https://doi.org/10.31926/jmb.2021.2.5

APLICAȚIE PENTRU MANAGEMENTUL DATELOR ÎNTR-UN SPITAL INTELIGENT

DATA MANAGEMENT APPLICATION IN A SMART HOSPITAL

Laura Floroian
Universitatea Transilvania din Brașov
Autor corespondent: ***Laura Floroian***, email *lauraf@unitbv.ro*

Abstract:

Introduction: Information technology can contribute to the modernization of the medical system through dedicated software, which will improve the quality of medical services.

The main objective of the study: Development of a web and mobile application specific to a smart hospital, able to facilitate the management of patient data, based on free electronic resources (Java, JavaScript, HTML, CSS, Bootstrap, Firebase, XML, C ++).

Results and discussions: The developed application allows users (patients or healthcare professionals) to authenticate, to upload medical documents, which will be registered in the various cloud databases in the Firebase platform, and can be public or private. Users are allowed, depending on their grade, to see them, and even, in some cases, to act according to certain rules.

Conclusions: The system proposed in this paper is based on cloud technology, and can ensure the transmission of inter- and intra-hospital information in real time, so that each doctor can quickly receive all medical tests and other data necessary to decide on the actions to be undertaken to treat patients.

Rezumat:

Introducere: Tehnologia informației poate contribui la modernizarea sistemului medical prin softuri dedicate, care să îmbunătățească calitatea serviciilor medicale.

Obiectivul principal al studiului: Realizarea unei aplicații web și mobile specifice unui spital inteligent, capabilă să ușureze managementul datelor pacienților, bazată pe resurse electronice gratuite (Java, JavaScript, HTML, CSS, Bootstrap, Firebase, XML, C++).

Rezultate și discuții: Aplicația dezvoltată permite utilizatorilor (pacienți sau cadre medicale) să se autentifice, să încarce documente medicale, care vor fi înregistrate în diferitele baze de date de tip cloud din platforma Firebase, putând avea caracter public sau privat. Li se permite utilizatorilor, în funcție de gradul acestora, să le vadă, și chiar, în unele situații, să acționeze conform unor anumite reguli.

Concluzii: Sistemul propus în această lucrare aste bazat pe tehnologia cloud, și poate asigura o transmitere a informațiilor inter și intra spitalicești, în timp real, astfel încât fiecare medic să primească rapid toate analizele medicale și alte date necesare pentru a decide asupra acțiunilor care trebuie întreprinse pentru tratarea pacienților.

Key-words: *web and mobile application, management of patient data*
Cuvinte cheie: *aplicații web și mobile, managementul datelor în spitale*

Introducere

Sistemul de sănătate al unei țări este foarte complex și polivalent și reprezintă o organizare de persoane, instituții și resurse care oferă servicii ce ar trebui să întâlnească cerințele fiecărui pacient și să respecte standarde de calitate și de profesionalism maxime.

Managementul datelor din centrele medicale și transmiterea datelor către cadrele medicale sau pacienți reprezintă un criteriu important folosit pentru a defini calitatea sistemului de sănătate. În majoritatea spitalelor din România, toate datele sunt scrise pe hârtie si sunt transmise prin intermediul personalului, care astfel pierde timp prețios, în detrimentul îngrijirii bolnavilor.

Fiecare doctor trebuie să primească toate analizele medicale și restul de date necesare în luarea deciziilor pentru tratarea pacienților. Uneori timpul este foarte important, fiind situații în care un minut poate face diferența între viață și moarte.

Tehnologia informației a contribuit deseori la modernizarea sistemului medical

(Sinclair, 2017) prin softuri dedicate, care să îmbunătățească calitatea serviciilor medicale *(Greengard, 2015).*

De exemplu, Software-ul de management al spitalelor (HMS) poate ajuta instituțiile să creeze un sistem de date centralizat în care fiecare tip de informații este stocat într-un mod clasificat și poate fi regăsită cu ușurință în doar câteva clicuri, poate coordona toate tranzacțiile financiare care intră și ies din unitatea spitalicească. Astfel se asigură facturarea pacienților, plata personalului, reglarea conturilor cu furnizorii și generarea de rapoarte financiare. De asemenea el vine cu diverse instrumente care permit pacienților să îndeplinească anumite funcții, cum ar fi să contacteze medici, să își încarce istoricul medical, să programeze întâlniri și să plătească facturile.

