Parvis Magna",
  resave: false,
  saveUninitialized: false
}));

```

De asemenea am utilizat pachetul express-session pentru a putea stoca sesiunile. Următorul pas a fost utilizarea modului passport-local pentru a integra sistemul de autentificare bazat pe nume utilizator și parolă.

```
app.use(passport.initialize());
app.use(passport.session());
app.use(function (req, res, next) {
  res.locals.currentUser = req.user;
  res.locals.error = req.flash("error");
  res.locals.success = req.flash("success");
  next();
});

passport.use(new LocalStrategy(User.authenticate()));
passport.serializeUser(User.serializeUser());
passport.deserializeUser(User.deserializeUser());
```

După ce am realizat aceste setări, sistemul de login poate fi utilizat. Acesta va prelua numele utilizatorului introdus și va efectua o căutare în baza de date pentru a vedea dacă acesta există. În cazul pozitiv, asupra parolei introduse se va aplica aceeași metoda de hash ce a fost aplicată asupra parolei ce se găsește stocată în baza de date și se va realiza o comparație pentru a vedea dacă acestea se potrivesc. Dacă se va trece și de aceasta etapă, atunci se va trece automat la redirectionarea către pagina principală a aplicației. În caz contrar, utilizator va fi atenționat de faptul că a apărut o eroare în timpul procesului de autentificare.

Așa cum am menționat mai sus, parola inițială este o parola generată automat și nu se poate considera foarte sigură. Din acest motiv este recomandat ca aceasta să fie schimbată imediat ce contul unui utilizator a fost creat. De asemenea, există posibilitatea ca utilizatorul să uite parola setată. Astfel, am conceput un sistem de recuperare/schimbare ce se bazează pe adresa de mail a utilizatorului. În cazul în care se înregistrează o cerere de schimbare a parolei, aplicația va trimite un mail utilizatorului cu un link activ pentru o perioadă limitată de timp. Accesând linkul, utilizatorul va putea să reintroducă noua parola.

Pentru a realiza acest lucru am utilizat pachetul async-waterfall ce îmi permite să rulez un vector de funcții, rezultatul funcției precedente fiind utilizat drept intrare pentru funcția actuală. De asemenea dacă este înregistrată o eroare în timpul rulării unei funcții, se va opri întreaga execuție a secvenței de funcții.


```

crypto.randomBytes(20, function (err,buf) {
    var token = buf.toString('hex');
    done(err, token);
});

```

Prima secvență utilizează pachetul crypto pentru a realiza criptarea unui șir aleator de 20 de biți. Variabila token va stoca această valoare după ce va realiza o conversie hexazecimala.

```

User.findOne({username: req.body.email }, function (err, user) {
    if(err){
        console.log(err);
    } else {
        if(!user) {
            req.flash("error", "Nu exista un cont cu
ce utilizeaza aceasta adresa de email.");
            return res.redirect("/forgot");
        }
        user.resetPasswordToken = token;
        user.resetPasswordExpires = Date.now() +
3600000; //o ora

        user.save(function (err) {
            done(err, token, user);
        });
    }
});

```

Următoarea secvență verifică dacă există adresa de email introdusă este una validă, salvează token-ul format în secvența anterioară și setează o limită egală cu o oră, în care operațiunea de resetare a parolei este validă.

```

var smtpTransport = nodemailer.createTransport({
  service: 'Gmail',
  auth: {
    user: 'vlad.licenta@gmail.com',
    pass: 'ACS#licenta'
  }
});
var mailOption = {
  to: user.username,
  from: "vald.licenta@gmail.com",
  subject: "Password Reset",
  html:
    "<p>You are receiving this because you (or  

    someone else) have requested the reset of the password for your  

    account.</p>" +
    "<p>Please click on the following link, or paste  

    this into your browser to complete the process: " +
    "http://" + req.headers.host + "/reset/" + token
    +
    "</p><p>If you did not request this, please  

