

UNIVERSITATEA „ALEXANDRU IOAN CUZA” IAȘI

FACULTATEA DE INFORMATICĂ



LUCRARE DE LICENȚĂ

VoteIT

propusă de

Roman Ștefan

Sesiunea: *iulie 2019*

Coordonator științific

Drd. Florin Olariu

UNIVERSITATEA „ALEXANDRU IOAN CUZA” IAȘI

FACULTATEA DE INFORMATICĂ

LUCRARE DE LICENȚĂ

VoteIT

propusă de

Roman Ștefan

Sesiunea: *iulie 2019*

Coordonator științific

Drd. Florin Olariu

Avizat,

Îndrumător Lucrare de Licență

Titlul, Numele și prenumele _____

Data _____ Semnătura _____

DECLARAȚIE privind originalitatea conținutului lucrării de licență

Subsemnatul(a)

domiciliul în

născut(ă) la data de, identificat prin CNP, absolvent(a) al(a) Universității „Alexandru Ioan Cuza” din Iași, Facultatea de specializarea promoția, declar pe propria răspundere, cunoscând consecințele falsului în declarații în sensul art. 326 din Noul Cod Penal și dispozițiile Legii Educației Naționale nr. 1/2011 art.143 al. 4 și 5 referitoare la plagiat, că lucrarea de licență cu titlul:

_____elaborat

ă sub îndrumarea dl. / d-na _____, pe care urmează să o susțină în fața comisiei este originală, îmi aparține și îmi asum conținutul său în întregime.

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

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

Data azi,

Semnătură student

ACORD PRIVIND PROPRIETATEA DREPTULUI DE AUTOR

Facultatea de Informatică este de acord ca drepturile de autor asupra programelor-calculator, în format executabil și sursă, să aparțină autorului prezentei lucrări, *Roman Ștefan*.

Încheierea acestui acord este necesară din următoarele motive:

Iași, *data*

Decan *Prenume Nume*

(semnătura în original)

Absolvent *Prenume Nume*

(semnătura în original)

Rezumat

În această lucrare se descrie funcționalitatea aplicației denumită VoteIT împreună cu toate componentele ce o alcătuiesc. VoteIT este o aplicație web construită cu ajutorul tehnologiei ASP.NET, venită în ajutorul utilizatorului pentru a ușura acestuia modalitatea prin care se poate vota. Printr-o simplă introducere a unei poze de buletin, a introducerii unui username și a unei parole, îi este creat un cont de votant prin intermediul căruia acesta își poate exprima votul. Prin introducerea pozei de buletin, se asigură o corectitudine a preluării datelor în urma unui OCR ce preia numele, prenumele și CNP-ul votantului. De altfel, se asigură și o unicitate a votului, deoarece un utilizator cu drept de vot, va avea dreptul la un singur vot la fiecare sesiune deschisă. De asemenea, prin intermediul aplicației, se pot înscrie și candidați la sesiunile de votare.

Cuvinte cheie

Asp.NET, aplicație, web, vot, OCR, unicitate, votant, candidat.

Cuprins

Introducere si motivație.....	7
Contribuții	8
Capitolul I – Descrierea problemei	9
Capitolul II – Abordări anterioare.....	9
II.1 ElectionBuddy	9
II.2 Voteaza.live	11
II.3 Concluzii	12
Capitolul III - Descrierea soluției	13
III.1Tehnologii folosite.....	13
III.1.1.1 .Net Framework	13
III.1.1.3 ASP.Net Core	13
III.1.1.4 Entity Framework Core.....	14
III.1.2 React	14
III.1.2.1 Componente React.....	14
III.1.2.2 Configurarea React cu ASP.Net	16
III.1.3 OCR	17
III.1.4 Sistem de versionare	18
III.2-Aritectura aplicației	19
III.2.2 Arhitectura pe mai multe nivele (n-tier).....	19
III.2.2.1 Nivelul de Logică	20
III.2.2.2 Nivelul de Date	21
III.2.2.3 Nivelul de Prezentare	22
III.3-Prezentarea generala a aplicației	23
III.3.1 Desfășurarea aplicației	23
III.3.2 Înregistrarea și logarea în aplicație	23
III.3.3 Paginile principale ale aplicatiei	26
III.3.3.1 Componenta Admin	26
III.3.3.2 Componenta Votant	27
III 3.3.3 Componenta Candidat.....	28
Concluzia lucrării.....	29
Bibliografie.....	30
Anexa 1	31

Introducere si motivație

Punctul de plecare al acestei lucrări de licență este reprezentat de către evoluția fără precedent a tehnologiei și inserarea resurselor acesteia în toate sectoarele de activitate ale societății, chiar și în politică.

Nevoia unei astfel de implementări este subliniată, pe de o parte, de transformările din domeniul tehnologiei și, pe de altă parte, de importanța accentuată pe care resursele tehnologiilor digitale o au în viața nativilor digitali.

Prin această lucrare se încearcă rezolvarea problemelor din timpul alegerilor politice.

În ultimul timp observăm în afara țării neputința autorităților de a rezolva problema oamenilor ce stau ore întregi la coada pentru a vota și în cele din urmă nu reușesc. Aceasta aplicație nu vine doar cu un mod mult mai ușor de a vota, pe lângă acest fapt este combătută și încercarea fraudărilor alegerilor prin voturi multiple de către aceeași persoană în numele altor persoane.

Motivația ce a stat la baza acestei lucrări de licență este formată din două arii principale:

- Aria non-tehnica
 - Timpul pierdut de votanți pe drum până la secțiile de votare și în unele cazuri timpul petrecut la cozile din secțiile de votare.
 - Dorința de a avea o modalitate simplifică în procesul de votare clasic.
- Aria tehnică
 - Dorința de a dovedi că se poate construi o aplicație de votare și în România unde se poate vota într-un timp mult mai scurt.
 - Pasiunea pentru programarea în .Net pe Web.