Alt exemplu ar fi eHospital Systems, care este un sistem de management al spitalelor personalizabil, cuprinzător și integrat, conceput pentru clinicile cu mai multe specialități și medicii. Funcționalitatea cu mai multe locații permite interconectarea spitalelor, clinicilor satelit și magazinelor medicale.

Deși aceste softuri existente pe piață sunt utile pentru unitățile spitalicești, complexitatea lor le face să aibă prețuri care pot ajunge și la peste 15000 $, ceea ce le face greu accesibile unităților mici.

În această lucrare se prezintă realizarea unei aplicații web și mobile specifice unui spital inteligent, capabilă să ușureze managementul datelor pacienților, bazată pe resurse electronice gratuite.

O aplicație web este o aplicație software ce oferă informații, conținut sau servicii prin intermediul unui browser și rulează pe o structură client-server, fiind realizată cu ajutorul anumitor tehnologii, de exemplu: HTML, CSS, JavaScript. Tehnologiile utilizate în aplicațiile Web reprezintă limbajele de programare suportate de browsere, care folosesc un sistem de randare pentru a rula executabilul acestora. Avantajele acestor aplicații sunt multiple: pot fi rulate de pe orice dispozitiv pe care poate fi instalat o aplicație de tip browser, sunt independente față de sistemele de operare, astfel putând rula de pe orice dispozitiv, actualizările sunt ușor de realizat, de obicei printr-o simplă comandă, pot fi accesate de pe orice dispozitiv, desktop, tabletă, telefon mobil, fiecare utilizator poate accesa aceeași versiune, eliminându-se astfel problemele de compatibilitate, nu sunt instalate pe vreun spațiu de stocare al utilizatorilor, astfel eliminând limitările spațiului, au ciclu ușor de dezvoltare.

În timpul realizării aplicațiilor web, ele sunt împărțite pe două nivele: Front-end și Back-end. Nivelul de Front-end reprezintă ceea ce utilizatorii pot vedea iar nivelul de Back-end construiește arhitectura care oferă suportul de prelucrare a datelor. Atunci când se folosește browserul, utilizatorul trimite cereri de tip HTTP/HTTPS iar clientul primește răspunsuri.

Aplicațiile mobile oferă funcții limitate și izolate și sunt realizate să fie executate pe dispozitive mobile: telefon, tabletă. Acestea de obicei sunt descărcate de pe o platformă ce le distribuie numite magazine de aplicații, care sunt deținute de compania care a creat sistemul de operare, dar pot fi și instalate manual prin pachete. Aplicațiile mobile sunt realizate prin intermediul unor medii de dezvoltare create special pentru sistemele de operare care rulează pe dispozitivele mobile. Aceste medii utilizează anumite limbaje de programare: Java, Swift. Avantajele acestor aplicații sunt: utilizatorii nu trebuie să aștepte încărcarea paginilor web, costul redus de dezvoltare, accesul facil al utilizatorilor, interfață grafică pentru utilizatori ușor de dezvoltat, optimizare relativ ușoară, utilizare ușoară a modulelor integrate în dispozitivele electronice.

Aplicațiile de tip mobile sunt proiectate în două stadii. Primul stadiu reprezintă interfața grafică care va ajuta persoanele să acceseze funcțiile aplicației prin elemente vizuale afișate pe ecran, cu care vor interacționa. Al doilea stadiu este descris prin crearea funcțiilor prin anumite limbaje de programare care vor rula în background și vor îndeplini scopul principal pentru care au fost create *(Smyth, 2016, pp. 300-370).*

Pentru realizarea aplicațiilor se pot folosi o serie largă de tehnologii, iar unele dintre acestea se descriu în continuare.

Java este un limbaj de programare orientat pe obiecte care permite dezvoltarea de aplicații care pot rula pe orice sistem de operare care pot suporta mașini virtuale (Java Virtual Machine) care interpretează codul în bytecode. Acesta a fost introdus de programatorii James Gosling,

Mike Sheridan și Patrick Naughton în 1991, care au vrut să creeze un mediu de dezvoltare care putea fi scris o singură dată și rulat peste tot *(Liang, 2019).*

JavaScript, sau cum de obicei este abreviat, JS, este un limbaj de programare orientat pe obiecte bazat conceptul prototipurilor *(Flanagan, 2020).* Alături de alte limbaje de programare precum HTML, CSS, acesta este o tehnologie de bază ale World Wide Web, astfel, peste 97% dintre site-urile de pe internet foloses aceste limbaj pentru back-end, astfel creând funcționalități pentru clienți. Acesta a fost creat de Netscape în 1996, alături de un support pentru CSS și o extensie pentru HTML.