    ignore this email and your password will remain unchanged.</p>"
  };
  smtpTransport.sendMail(mailOption, function (err)
  {done(err, "done");});

```

Ultima secvență este responsabilă pentru trimiterea unui mesaj pe adresa de email oferită. Aici sunt introduse informațiile adresei de email utilizată de aplicație, precum și subiectul și mesajul emailului ce urmează a fi trimis. Pentru a realiza acest lucru am utilizat pachetul nodemailer, pachet ce a fost conceput special pentru realizarea unor astfel de operațiuni.

Următorul pas trebuie efectuat de către utilizator, mai exact să se autentifice în aplicația de email utilizată și să acceseze linkul oferit de către aplicația MyACS. După ce va face acest lucru, va fi redirecționat către adresa aplicației MyACS și i se va permite să reintroducă noua parolă. În momentul în care acest lucru este realizat, primul pas pe care aplicația îl va urma va fi să verifice dacă data de expirare a tokenului nu a fost atinsă. Dacă perioada depășește o ora, atunci tokenul va fi șters din baza de date, iar parola va rămâne neschimbată. Dacă perioada este mai mică de o ora, se va compara dacă valorile din câmpurile „parolă nouă” și „confirmă parola nouă” sunt identice, și se va salva noua parolă în cazul pozitiv.

Securitatea aplicației

Pentru a mă asigura că procesul de autentificare nu este evitat prin introducerea unei adrese cunoscute, de ex: index/events, moment în care node va primi cererea și va realiza toate operațiile ce urmează după aceasta cerere, am adăugat un middleware. Middleware, reprezintă funcții ce sunt rulate între începutul operațiunii de rutare și trimiterea răspunsului către utilizator. Soluția pentru problema descrisă mai sus este:

```
middlewareObj.isLoggedIn = function (req,res,next) {  
    if(req.isAuthenticated()){  
        return next();  
    }  
    req.flash("error", "You need to be logged in to do  
that");  
    res.redirect("/login");  
};
```

Această funcție va fi rulată de fiecare dată când se va realiza o cerere și va verifica dacă operația de autentificare a fost efectuată. În situația în care răspunsul este negativ, se va face trimiterea automată către pagina de autentificare.

O alta măsură de protecție pe care mi-am dorit să o implementez a fost cea de evitare a injecției. Injecția de cod reprezintă exploatarea unui bug de calculator ce este datorată procesării unor date invalide. Acest lucru se poate realiza în momentul în care un atacator va introduce în interiorul unui câmp de date cod. În momentul în care se va realiza salvarea acelor informații în baza de date, codul introdus va fi recunoscut și rulat. Pentru a evita acest lucru, este necesară codificarea informației trimise către server.

În întâmpinarea acestei metode, am utilizat modulul sanitizer.

Route

Acronimul CRUD (Create Read Update and Delete – în română: Creare Citirea Actualizare și Ștergere) este foarte cunoscut în programare prin reprezentarea sa a celor patru funcții ce stau la baza stocării persistente a informațiilor. Aceste funcții au fost implementate în aplicațiile ce utilizează baze de date relaționale, fiecare literă din interiorul

acronimului făcând referire la o afirmație din limbajul SQL, dar și în HTTP și operațiile de servicii distribuite de date.

Transferul de stat reprezentativ (Representational State Transfer – prescurtare utilizată: REST) reprezintă un stil arhitectural pentru a defini un set de constrângeri și proprietăți bazate pe serviciul HTTP. Serviciile web ce respectă acest tip de arhitectură oferă interoperabilitatea dintre sistemele de tip calculator și internet. Serviciile web REST permit accesarea și manipularea reprezentării textuale a resurselor web prin utilizarea unor operații fără stare (stateless) uniforme și predefinite.

Acest tip de tehnologie este preferată peste mult mai robustul Protocol de acces al obiectelor simple (SOAP) deoarece utilizează o lățime de bandă mai mică, fiind astfel mult mai potrivită pentru utilizare folosind internetul. REST utilizează în mod explicit metodologia HTTP definită de protocolul RFC 2616, utilizând: GET pentru a prelua resurse, PUT pentru a schimba o stare sau pentru a realiza update-ul unei resurse, POST pentru a crea aceea resursă și DELETE pentru a o șterge.

Aplicația MyACS a fost construită utilizând aceste noțiuni, aproape fiecare colecție din baza de date are aceste posibilități. Tabelul de mai jos descrie mai detaliat procesul necesar implementării acestei arhitecturi.

Nume	Rută	Verbul HTTP Utilizat	Scop	Metodologia mongoose utilizată
Index	/event	GET	Listarea tuturor evenimentelor	Event.find()
New	/event/new	GET	Vizualizarea formularului pentru introducerea unui nou eveniment în baza de date	N/A
Create	/event	POST	Crearea unui nou eveniment	Event.create()
Show	/event/:id	GET	Vizualizarea informațiilor specifice ale unui eveniment	Event.findById()
Edit	/event/:id/edit	GET	Afișarea formularului pentru editarea unui	Event.findById()

			eveniment deja existent	
Update	/event/:id	PUT	Realizarea operațiuni de actualizare a informațiilor unui eveniment	Event.findByIdAndUpdate()
Destroy	/event/:id	DELETE	Ștergerea unui eveniment din baza de date	Event.findByIdAndRemove()

Tab. 1 – Descrierea notiuni REST

Punând aceste informații în practică, un fișier din folderul „routes”, ce reprezintă componenta de control pentru acțiunile ce privesc o anumită colecție din baza de date va avea următoarea structură:

```
var express    = require("express"),
    router     = express.Router(),
    middleware = require("../middleware");

var User = require("../models/user"),
    Event = require("../models/event");
```

Zona de apelare a modulelor ce sunt necesare pentru ca script-ul să ruleze în mod corespunzător dar și a colecțiilor ce vor fi utilizate.

```
//INDEX
router.get("/", middleware.isLoggedIn, function (req, res) {
  Event.find({}, function (err, allEvents) {
    if(err) throw err;
    res.render("event/index", {event: allEvents});
  });
});
```

Codul de mai sus corespunde zonei index, zonă ce va fi responsabilă de afișarea tuturor evenimentelor existente în baza de date. Structura codului este următoarea: metoda utilizată de cererea http, ruta, lansarea în execuție a funcției isLoggedIn, utilizată pentru a verifica dacă adresa respectivă este accesată de un utilizator ce s-a conectat la platforma web și o funcție.

În interiorul funcției, se va realiza o căutare generală a tuturor evenimentelor, se vor trata erorile în situația în care acestea există și se va lansa în execuție vizualizarea paginii de index a evenimentelor.

```
router.get("/new", middleware.isLoggedIn, function (req,res) {  
    res.render("event/new");  
});
```

Pentru secvența new nu este necesar decât să se lanseze în execuție vizualizarea paginii new, pagină ce va conține formularul necesar pentru introducerea unui nou element în interiorul colecției.

```
router.post("/", middleware.isLoggedIn, function (req,res) {  
    User.findById(req.user._id, function (err, user) {  
        if(err){  
            console.log(err);  
            req.flash("error", "Ceva a mers prost!");  
            res.redirect("/index/events");  
        } else {  
            Event.create(req.body.event, function (err, event) {  
                event.author.id = req.user._id;  
                event.author.username = req.user.username;  
                event.save();  
  
                user.events.push(event._id);  
                user.save();  
  
                req.flash("success", "Eventul a fost adaugat cu  
succes");  
  
                res.redirect("/index/events/");  
            });  
        }  
    });  
});
```

Pentru a crea un nou eveniment, pentru început va trebui să căutam userul actual. Dacă acesta va fi găsit și nu întâmpinăm nicio eroare atunci vom crea noul eveniment utilizând informațiile trimise prin formularul de pe pagina new. Deoarece între utilizator și evenimente există o legătură unu la mai mulți, în interiorul colecției events vom salva id-ul autorului precum și numele de utilizator, iar în interiorul colecției user vom adăuga în vectorul events id-ul noului eveniment.

```

//SHOW
router.get("/:id", middleware.isLoggedIn, function (req,res) {
    Event.findById(req.params.id, function (err, foundEvent) {
        if(err) throw err;
        res.render("event/show", {event: foundEvent});
    });
});

//EDIT
router.get("/:id/edit", middleware.isLoggedIn, function
(req,res) {
    Event.findById(req.params.id, function (err, foundEvent) {
        if(err) throw err;
        res.render("event/edit", {event: foundEvent});
    });
});

```

Pentru secvențele show și edit se va realiza o căutare după un id specific în colecția Event, iar în situația în care nu este întâmpinată nicio problemă atunci vom lansa în execuție vizualizarea paginii show, respectiv edit. În ambele situații vom trimite sub forma unui obiect toate informațiile despre evenimentul găsit.

```

//UPDATE
router.put("/:id", middleware.isLoggedIn, function (req,res)
{
    Event.findByIdAndUpdate(req.params.id, req.body.event,
function (err, foundEvent) {
    if(err) res.redirect("back");
    res.redirect("/index/events/"+foundEvent._id);
});
});

```

Pentru a realiza actualizarea informațiilor dintr-o colecție, vom efectua o căutare după id-ul evenimentului pe care dorim să-l modificăm și vom realiza operația de actualizare pe baza noilor informații primite de la formularul existent pe pagina edit.