Contribuții

În urma cercetării soluțiilor propuse de către statul român cu privire la rezolvarea problemelor votării din afara țării, s-a simțit nevoia implementării unui alt sistem de votare deoarece cele existente la ora actuală, nu rezolvă în totalitate problemele apărute. Datorită numărului mare de români plecați în afara țării, în ultimul timp s-a văzut o neputință a autorităților de a prelua votul tuturor persoanelor cu drept de vot ce și-au dorit să își exercite acest drept.

Principala contribuție în aceasta lucrare de licență a fost găsirea și dezvoltarea unei soluții ce permite votul de la distanță pentru a rezolva și problema deplasării până la urnele de vot. Distanța fiind unul dintre cele mai mari impedimente pentru cei ce lucrează în afara țării, din pricina numărului redus de urne, acestea fiind amplasate în principalele orașe ale unei țări.

O altă contribuție ce trebuie menționată este realizarea aplicației folosind tehnologii de actualitate precum Asp.Net Core și React, ce permit ca aplicația să ruleze pe mai multe sisteme de operare precum Windows, macOS, Linux și o integrare ușoară între cele două componente de Front-End și Back-End, aceste lucruri asigurând și o performanță la nivel de aplicație.

Modul în care a fost gândită aplicația trebuie văzut ca și o contribuție deoarece prin funcționalitatea sa, aceasta propune unele măsuri de siguranță pentru a combate încercările de a fraudă alegerile. O modalitate prin care se asigură un vot corect, ce combate ștampilarea buletinului de vot de către o altă persoană pentru propriile interese, ar fi crearea unui cont de utilizator unic pentru fiecare votant atunci când acesta se va înregistra pe baza buletinului. Astfel este redusă șansa ca o persoană să voteze în numele altei persoane, practică ce a fost des întâlnită în România. Tot la nivelul de înregistrare s-a recurs la folosirea de recunoaștere a textului de pe imagini pentru a prelua datele personale, nume, prenume, cnp de pe o carte de identitate, astfel sunt eliminate riscurile unor greșeli de prelucrare. O altă modalitate ce sporește șansele unui vot corect este validarea votului, astfel, atunci când cineva dorește să voteze, întâi se vor face verificări pentru a vedea dacă acesta a mai votat în sesiunea curentă. Prin urmare, se combate riscul unui vot multiplu.

Capitolul I – Descrierea problemei

Dreptul de vot este unul din principiile de bază ale democrației ce permite cetățenilor unui stat să voteze pentru a-și exprima voința cu ocazia unui scrutin. În aplicația sa contemporană, permite cetățeanului fie să își aleagă reprezentanți care să guverneze, fie să răspundă unei întrebări puse de guvernare sub forma unui referendum.

Punctul de plecare al acestei aplicații este reprezentat de dorința de a îmbunătăți condițiile de votare ale cetățenilor români. După cum se știe, în ultimii ani, prezența la urnele de vot a românilor din afara țării simte o creștere considerabilă. Datorită acestui fapt, nu toți mai apucă să ajungă la urne, cei mai mulți petrecând ore întregi în cozile interminabile.

O soluție pentru această problemă este crearea unei aplicații online, ce trebuie să substituie procedura de votare clasică, astfel evitând deplasarea până la urnele de vot și statul la coadă.

Capitolul II – Abordări anterioare

Există o varietate mare de aplicații care încearcă să aducă cele mai bune rezultate în procedura de votare și care promit o soluție securizată și o metoda simplă de votare. În acest capitol vom enumera și prezenta unele dintre ele.

II.1 ElectionBuddy

ElectionBuddy este rezultatul muncii unor studenți în urma frustrărilor dobândite după pierderea multor ore în care au numărat buletine de vot și au calculat rezultatele voturilor. Ca urmare a acestui fapt s-a realizat o aplicație ce asigură integritatea alegerilor, salvează ore bune și îmbunătățește prezența la vot pentru votul la distanță. Cel ce votează poate vota o singură dată iar votul său va rămâne unul anonim. Fiecare vot are o cheie securizată iar votul poate fi verificat și observat independent. Cheia de votare este creată și folosită o singură dată și folosește o criptare pe 256 biți.

Aplicația putând fi utilizată urmând trei pași simpli:

1. Configurarea tipului de alegere

Details > Ballot > Notice > Voters > Review > Pay > Results

To begin, start by answering questions about your election.

Information

Election Title

Organization

Shared with voters on ballots and election results

Primary Election Administrator

Ballots will include name and email to facilitate voter questions

☐ This is a test election
Try all the features by creating a free sample election with up to 5 voters

Fig1. Configurarea unei alegeri

În Figura 1 putem vedea cum se adaugă un nou tip de alegeri. Prin urmare va trebui introdus tipul de alegeri, date și setări iar mai apoi listele cu candidați.

2. Metoda de votare

Ballot > Verify > Confirmation

RightLabs

RightLabs Inc.
Board of Directors Election

Executive Board

Select exactly 2 of 6 candidates or choose abstain to vote for none of the candidates.

Candidate	
Robert Clarke	<input type="checkbox"/>
Betty Harris	<input type="checkbox"/>

Fig2. Alegerea unui Candidat

Figura 2 prezintă lista cu candidați ce vor fi supuși votului. Un votant are un singur vot la

dispoziție iar în urma folosirii acestuia nimeni nu va putea avea dreptul să vadă cu cine a votat.

3. Afișarea rezultatelor

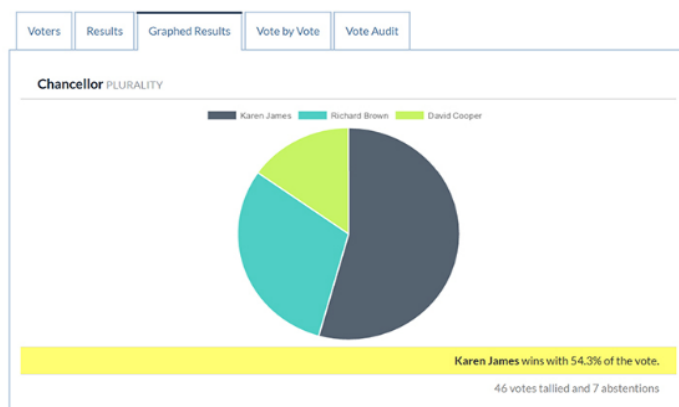


Fig.3 Modul prin care sunt afișate rezultatele

Figura 3 prezintă cum vor fi afișate rezultatele finale. În urma încheierii unei sesiuni, câștigătorii sunt automat aleși și vom avea la dispoziție și un grafic ce va prezenta în procentaje rezultatele alegerilor.