HyperText Markup Language (HTML5) este un limbaj de marcare a unor documente create pentru afișarea acestora într-un browser web, fiind asistat de cele mai multe ori de limbaje de programare precum CSS, Bootstrap, JavaScript *(Frain 2020).* HTML5 este compus din mai multe componente cheie, acestea având deseori etichete (și atributele lor). Etichetele acestora apar de obicei în perechi, fiind etichete de început și de sfârșit. Un alt element important este declararea tipului de document, ce declanșează redarea modului standard *(DuRocher, 2021).*

Cascade Style Sheets (CSS) este un limbaj folosit pentru descrierea prezentării unui obiect scris într-un limbaj de marcare, precum HTML. Este utilizat pentru a reduce complexitatea și repetarea în structură și pentru fezabilitatea de a prezenta aceleași elemente grafice prin diferite metode de redare. Selectorii declară pe ce parte a paginii Web se vor aplica diferite stiluri.

Bootstrap este un framework open-source bazat pe CSS care este utilizat în partea de Front-End al unui site web. Acesta conține aceleași elemente ca în CSS, iar opțional putem avea elemente din JavaScript. A fost dezvoltat de Mark Otto și Jacob Thornton la compania Twitter pentru a încuraja consistența în elementele de design.

Firebase este o platformă creată special de compania Google pentru a realiza mai ușor servicii de BackEnd *(Yahiaoui, 2019).* Acesta creează API care stochează datele într-o rețea cloud și le transmite în diferite aplicații web sau mobile. Este o platformă NONSQL în care datele sunt stocate sub forma unor crengi de copac *(Moroney, 2017).*

Extensible Markup Language (XML) este un limbaj de marcare care definește un set de reguli pentru documente care pot fi citite atât de om, cât și de dispositive. Acesta prezintă simplitate și generalitate în internet și este foarte utilizat pentru reprezentarea structurilor de date arbitrare, ca de exemplu cele folosite în servicii web sau mobile.

C++ este un limbaj de programare general realizat de Bjarne Stroustrup cu scopul de a fi o extensie a limbajului C. Acesta a evoluat în timp, adăugându-i-se mai multe atribute, cum ar fi orientarea pe obiecte sau pentru o utilizare mai mică a memoriei. Acesta este folosit în special pentru aplicațiile de tip embedded, dar poate fi utilizat și în alte moduri prin diferite tehnologii sau biblioteci *(Downey, 2019).*

Obiectiv

Obiectivul studiului este realizarea unei aplicații, web și mobile, care să asigure transmiterea informațiilor prin cloud în interiorul spitalelor sau între mai multe unități spitalicești, pentru a reduce timpul de răspuns al cadrelor medicale și salvarea istoricului medical al unui pacient într-o bază de date, pentru aflarea mai ușoară a informațiilor.

Metode

Pentru realizarea aplicației s-au utilizat următoarele tehnologii "open source": Java, JavaScript, HTML, CSS, Bootstrap, Firebase, XML, C++.

Rezultate

Aplicațiile web și mobile dezvoltate în această lucrare sunt compuse din mai multe module:

Modulul 1, „Autentificare / Înregistrare", permite utilizatorilor să își creeze un cont în care vor fi înregistrate toate datele cu caracter personal, cu posibilitatea de a le accesa în orice moment al zilei și a le păstra confidențiale, astfel putând fi accesate doar cu autentificarea cu un e-mail unic pentru fiecare utilizator.

Modulul 2, „Confirmare e-mail", oferă o anumită siguranță că nu se vor înscrie persoane cu date false.

Modulul 3, „Administrator", le permite angajaților desemnați de instituția care reali-

zează mentenanța zilnică, să anunțe eventualele sesizări făcute de utilizatorii simpli și să poate să controleze diferitele sisteme din spital.

Modulul 4, „Colectarea datelor", va permite înregistrarea în diferitele baze de date de tip cloud din platforma Firebase. Acestea vor avea caracter public sau privat, depinzând de tipul de utilizator autentificat.

Modulul 5, „Afișarea datelor", va realiza afișarea informațiilor colectate din diferite zone și va permite utilizatorilor, în funcție de gradul acestora, să le vadă, și chiar, în unele situații, să acționeze conform unor anumite reguli.