```
//DELETE
router.delete("/:id", middleware.isLoggedIn, function
(req,res) {
    Event.findByIdAndRemove(req.params.id, function (err) {
        if(err) res.redirect("back");
        req.flash("success", "Eveniment Sters");
        res.redirect("/index/events");
    });
});

module.exports = router;
```

Ștergere informațiilor dintr-o baza de date se va realiza într-o manieră asemănătoare cu cea menționată mai sus, singura excepție fiind că vom apela funcția `findByIdAndRemove` în loc de `findByIdAndUpdate`. După ce vom finaliza ștergerea evenimentului, vom fi redirecționați către pagina index a evenimentelor.

Ultima linie de cod este utilizată pentru a spune ce va returna scriptul `event.js` din interiorul componentei de control.

Filtrarea datelor

După ce am realizat implementarea de tip REST a tuturor colecțiilor, am început să adaug funcționalități suplimentare. În cazul paginilor de vizualizare a evenimentelor și a cursurilor, acestea constau în oferirea posibilității de a realiza filtrarea datelor.

Acestea au fost implementate prin adăugarea în pagina de vizualizare a unor noi formulare, formulare în care un utilizator va menționa ce tip de filtrare dorește să realizeze. Posibilitățile oferite constau în căutarea după titlu, căutarea unui cuvânt sau serii de cuvinte specifice, sortări ascendente sau descendente utilizând anumite câmpuri, căutarea unor taguri specifice sau afișarea doar a acelor cursuri sau evenimente ce au fost create de utilizatorul curent.

După ce informațiile vor fi trimise în partea de server, pe baza acestora se vor realiza căutări în baza de date și trimiterea rezultatelor găsite înapoi spre pagina de

vizualizare. De asemenea, mi-am dorit ca utilizatorul să aibă posibilitatea de a realiza și combinații de metode de filtrare, astfel, tot în partea de server vor fi trimise și solicitările anterioare pentru a putea fi refolosite mai târziu. Metoda aplicată pentru prelucrarea preferințelor studenților, profesorilor sau a altor persoane autorizate a fost de formare a unui obiect. Acest obiect are o structură asemănătoare cu cea a schemei colecției, denumirile proprietăților fiind identice. Ulterior, am introdus acel obiect în interiorul funcției fiind () din Mongoose, funcție ce va căuta în colecție toate înregistrările ce respectă criteriile introduse în obiect.

Sortarea a fost realizată prin introducerea unui alt parametru în interiorul funcției fiind (), acest parametru fiind reprezentat tot de un obiect de tip javascript.

Calendar

Pentru a implementa funcționalitățile calendarului am pornit de la utilizarea atașamentului „fullcalendar”. Acesta oferă posibilitatea dezvoltatorului de a personaliza într-o manieră simplificată felul în care calendarul va arăta. Aceste opțiuni variind de la limba utilizată, setarea zilei din săptămâna ce va apărea prima în calendar, zona orară aleasă până la ce modalități de vizualizare dorim să selectăm. Toate aceste lucruri vor fi introduse într-o funcție ce va avea formatul următor:

```
var calendar = $(this);
calendar.fullCalendar({
  header: {
    left: 'month,agendaWeek,agendaDay',
    center: 'title',
    right: 'prev,next today'
  },
  firstDay: 1,
  locale: 'ro',
  firstHour: '7',
  timezone: 'local',
  defaultView: 'month',
  selectable: true,
  selectHelper: true,
  events: '../..//calendar.json',
  editable: true,
  eventLimit: true,
  eventOverlap: false
```

O componenta principală a paginii „Calendar” este aceea că: utilizatorul va avea posibilitatea de a vizualiza evenimente la care a participat sau ce urmează să se desfășoare. Acestea pot fi orele de curs ale unui student sau ale unui profesor din orar, sau alte evenimente personale formate de aceștia. Evenimentele vor fi citite dintr-un document json, document ce va fi format și reactualizat în timpul procesării cererii de acces a paginii „calendar”.

Și de această dată am implementat cele 4 funcționalități CRUD. Astfel, pentru a forma un nou eveniment, în momentul în care utilizatorul va selecta o zi din calendar se va apela o funcție afișată mai jos. În interiorul ei se vor realiza modificări pentru anumite câmpuri ale unui formular ce va fi deschis în pagina de vizualizare.