Figurile de mai sus: 1, 2 și 3 sunt preluate de la următorul link: [1].

Printre principalele îmbunătățiri pe care le aduce această aplicație enumerăm sistemul de verificare asupra votului de către o entitate neutră, sistemul de notificare ce este îndreptat către utilizatorii ce nu au votat.

II.2 Voteaza.live

Recent, în România, cu ocazia alegerilor europarlamentare 2019 s-a testat prima aplicație construită de către români, voteaza.live prin care votanții la europarlamentare își pot urmări votul în timp real.



Fig.4 Aplicația voteaza.live

Figura 4 prezintă interfața cu utilizatorul. Aceasta prezintă o interfață plăcută, interactivă. Pentru a trimite votul mai departe, utilizatorul va trebui să se autentifice în aplicație cu contul de Facebook. Aceasta platformă de votare a fost retrasă la scurt timp după lansare și nu mai permite utilizatorilor să intre pe portal.

II.3 Concluzii

În concluzie, putem vedea multiple propuneri de implementări pentru această problemă a votului, dar aceste soluții nu se regăsesc și în România la nivel practic, în ultimul timp doar s-au făcut teste pe o astfel de platformă. Ca urmare a acestui fapt, se simte nevoia unei soluții în problema votului și o aplicație nouă ar fi bine-venită.

Capitolul III - Descrierea soluției

III.1 Tehnologii folosite

Acest capitol este creat cu scopul de a prezenta tehnologiile folosite în crearea acestui proiect. Scopul aplicației este de a prelua votul de la un utilizator cu drept de vot și de a centraliza toate voturile în scopul desemnării unui câștigător. Pe de altă parte, aplicația poate fi văzută și ca un proiect de cercetare pentru a vedea dacă sistemul de votare din Romania poate fi îmbunătățit prin folosirea acestei aplicații web.

III.1.1.1 .Net Framework

.NET Framework este un framework dezvoltat de către Microsoft și include tehnologii precum ASP.NET WebForms, ASP.NET MVC, Web Api și nu în ultimul rând .NET Core 2.0.

III.1.1.2 .Net Framework Core

Principalul avantaj în a folosi .NET Core îl constituie faptul că este cross-platform, acest lucru însemnând că aplicația poate fi rulată pe mai multe platforme printre care se enumeră Windows, Mac, Linux. Pe lângă acest lucru, .NET Core este open source, ceea ce înseamnă că utilizatorii pot ajuta la îmbunătățirea framework-ului.

III.1.1.3 ASP.Net Core

Funcționalitatea aplicației a fost construită integral în Asp.Net Core. Acesta este un framework pentru construirea aplicațiilor web pe platforma .Net. S-a recurs la folosirea Asp.Net Core deoarece această tehnologie suportă crearea de servicii REST, știute și sub numele de web api-uri. Sunt necesare aceste web api-uri deoarece aceasta este metoda prin care cele două componente (front-end și back-end) pot face schimb de date între ele.

III.1.1.4 Entity Framework Core

Entity Framework(EF) Core este un object-relational mapper(ORM) și o tehnologie de acces a datelor pentru proiecte ce folosesc .NET și ASP.NET ce asigură lucrul cu baze de date relaționale. Principalul avantaj pentru un developer în folosirea EF este modul mai ușor de manipulare a datelor din baza de date, prin crearea și updatarea acesteia în funcție de ce nevoi are programatorul în aplicația sa. Acest framework introduce conceptul de Code-First, unde accentul principal este îndreptat către funcționalitatea aplicației, renunțând astfel la crearea bazei de date în primă fază. La început de aplicație, cu ajutorul Entity Framework am reușit să mă axez pe funcționalitatea proiectului deoarece baza mea de date era reprezentată de modele de date. Fiecare model fiind corespunzător unei tabele din baza de date.

Mai multe informații despre configurarea proiectului cu Entity Framework se pot găsi în Anexa 1 pe pagina 30.

III.1.2 React

React (cunoscut și sub numele de React.js sau ReactJS) este o librărie de JavaScript construită de către Facebook în parteneriat cu o comunitate de developeri și companii, pentru crearea aplicațiilor mobile cât și desktop. Principalele avantaje ale Reactului sunt îmbunătățirea productivității, viteza de producere și livrare a aplicațiilor datorită structurii simplu de înțeles.

III.1.2.1 Componente React

Aplicația se bazează pe patru componente de baza, pagina home, pagina de admin, pagina de candidat și pagina de votant.

VotelT
Home Login

Voters
Candidates

Voters Registration

Username:

Password:

Identity Card Photo:

No file chosen

Fig.5 Pagina de Home

Figura 5 prezintă pagina de Home unde avem funcționalitățile de register și de login.

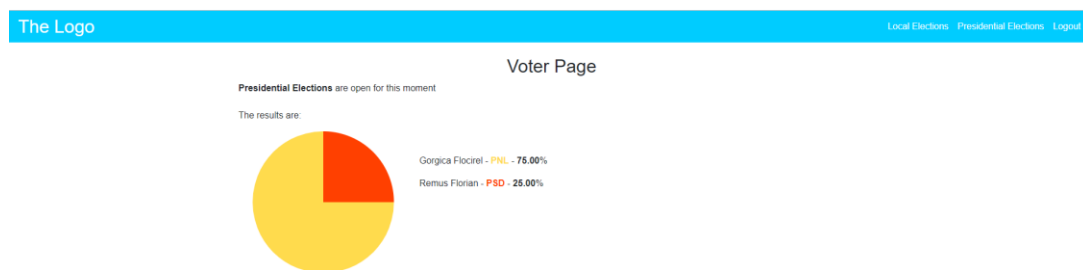


Fig.6 Pagina de Votant

Figura 6 prezintă pagina pentru Votant și posibilitatea de a alege între cele două tipuri de sesiuni.