Modulul de înregistrare și autentificare

Pentru ca un utilizator să devină membru pe aceste aplicații este necesară înregistrarea, care este realizată prin câțiva pași simplu de urmat:

1. Completarea cu e-mail și libera alegere a unei parole ușor de memorat;
2. Alegerea unui nume de utilizator;
3. Completarea cu date confidențiale: CNP, dată de naștere, serie și număr CI;

Fiecare pas este dotat cu un câmp în care utilizatorul completează aceste date cu caracter personal, completarea lor având caracter obligatoriu, lipsa uneia împiedicând trecerea la pasul următor. Dacă un câmp va fi descoperit de protecția impusă, va fi afișat un mesaj cu textul „Completează câmpul obligatoriu", iar după completarea tuturor pașilor redirecționarea se va face către pagina principală. Dacă utilizatorul este deja înregistrat, va fi afișat mesajul „Sunteți deja înregistrat". Pot exista și alte modalități de înregistrare sau autentificare, prin intermediul contului de Google sau Facebook.

Utilizatorii își vor putea modifica în orice moment numele de utilizator și anumite date care sunt destinate publicului.

Există o diferență între utilizatorii simpli și anumiți utilizatori speciali, precum cadrele medicale sau administratorii. Funcționalitățile aplicației sunt limitate pentru utilizatorii simpli pentru a nu se crea probleme, ei putând doar introduce date, în timp ce utilizatorii speciali au acces la diferite secțiuni unde sunt stocate datele și vizualizate, pot da rezoluții și transfera datele la alte compartimente, de competența cărora este nevoie.

Secvența de cod care realizează înregistrarea unui utilizator nou în baza de date „Firebase Authentication" pentru mobile conține următoarele instrucțiuni:

- **Firebase –** apelarea platformei Firebase;
- **.auth() –** direcționarea spre funcționalitatea "Firebase Authentication";
- **.createUserWithEmailAndPassword() –** crearea contului de utilizator folosind cele două variabile de tip String furnizate de utilizator;
- **.addOnCompleteListener() –** apelarea bazei de date atunci când procedura este completă;
- **Log –** afișarea pentru utilizator a mesajului de confirmare sau de eroare sau chiar și pentru dezvoltator in debugger;
- **.catch(error) –** stocarea erorii;
- **startActivity() –** redirecționarea către altă clasă cu un anumit design și funcționalitate;
- **.show() –** afișarea pe ecranului telefonului/ tabletei anunțul de funcționalitate corectă sau greșită;
- **.setVisbility –** schimbarea vizibilității anumitor elemente grafice care vor dispărea și vor apărea în funcție metoda folosită;
- **setOnClickListener() –** oferă rolul ca la fiecare apăsare funcția atribuită să ruleze.

Iar pentru aplicația web:

- **console.log(firebase) –** scrierea în consola browserului că baza de date a fost accesată;
- **function –** realizarea unei funcționalități dorite, continuarea fiind numele acesteia;
- **getElementbyId() –** accesarea elementului cu numele unic dorit;
- **value –** returnarea valorii elementelor dorite;
- **toString –**transformarea valorilor obținute în valori de tip String;
- **window.alert() –**crearea unei notificări în browser pentru a asigura utilizatorul că operațiunea a fost dusă la bun sfârșit;
- **GoogleAuthProvider() –** accesarea contului de google salvat, sau adăugarea altuia și obținerea datelor necesare înregistrării;
- **signInWithRedirect() –** redirecționarea către pagina de înregistrare specifică Google și salvarea datelor oferite

Utilizatorii vor fi stocați sub denumirea de "Users" sub chei unice fiecărui utilizator, create în momentul înregistrării. După ce utilizatorii

sunt creați, aceștia rămân în cont o anumită perioadă de timp, după care deconectarea se va efectua automat. Pentru a reveni în contul acestora, vor putea utiliza pagina de ”Login” reprezentată în Fig. 1 și Fig. 2:

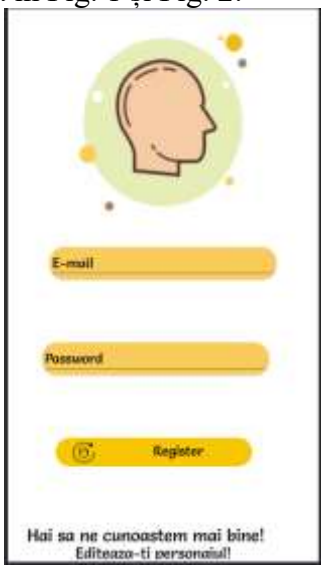

Fig. 1 Interfață grafică pentru mobile

Fig. 2 Interfață grafică pentru web

După efectuarea înregistrării, noul utilizator va avea obligația de a-și salva datele cu caracter personal (Fig.3).