```
dayClick:function (date, jsEvent, view) {  
    $('#eventTitle').html("Adauga Eveniment: ");  
  
    $('#calTitle').val("");  
    $('#calDesc').val("");  
    $('#calDate').val(date.format());  
    $('#calCul').val("#00FF00");  
    $('#formCalendar').modal();  
},
```

După ce se vor completa datele și se va apăsa butonul „Adauga Eveniment”, se va realiza apelul unei funcții. Această funcție are rolul de a forma obiectul ce urmează să fie trimis către server prin intermediul unei cereri de tip Ajax. Ajax este o tehnologie ce permite formarea de aplicații web asincrone. Acest lucru înseamnă că se vor putea trimite și primi date către/de la server fără a fi necesară reîncărcarea paginii web, așa cum se întâmplă de obicei.

```
$.ajax({  
    type: 'POST',  
    url: '/index/calendar',  
    data: objEvent,  
    success:function (msg) {  
        if(msg){  
            $('#calendar').fullCalendar('refetchEvents');  
        }  
    }  
});
```

Noul obiect va fi trimis către server, iar informațiile vor fi salvate în interiorul bazei de date. Următorul pas este reformarea obiectului calendar.json și întoarcerea către pagina de vizualizare. Dacă cererea ajax a fost realizată cu succes, atunci va avea loc o reactualizare a evenimentelor, această activitate fiind efectuată prin recitirea documentului calendar.json.

Modificarea unui eveniment sau ștergerea acestuia se va realiza în mod asemănător, singura excepție ce poate fi remarcată fiind: se vor prelua informațiile existente în baza de date despre evenimentul selectat și acestea vor fi utilizate pentru modificarea câmpurilor afișate în interiorul formularului.

Pentru a simplifica modalitatea de modificare a evenimentelor, am adăugat și posibilitatea ca modificările temporale să se poată realiza mult mai rapid. Tot ceea ce un utilizator va trebui să facă este să selecteze evenimentul dorit și să îl tragă în căsuța corespunzătoare unei alte zile. Acest tip de funcționalitate a fost posibilă prin adăugarea următoarei funcții:

```
eventDrop: function (callEvent) {
    $('#calID').val(callEvent._id);
    $('#calTitle').val(callEvent.title);
    $('#calDesc').val(callEvent.description);
    var Date = callEvent.start.format().split("T");
    console.log(Date);
    $('#calDate').val(Date[0]);
    $('#calOra').val(Date[1]);
    $('#calCul').val(callEvent.color);
    GUI();
    console.log(NewEvent);
    editInformation(NewEvent, true);
},
```

Deoarece corectitudinea datelor ce reprezentau evenimentele din orarul universitar trebuiau să fie garantate, în partea de server am adăugat un control suplimentar pentru persoana ce dorește să realizeze modificări pentru acele evenimente. În situația în care persoana nu este una autorizată, modificările nu vor fi efectuate. Dacă persoana este autorizată, atunci acestea vor avea loc și adițional se va realiza și o notificare a tuturor persoanelor ce desfășurau cursul/laboratorul respectiv cu privire la modificările ce au fost efectuate, utilizând adresa de email personală.

Harta

Pentru dezvoltarea hărții, am pornit de la studierea planurilor facultății. Apoi, utilizând aplicația AutoCAD am realizat un plan bidimensional al fiecărui etaj din facultate, încercând să respect pe cât de mult posibil proporțiile reale. Imaginile astfel obținute, au fost ulterior introduse în aplicație.

Mi-am dorit ca aceste imagini să fie utilizate și pentru a afla informații suplimentare despre o anumită sală. Astfel, pentru a ușura procesul de identificare a unei săli am utilizat un câmp de tip map în interiorul paginii de vizualizare. Această hartă are definite mai multe arii, prin utilizarea unor perechi de coordonate de tip (x,y), arii ce definesc o anumită sală.