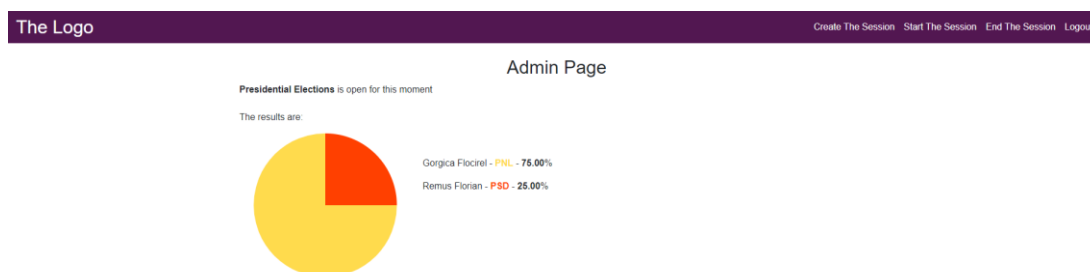


Fig.7 Pagina de Administrator

În figura 7 ne este evidențiată pagina de administrator împreună cu funcționalitățile pe care acesta le poate îndeplini(create, start, end session).

The screenshot shows a web interface titled 'Candidate Page' with a subtitle 'Register candidature'. At the top, there is a dark purple header bar containing 'The Logo' on the left and 'Register candidature' and 'Logout' on the right. The main content area has a light gray background. It starts with the heading 'Personal Description:' followed by a text input field containing the text 'I will work harder every day to improve your live conditions'. Below this is a section for 'Political Party' with a dropdown menu currently showing 'PSD'. Underneath is a section for 'Choose type of candidature' with a dropdown menu currently showing 'Local Elections'. At the bottom of the form is a green button labeled 'Save Profile'.

Fig.8 Pagina de înregistrare a candidaturii

În figura prezentată mai sus, observăm funcționalitatea pe care un candidat o are la dispoziție.

Mai multe detalii despre structura unui proiect React se pot găsi în Anexa 1 pe pagina 31.

III.1.2.2 Configurarea React cu ASP.Net

Asp.Net rulează diferit aplicația React în cele doua moduri: modul de producție și modul de dezvoltare. În modul de dezvoltare, serverul de dezvoltare al React-ului rulează automat în background, așa că resursele provenite de pe partea clientului sunt construite dinamic iar pagina își va da refresh automat când se va schimba conținutul unui fișier. Pentru a configura React în modul de dezvoltare, este nevoie ca acesta să ruleze prin serviciile ASP.NET Core SPA. Acest lucru se va petrece în clasa Startup.cs în metoda Configure.


```

app.UseSpa(spa =>
{
    spa.Options.SourcePath = "ClientApp";

    if (env.IsDevelopment())
    {
        spa.UseReactDevelopmentServer(npmScript: "start");
    }
});

```

Fig.9 Configurarea Serverului React în modul de dezvoltare

```

public void ConfigureServices(IServiceCollection services)
{
    services.AddMvc().SetCompatibilityVersion(CompatibilityVersion.Version_2_2);

    // In production, the React files will be served from this directory
    services.AddSpaStaticFiles(configuration =>
    {
        configuration.RootPath = "ClientApp/build";
    });
}

```

Fig10. Configurarea Serverului React în modul de producție.

Figura 10 prezinta codul care în modul de producție, fișierele react vor fi preluate din directorul ClientApp/build.

III.1.3 OCR

Recunoașterea Optică a Caractere, abreviată și OCR, este o tehnologie ce permite convertirea diferitelor tipuri de documente, precum imagini, documente PDF în text editabil.

III.1.3.1 Tesseract

Pentru a reuși preluarea textului(nume, prenume și cnp) de pe cartea de identitate a unui votant s-a folosit Tesseract. Acesta este unul dintre cele mai exacte motoare de OCR ce poate fi folosit pe mai multe sisteme de operare. Tesseract permite convertirea imaginii încărcate de un votant(cartea de identitate) în text, identificând caracterele prezente într-o imagine și pune aceste caractere în cuvinte, permițând să se caute apoi și să se editeze conținutul rezultat.

Pentru a configura aplicația cu Tesseract s-au urmat pașii următori:

1. S-a adăugat motorul Tesseract din NuGet Package Manager
2. S-a adăugat fișierul cu date de antrenament pentru Tesseract

```
var img = Pix.LoadFromFile(_path);  
var pathTessData = "D://VoteIT//VoteIT//VoteIT//tessdata";  
var engine = new TesseractEngine(pathTessData, "ron", EngineMode.Default);  
var page = engine.Process(img, PageSegMode.Auto);  
string result = page.GetText();
```

Fig.11 Prelucrarea unei imagini cu Tesseract

După cum putem observa în figura 11 pentru a putea prelucra imaginea, Tesseract are nevoie de un Bitmap așa că la încărcare, imaginea va fi preluată de la un path iar apoi va fi convertită în format Bitmap. Apoi cel ce se va ocupa de prelucrarea textului de pe imagine va fi motorul Tesseract.

III.1.4 Sistem de versionare

Existând întotdeauna șansele unor probleme tehnice, pentru a nu pierde toata munca depusă la aplicație s-a ajuns la concluzia că ar fi bine să se stocheze codul sursă într-un sistem de versionare. Acest sistem permite gestionarea versiunilor multiple a unor fișiere, precum și lucrul în paralel asupra unor fișiere. Sistemele de versionare au fost concepute pentru a permite membrilor mai multor echipe să opereze modificări pe același proiect, aceste modificări urmând a fi reunite într-o nouă versiune a proiectului.