Fig. 3 Interfață grafică pentru date cu caracter personal

Colectarea datelor

Informațiile publice vor fi stocate în diferite secțiuni pentru o organizare mai bună și un control mai mare asupra lor (Fig.4). Astfel, citirea acestora se va realiza într-un mod unic pentru fiecare scop.

- **#define** – definește o constantă pe toată perioada de execuție a întregului program;
- **FIREBASE_HOST** – realizează conexiunea la baza de date;
- **FIREBASE_AUTH** – realizează conexiunea la întreaga listă de utilizatori înregistrați;
- **WIFI_SSID** – constantă cu numele rețelei de internet ce va fi utilizată;
- **WIFI_PASSWORD** – constantă cu parola rețelei de internet;

```
#define FIREBASE_HOST "https://esp32
#define FIREBASE_AUTH "3zD04bBV2r8zw

#define WIFI_SSID "Tenda_391F60"
#define WIFI_PASSWORD "unitsoul320"
```

Fig. 4 Declararea datelor

Conexiunea la internet va fi verificată la fiecare resetare pentru a fi asigurat accesul la baza de date (Fig.5).

- **Wifi.localIP()** – afișarea IP-ului rețelei de internet la care se realizează conexiunea;
- **Firebase.begin()** – conectarea la baza de date cu datele cu caracter securizat oferite în mod automat de Google;

```
Serial.print("Connected to ");
Serial.println(WIFI_SSID);
Serial.print("IP Address is : ");
Serial.println(WiFi.localIP());
Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
```

Fig. 5 Începerea conexiunii cu baza de date

- **void setup()** – această funcție este apelată doar o dată și are scopul de inițializa variabilele, bibliotecile și modurile de utilizare a pinilor;
- **pinMode(OUTPUT)** – pinii în acest mod sunt în stare de impedanță mică și furnizează celorlalte circuite o cantitate substanțială de curent;
- **pinMode(INPUT)** – pinii sunt în stare de impedanță mare și sunt utilizați pentru a citi starea circuitelor conectate;

- **Serial.println() –** scrierea anumitor texte dorite în monitorul oferit de mediul de dezvoltare;
- **delay() –** realizarea unei întârzieri înaintea fiecărei executări a unei funcții;
- **Wifi.begin() –** conectarea plăcii de dezvoltare la o rețea de internet.

Transmiterea analizelor

Asistenții medicali vor putea transmite analizele pacienților împreună cu datele acestora prin selectarea unui fișier de pe calculator (Fig.6). Aceștia vor primi o notificare prin faptul că încărcarea a fost un succes. Resetarea elementului de încărcare se va face automat, astfel va putea fi pregătit oricând de o nouă utilizare (Fig.7).

Fig. 6 Interfață grafică pentru încărcarea unui document

Prin digitalizarea acestui proces, cadrele medicale vor avea acces cât mai rapid la datele necesare tratării unui pacient, reducându-se timpul de așteptare pentru primirea documentelor, accesarea lor fiind posibilă prin simpla apăsare a unui buton.

```
upload.addEventListener('change',function(e){
var file =e.target.files[0];
var storageRef=firebase.storage().ref('analize/'+file.name);
storageRef.put(file);
storageRef.getDownloadURL()
.then((url) => {
  // Insert url into an <img> tag to "download"

 // some code..

    window.alert("Ai transmis cu succes analizele!");
    var postref=firebase.database().ref('Analize');
    var newpostref=postref.push();
    var nume1=document.getElementById("nume_pacient").value;
  var email1=document.getElementById("e_mail_pacient").value;
  var cnp1=document.getElementById("cnp_pacient").value;
  var serie1=document.getElementById("serie_pacient").value;

    newpostref.set({
url:url,
nume:nume1,
email:email1,
cnp:cnp1,
serie:serie1

    });
  upload.value="";
})
```

Fig. 7 Funcție JavaScript pentru încărcarea unui document

- **getDownloadURL() -** la fiecare încărcare se va genera automat un link de descărcare care va fi stocat și accesat prin baza de date;

Afișarea datelor

Datele colectate vor fi afișate fie în anumite liste care vor fi integrate în anumite pagini specifice, fie în anumite secțiuni de text pentru datele simple care vor avea doar o singură variabilă. Listele se vor actualiza în mod automat la fel ca secțiunile de text, diferența dintre ele este faptul că listele vor avea un tipar numit ”card” sau ”view” în aplicația mobile care va reprezenta baza fiecărui obiect creat.