```
<div class="map" id="ED4">
  
  <map name="ed4">
    <area shape="poly" coords="546,593,485,593,485,508,546,508"
onclick="mapFunction(this.id) " id="Toaleta Femei">
    <area shape="poly"
coords="485,593,485,508,546,508,546,465,426,465,426,593"
onclick="mapFunction(this.id) " id="Toaleta Barbatii">
    <area shape="poly" coords="546,465,426,465,426,414,546,414"
onclick="mapFunction(this.id) " id="Scari">
    <area shape="poly" coords="546,414,546,367,470,367,470,414"
onclick="mapFunction(this.id) " id="Lift">
    .....
```

Asemănător cu modalitatea de funcționare a paginii Calendar, funcția mapFunction va realiza o cerere de tip ajax către server, se va realiza o căutare în baza de date a sălii ce are denumirea id-ului ariei selectate și se vor trimite acele informații paginii de vizualizare. Un nou câmp va fi utilizat pentru vizualizarea unor date suplimentare, adăugând și posibilitatea de a le vedea în întregime într-o altă pagină.

5. Concluzii

Implementări viitoare

La fel ca orice alta platformă sau aplicație web, dezvoltarea acesteia nu va înceta în momentul în care a fost atins termenul de livrare al aplicației pe piață. Se poate considera că platforma MyACS se afla la veriunea 1.0 din durata ei de viață. Aceasta conține o serie de funcționalități, funcționalități ce pot fi îmbunătățite. Desigur, exista și posibilitatea indrodcerii unor funcții complet noi. Mai jos se va putea gasi o listă ce conține posibile implementări viitoare ale aplicatiei MyACS:

- Oferirea unei experiențe mai personalizate, obținută prin adaugarea posibilitații de a modifica pagina de start astfel încât aceasta să conțină doar acele informații considerate mai importante de către utilizatori. Rezultatul așteptat este ca prin oferirea unui nivel mai ridicat de control utilizatorilor, acestia să beneficieze de o aplicație ce nu este a lor („My”) doar prin datele oferite ci și prin modalitatea de afișare a acestora;
- Introducerea unui sistem de recomandari pentru paginile de vizualizare a cursurilor și a evenimentelor. Inițial se poate introduce un sistem de filtrare bazată pe conținut, prin utilizarea tag-urilor ce au fost deja implementate în colecțiile cursuri și evenimente. Algoritmul poate fi dezvoltat și mai mult, astfel încât acesta să utilizeze situația școlară a studenților. După introducerea acestei metode, se poate merge și mai departe prin adaugarea unei filtrari colaborative. Acest lucru va putea fi realizat prin analiza paginilor cautate și/sau accesate de studenți, prin adaugarea unui sistem de notare a acestora sau de verificare a preferințelor studenților din aceeași grupa sau serie. Ultima etapă a acestei posibilități de dezvoltare va conta în formarea unui algoritm ce va combina cele doua modalități de filtrare, obținând astfel un sistem hibrid;
- Un al treilea punct de dezvoltare ar putea fi introducerea unui sistem de poziționare în interiorul facultății astfel încat utilizatorul să primească direcție de deplasare în timp real, nu doar static.

Opinie personală

Timpul este limitat, iar acest lucru face ca această resursă să fie extrem de importantă. Este esențial să prețuim fiecare moment și să încercăm să irosim cât mai puține secunde din viață. Din acest motiv este fundamental ca de oriunde ne-am afla să avem parte de acces rapid la informațiile de care avem nevoie.

Prin platforma MyACS am încercat să ofer fix acest lucru. Un loc ce poate fi accesat de fiecare student, profesor sau angajat al Facultății de Automatică și Calculatoare în orice moment al zilei pentru aflarea unor date ce privesc: cursurile ce se desfășoară în această facultate, evenimentele ce pot stârni interes și creativitate sau răspunsurile la întrebările ce ne preocupă. Aceste informații, precum și posibilitatea de a avea o agendă organizată a evenimentelor zilnice și o hartă a sălilor în care acestea se desfășoară cu siguranță vor fi considerate benefice de către utilizatorii ținta.

6. Bibliografie

Accesat în data de 12/06/2018

- „NodeJS”; informația poate fi găsită accesând adresa:
<https://en.wikipedia.org/wiki/Node.js>
- „What exactly is Node.js?” – Priyesh Patel; informația poate fi găsită accesând adresa: <https://medium.freecodecamp.org/what-exactly-is-node-js-ae36e97449f5>
- „What is NodeJs?”; informația poate fi găsită accesând adresa:
https://www.tutorialspoint.com/nodejs/nodejs_introduction.htm
- „Bootstrap”; informația poate fi găsită accesând adresa: <https://getbootstrap.com/>
- „MVC”; informația poate fi găsită accesând adresa:
<https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>
- „MVC Framework” – Introduction; informația poate fi găsită accesând adresa:
https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm
- „HTTP”; informația poate fi găsită accesând adresa:
https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol
- „Integrated Development Environment”; informația poate fi găsită accesând adresa:
https://en.wikipedia.org/wiki/Integrated_development_environment
- „Webstorm”; informația poate fi găsită accesând adresa:
<https://www.jetbrains.com/webstorm/>

Accesat în data de 25/05/2018

- „Express Web Framework”; informația poate fi găsită accesând adresa:
https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs
- „Express NodeJS”; informația poate fi găsită accesând adresa:
<https://www.upwork.com/hiring/development/express-js-a-server-side-javascript-framework/>

- „What is Express?”; informația poate fi găsită accesând adresa:
<https://www.programmableweb.com/news/what-expressjs-and-why-does-it-matter/analysis/2017/05/05>
- „ExpressJs”; informația poate fi găsită accesând adresa: <http://expressjs.com/>
- „What is MongoDB?”; informația poate fi găsită accesând adresa:
<https://www.mongodb.com/what-is-mongodb>
- „MongoDb”; informația poate fi găsită accesând adresa:
<https://en.wikipedia.org/wiki/MongoDB>
- „MongoDb”; informația poate fi găsită accesând adresa:
<https://www.techopedia.com/definition/30340/mongodb>
- „Sistem de gestiune a bazelor de date”; informația poate fi găsită accesând adresa:
<http://www.scribub.com/stiinta/informatica/SISTEME-DE-GESTIUNE-A-BAZELOR-92297.php>
- „NoSQL”; informația poate fi găsită accesând adresa:
<https://en.wikipedia.org/wiki/NoSQL>
- „Curs Baze de Date” – Prof. Dr. Ing. Dorin Carstoiu

Accesata în data de 24/04/2018

- „Css”; informația poate fi găsită accesând adresa:
https://en.wikipedia.org/wiki/Cascading_Style_Sheets
- „What is Css”; informația poate fi găsită accesând adresa:
https://www.tutorialspoint.com/css/what_is_css.htm
- „JavaScript”; informația poate fi găsită accesând adresa:
<https://en.wikipedia.org/wiki/JavaScript>
- „What is JavaScript”; informația poate fi găsită accesând adresa:
https://www.tutorialspoint.com/javascript/javascript_overview.htm

Accesat în data de 18/04/2018

- „Internet”; informația poate fi găsită accesand adresa:
<https://en.wikipedia.org/wiki/Internet>
- „HTML”; informația poate fi găsită accesand adresa:
<https://en.wikipedia.org/wiki/HTML>
- „What is HTML”; informația poate fi găsită accesand adresa:
<http://www.yourhtmlsource.com/starthere/whatishtml.html>

Accesat în data de 04/04/2018:

- „Start to be smart” – Asociatia SmartCityPro; informația poate fi găsită accesand adresa: <http://smartcitypro.ro/concept/notiuni-si-definitii/>
- „Smart City Concept” – Ministerul Comunicatiilor si Societatii informationale; informația poate fi găsită accesand adresa : <https://www.comunicatii.gov.ro/smart-city-concept-2/>
- „UPB va deveni primul SMART CAMPUS din Romania”; informația poate fi găsită accesand adresa: <http://www.upb.ro/evenimente/upb-va-deveni-primul-smart-campus-din-romania.html>
- „OnCampus: a mobile platform towards a smart campus” – Xin Dong, Xiangjie Kong, Flin Zhang, Zhen Chen si Jialiang Kang; informația poate fi găsită accesand adresa : <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4931999/>

Accesat în data de 11/03/2018:

- „Conceptul Smart City” – Prof. Mircea Eremia si Conf. Lucian Toma; informația poate fi găsită accesand adresa: <http://romaniasmartcities.ro/wp-content/uploads/2017/02/2015-01.pdf>
- „Orasul Inteligent” – Dragos Dinica, Catalin Vrabie si Catalin Dumitrica; informația poate fi găsită accesand adresa:
http://administratiepublica.eu/smartcitiesconference/publicatii/Orasul_inteligent_editia_03.pdf

- „Internet of Everything – Powering the Smart Campus & the Smart City” – Brad Davies; informația poate fi găsită accesând: <https://www.bhert.com/events/2015-06-08/Smart-Cities-Round-Table-Report-June-2015.pdf>

Preluare imagini:

- Fig.1 a fost preluată de pe pagina: <http://smartcitypro.ro/concept/notiuni-si-definitii/>