Avantajele folosirii unui sistem de versionare:

1. Fiind salvat istoricul modificărilor, se poate reveni cu ușurință la o variantă de proiect mai veche dacă se descoperă introducerea unor defecte în ultima versiune.
2. Utilizându-se un serviciu de hosting, codul sursă va avea mereu o copie de siguranță online
3. Dezvoltatorii proiectului vor avea mereu acces la ultima versiune a codului, făcând astfel colaborarea și sincronizarea mult mai ușoară decât în cazul trimiterii de fișiere ce conțin cod sursă.

III.1.4.1 GitHub

GitHub este un serviciu de hosting al proiectelor git (un proiect git mai este numit și repository). Acesta permite păstrarea unei copii a proiectului online și vizualizarea tuturor fișierelor și a modificărilor aduse acestora.

Mai multe informații în legătură cu crearea unui nou repository se găsesc în Anexa 1 la pagina 32.

III.2-Aritectura aplicației

O arhitectură reprezintă setul de decizii semnificative privind organizarea unui sistem software, selectarea elementelor structurale și a interfețelor acestora prin care este compus sistemul, împreună cu comportamentul specificat în colaborările dintre elementele respective. Aceasta va putea fi aleasă după ce vor fi studiați mai mulți factori precum posibilitatea de modificare a codului, performanța cu scopul de a îmbunătăți funcționalitatea aplicației.

III.2.2 Arhitectura pe mai multe nivele (n-tier)

În aplicație s-a folosit o arhitectură pe trei nivele, aceasta fiind probabil una dintre cele mai cunoscute arhitecturi (n-tier). S-a recurs la acest tip de arhitectură deoarece funcționalitatea este construită în jurul unei baze de date și din dorința de a separa aplicația în mai multe nivele (nivelul de prezentare, nivelul de logică și nivelul de date) cu scopul de a gestiona ușor codul dintr-un nivel fără a afecta și celelalte nivele.

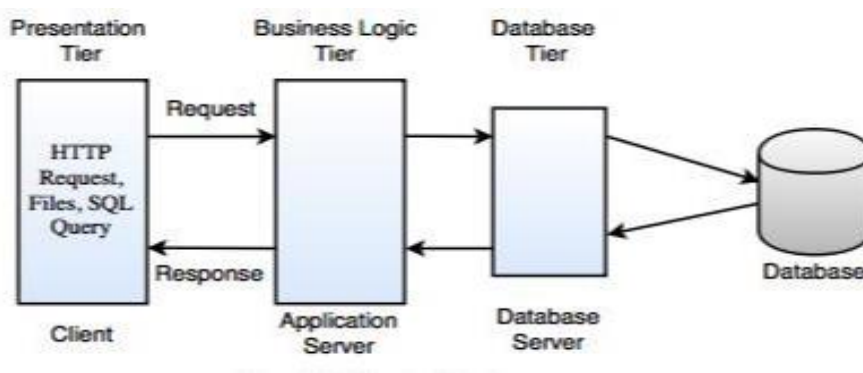


Fig. 12 Arhitectura pe 3 nivele

Figura 12 este preluată și are ca sursă următorul link[2].

III.2.2.1 Nivelul de Logică

Nivelul de logică este scris în C# și conține logica ce formează funcționalitățile de baza ale aplicației. În aplicația VoteIT acest nivel este asigurat de către multe clase controller. Aceste controllere au rolul de a gestiona cererile browserului și în funcție de acestea va furniza datele dorite de către utilizator.

Un exemplu din aplicație ar fi UserController ce gestionează partea de logare în aplicație a unui utilizator. Când un request de login este trimis de la user către aplicație, în metoda din controller se verifică autenticitatea datelor iar dacă totul este în regulă, se permite userului să intre în aplicație pe o pagină corespunzătoare în funcție de tipul de utilizator logat (votant, candidat, administrator).

Un alt exemplu ar fi în VotingHistoryController atunci când se înregistrează o cerere de votare.

```
[Microsoft.AspNetCore.Mvc.Route("PostNewVote")]
[Microsoft.AspNetCore.Mvc.HttpPost]

public ActionResult<IHttpActionResult> PostVote([System.Web.Http.FromBody] VoteDetails voteDetails)
{
    var voterID = voteDetails.VoterId;
    var _user = _context.Users.FirstOrDefault(user => user.Id == voterID);

    //check if user voted already
    if (_user.Voted == false)
    {
        //Update voting option for user -> TRUE(voted)
        _user.Voted = true;
        //Create new Vote
        VotingHistory newVote = new VotingHistory();
        newVote.CandidateId = voteDetails.CandidateId;
        newVote.SessionId = voteDetails.SessionId;
        newVote.SessionDate = voteDetails.SessionDate;
        newVote.CityId = voteDetails.CityId;

        _context.VotingsHistory.Add(newVote);
        _context.SaveChanges();
        return StatusCode(201);
    }
    else
    {
        return StatusCode(409);
    }
}
```

Fig.13 Tratarea cererii de votare a unui utilizator

După cum putem observa în Figura 13 atunci când un utilizator încearcă să voteze, întâi este căutat în baza de date, apoi se verifică dacă acesta a mai votat, iar în caz că nu a mai votat, îi este înregistrat votul.

III.2.2.2 Nivelul de Date

Acest nivel de arhitectura are rolul de a gestiona baza de date și de a furniza datele din aceasta. Cuprinde baza de date/sistemul de stocare a datelor și stratul de acces la date. Exemple de altfel de sisteme sunt MySql, Oracle, PostgreSQL, Microsoft SQL Server, MongoDB, etc. Acest nivel este compus la rândul lui din domeniul de date și persistenta datelor.