Secvența de cod care va realiza procedura menționată anterior în site-ul web va fi creată prin schimbarea vizibilității imaginilor, astfel utilizatorul va putea afla informații în timp real prin imagini interactive și animate.

Fiecare doctor va avea o pagină de tip HTML pentru a vizualiza fiecare document medical pentru pacienții lui și va primi în mod automat anumite date cu caracter personal atunci când aceștia vor trebui să primească anumite servicii medicale.

Sub fiecare secțiune “Analiză”, va exista un buton de descărcare (Fig.8) iar redirecționarea se va face automat în momentul acela, astfel încât cadrul medical ce utilizează această pagină nu va mai fi nevoit să instaleze alte programe de tip software pentru a le vizualiza, datorită suportului mare de fișiere oferit de majoritatea navigatoarelor de internet. Butonului de descărcare îi va fi atribuit un link la fiecare încărcare în sistem. De asemenea, cadrul medical poate redirecționa documentele către alt departament, sau chiar către alt spital, în funcție de necesități.

ANALIZE – 1

- *Nume și prenume: Popescu Flavius*
- *CN:P 1710918352874*
- *Serie și Număr: CI ZV 146238*

Download

Fig. 8 Tipar pentru afișarea analizelor

Concluzii

Tehnologia informației poate contribui la modernizarea sistemului medical românesc, prin softuri dedicate, care să îmbunătățească calitatea serviciilor medicale. Sistemul propus în această lucrare aste bazat pe tehnologia cloud, și poate asigura o transmitere a informațiilor inter și intra spitalicești, în timp real, astfel încât să reducă timpul, stresul și eventualele întârzieri cauzate de erorile umane. În același timp, poate restricționa accesul la anumite date confidențiale, dar și ușurează căutarea istoricului medical, prin reducerea timpului petrecut căutând în arhive sau evitând situații neprevăzute, precum pierderea dosarelor medicale sau distrugerea lor într-un incendiu. Prin crearea acestei aplicații web și mobile specifice unui spital inteligent, capabilă să faciliteze gestionarea datelor pacienților, fiecare medic va primi toate analizele medicale și alte date necesare pentru a decide asupra acțiunilor care trebuie întreprinse pentru tratarea pacienților.

Bibliografie

[1] Sinclair B., IoT Inc: How Your Company Can Use the Internet of Things to Win in the Outcome Economy, McGraw Hill Education, 2017, pp. 50-70.

[2] Greengard S., The Internet of Things, The MIT Press, 2015.

[3] Smyth N., Android Studio Development Essentials - Android 7 Edition, eBookFrenzy, 2016, pp. 300-370.

[4] Liang Y.D., Introduction to Java Programming and Data Structures, Comprehensive Version, Pearson, 2019, pp. 1000-1200.

[5] Flanagan D., JavaScript: The Definitive Guide: Master the World's Most-Used Programming Language, O'Reilly, 2020, vol. 7.

[6] Frain B., Responsive Web Design with HTML5 and CSS: Develop future-proof responsive websites using the latest HTML5 and CSS techniques, Packt Publishing, 2020, vol. 3, pp. 1-150.

[7] DuRocher D., HTML and CSS QuickStart Guide: The Simplified Beginners Guide to Developing a Strong Coding Foundation, Building Responsive Websites, and Mastering the Fundamentals of Modern Web Design, ClydeBank Media, 2021, pp. 25-120.

[8] Yahiaoui H., Firebase Cookbook: Over 70 recipes to help you create real-time web and mobile applications with Firebase, Packt Publishing, 2019.

[9] Moroney L., The Definitive Guide to Firebase: Build Android Apps on Google's Mobile Platform, Apress, 2017.

[10] Downey A.B., Think Java: How to Think Like a Computer Scientist, Green Tea Press, 2019.

Contribuția autorului: conceptualizare LF; designul cercetării: LF, validarea metodologiei: LF; culegerea datelor: LF, analiza datelor și / sau interpretarea datelor: LF; scriere-pregătirea textului inițial LF, revizuire și editare: LF

Surse de finanţare: Această cercetare a fost finanțată de Uniunea Europeană, proiectul de fonduri structurale PRO-DD (POS-CCE, O.2.2.1., ID 123, SMIS 2637, ctr. Nr. 11/2009).

Conflicte de interese: autorul nu are conflicte de interese relevante pentru acest articol.

Mulţumiri: prof. dr. Mihaela Badea, pentru supervizare.