Domeniul de Date

Acest domeniu este compus din toate modelele ce sunt folosite pentru a genera baza de date a aplicației (User, VotingHistory, VotingSession). Fiecare model are propriile sale proprietăți ce vor corespunde proprietăților unei tabele din baza de date. Câteva exemple de proprietăți pentru modelul User din aplicație ar fi FirstName, LastName, Cnp, Voted.

```
namespace VoteIT.Models
{
    public class VotingHistory
    {
        public int Id { get; set; }

        public int CandidateId { get; set; }

        public int SessionId { get; set; }

        public int CityId { get; set; }

        public DateTime SessionDate { get; set; }
    }
}
```

Fig.14 Modelul pentru tabela VotingHistory

Persistenta Datelor

O importanță majoră în structura proiectului este folderul unde se păstrează toate migrările făcute în timpul dezvoltării aplicației. Având acces la istoricul migrărilor, în fiecare moment ne

putem întoarce ușor la o versiune anterioară a bazei de date în cazul în care nu merge corespunzător versiunea curentă.

Structura bazei de date

Pentru a înțelege cea mai importantă funcționalitate din aplicație, cea de votare, trebuie să știm structura bazei de date, respectiv fiecare model (Votant, Candidat, Istoric Votări, Sesiuni de Votare și Oraș) și prin ce câmpuri comune sunt legate acestea între ele.

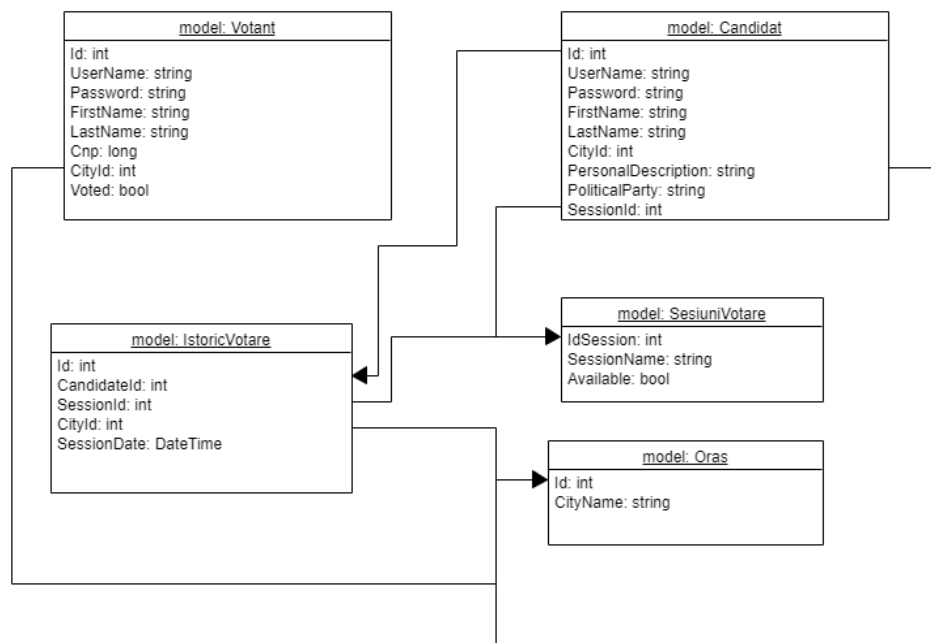


Fig.15 Structura bazei de date

III.2.2.3 Nivelul de Prezentare

Nivelul de prezentare este stratul de front end din arhitectura pe trei nivele și constă în interfața cu utilizatorul. Această interfață este ceea ce vede utilizatorul și cu ce interacționează

acesta. Aici intră informațiile introduse de utilizator, iar acest nivel se comportă ca o interfață între nivelul de date și utilizator, transmițând diferitele acțiuni ale utilizatorului către nivelul logic.

Acest nivel este adesea construit în tehnologii precum HTML5, JavaScript, CSS sau prin alte framework-uri sau librării de dezvoltare precum Angular, React, cum este și cazul aplicației de față ce se folosește de React.

III.3-Prezentarea generala a aplicației

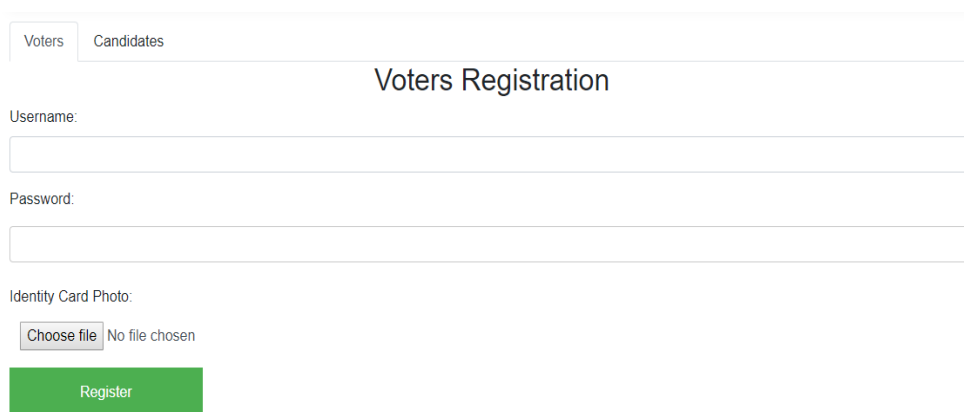
Aceasta aplicație este reprezentată de o parte de back-end construită în ASP.Net și o parte de front-end construită în React. Scopul ei principal este cel de a oferi o modalitate cât mai simplă pentru a vota online și de a evita cozile de la secțiile de votare.

III.3.1 Desfășurarea aplicației

VoteIT este o aplicație ce suportă mai multe tipuri de utilizatori și propune pentru fiecare dintre utilizatori o funcționalitate diferită în funcție de cel logat.

Atunci când un nou utilizator va dori să folosească aplicația, el va fi întâmpinat de prima pagină, unde i se va sugera să se înregistreze. În cazul în care acesta va avea deja creat un cont, se va loga în aplicație.

III.3.2 Înregistrarea și logarea în aplicație



The screenshot shows a web interface for 'Voters Registration'. At the top, there are two tabs: 'Voters' (selected) and 'Candidates'. The title 'Voters Registration' is centered. Below the title, there are three input fields: 'Username:', 'Password:', and 'Identity Card Photo:'. The 'Identity Card Photo:' field has a 'Choose file' button and the text 'No file chosen'. At the bottom, there is a green 'Register' button.

Fig.16
Înregistrare
a unui nou
Votant

Figura 16 reprezintă modalitatea de înregistrare pentru utilizatorul de tip votant.

Principalul avantaj al utilizării aplicației este faptul ca atunci când se va dori crearea unui nou cont de votant, va fi necesară introducerea unui username și a unei parole iar apoi utilizatorul va trebui să încarce poza de buletin. Prin încărcarea pozei, la crearea contului, acestuia i se va asocia în baza de date numele, prenumele și cnp-ul extrase de pe buletin pentru a ține în permanență evidența dacă acea persoană și-a exercitat dreptul la vot.

Modul de verificare CNP

Codul numeric personal reprezintă un număr semnificativ ce individualizează o persoană fizică și constituie singurul identificator pentru toate sistemele informatice care prelucrează date nominale privind persoana fizică.

Cu scopul de a reduce pe cât posibil introducerea de date false, s-a introdus și un algoritm de verificare a cnp-ului în urma preluării acestuia de pe buletin. Pentru a reuși implementarea acestui algoritm va trebui întâi înțeleasă structura celor treisprezece cifre din cnp și ce reprezintă fiecare.

Cnp-ul are următoarea structură:

|S| |AA| |LL| |ZZ| |JJ| |ZZZ| |C|

|C| - cifra de control

|ZZZ| - numărul de ordine atribuit persoanei

|JJ| - codul județului

|ZZ| - ziua nașterii

|LL| - luna nașterii

|AA| - anul nașterii

|S| - Cifra sexului (M/F) pentru:

1/2 - cetățeni romani născuți între 1 ian 1900 și 31 dec 1999

3/4 - cetățeni romani născuți între 1 ian 1800 și 31 dec 1899

5/6 - cetățeni romani născuți între 1 ian 2000 și 31 dec 2099

7/8 - rezidenți

9 - persoanele cu cetățenie străină

Algoritmul de validare al unui CNP

- Pas Preliminar: Se testează dacă codul respectă formatul unui cod CNP. Prima cifră să fie cuprinsă în intervalul 1 - 6 sau să fie 9 pentru străini. Următoarele șase cifre trebuie să constituie o dată calendaristică validă în formatul AALLZZ.
- Pas 1: Se folosește cheia de testare "279146358279". Primele douăsprezece cifre se înmulțesc pe rând de la stânga spre dreapta cu cifra corespunzătoare din cheia de testare.
- Pas 2: Cele douăsprezece produse obținute se adună și suma obținută se împarte la 11.
 - dacă restul împărțirii la 11 este mai mic ca 10, atunci acesta va reprezenta cifra de control.
 - dacă restul împărțirii este 10 atunci cifra de control este 1

Pentru un CNP valid cifra de control va trebui să coincidă cu cifra de pe poziția treisprezece din CNP-ul inițial.

Voters Candidates

Candidates Registration

First Name:

Last Name:

Username:

Password:

Personal Description:

Political Party

PSD

Register

Fig.17
Înregistrare
a unui nou
Candidat

Figura 17 reprezintă modalitatea de înregistrare pentru utilizatorul de tip candidat.

Partea de candidat este una puțin mai simplă, acesta va trebui să introducă toate informațiile personale, o scurtă descriere despre el și partidul politic ce vor fi afișate în dreptul candidatului atunci când se va afișa buletinul de vot în urma deschiderii alegerilor.

Atunci când un utilizator va dori să se logheze în aplicație, pe partea de back-end se va trata requestul de logare iar dacă username-ul și parola corespund celor din baza de date, accesul va fi

permis către o pagina corespunzătoare în funcție de tipul de user ce va solicita accesul.

III.3.3 Paginile principale ale aplicației

Existând trei tipuri de beneficiari ai aplicației, iar fiecărui beneficiar îi este asociată una din cele trei componente: componenta de admin, componenta pentru votant și componenta pentru candidat.

III.3.3.1 Componenta Admin

Din dorința de a separa votanții și candidații de lucrurile administrative precum crearea unui tip de candidatură, de începerea unei sesiuni de vot și de încheierea acesteia s-a ajuns la concluzia că este necesar un al treilea cont de utilizator, fiind vorba despre cel de administrator al aplicației.

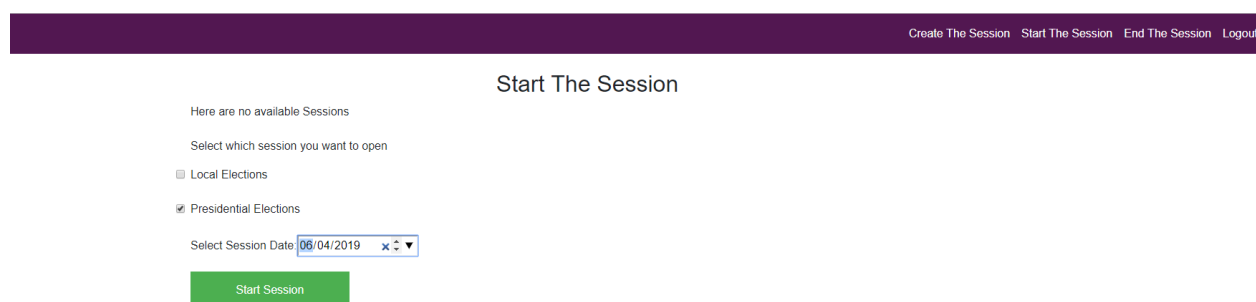


Fig. 18 Deschiderea unei noi sesiuni de vot

În această figură putem observa că în cazul în care nu este deschisă o sesiune, se propune administratorului să selecteze unul din cele două tipuri de alegeri, prezidențiale sau locale și data la care se vor desfășura alegerile cu scopul de a deschide o sesiune nouă. De asemenea, administratorul mai are și dreptul de a crea o nouă sesiune cât și să încheie sesiunea deschisă.

Pe prima pagină de Admin se vor putea vedea rezultatele alegerilor deschise în timp real. Acestea sunt afișate într-un Pie Chart, iar fiecare secțiune din Pie Chart va fi reprezentată cu o culoare sugestivă fiecărui partid pe care un candidat îl reprezintă.

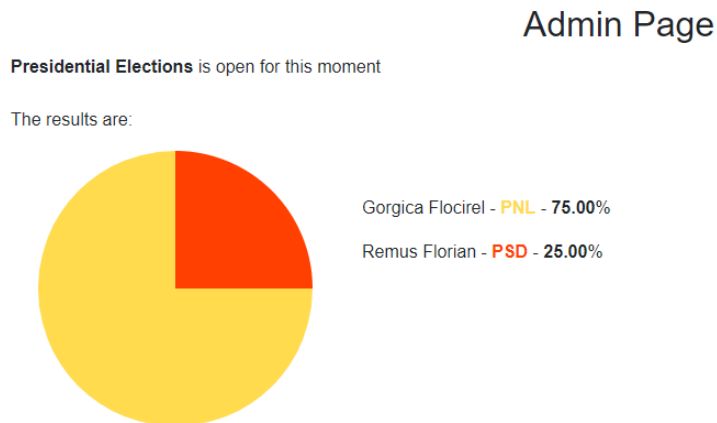


Fig. 19 Pie Chart reprezentând rezultatul alegerilor

III.3.3.2 Componenta Votant

Pentru utilizatorul de tip votant s-a creat această aplicație, cu scopul de a reduce timpul de vot în favoarea lui, așa că el este principalul beneficiar al aplicației.

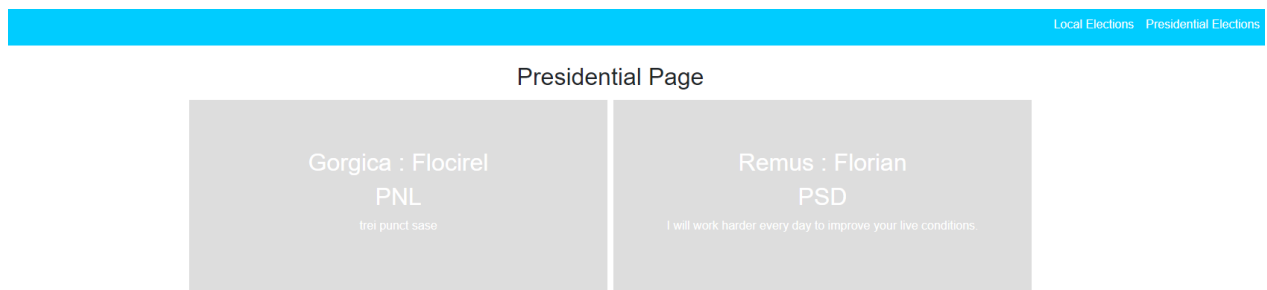


Fig. 20 Pagina de alegeri prezidențiale

În figura 20 putem observa modul simplu în care a fost gândit sistemul de votare. Atunci când se va intra pe o pagină cu tipul de alegeri dorit, în caz că sesiunea nu este deschisă se va afișa un mesaj corespunzător ce ne transmite că încă nu este deschisă sesiune selectată. În cazul în care

este deschisă sesiunea, se vor afișa toți candidații care sunt înscriși pentru acest tip de alegeri.

Diferența dintre cele două tipuri de alegeri, locale și prezidențiale se va face la nivel de implementare. Alegerile locale vor afișa doar candidații înscriși din același județ cu cel ce votează. Odată ce un utilizator se va loga, se va ști în permanentă de ce județ aparține deoarece lui i s-a asociat un cod de județ preluat din codul numeric personal în urma extragerii textului de pe buletin.

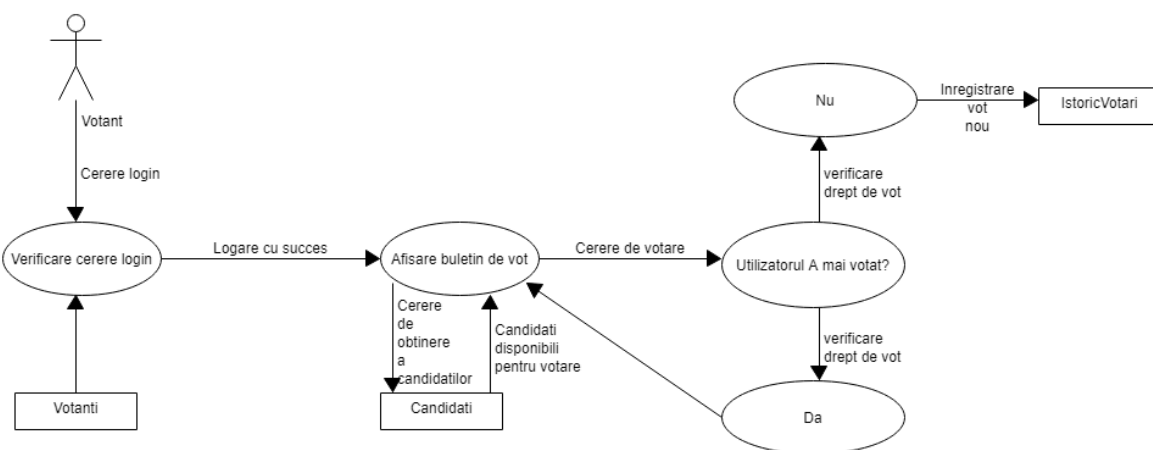


Fig.21 Diagrama sistemului de votare pentru un votant

Alte funcționalități ce ar putea fi oferite unui votant ar fi să i se ofere un istoric cu persoanele cu care a votat pe parcursul timpului și să i se ofere alternativa de a-și putea schimba votul în caz că se va răzgândi, dar doar înainte de a se termina sesiunea de votare.

III 3.3.3 Componenta Candidat

Această componentă are cele mai puține privilegii. Un nou candidat având posibilitatea de a se înregistra într-o sesiune de votare atunci când se va deschide una. O funcționalitate ce ar putea îmbunătăți pagina unui candidat ar fi adăugarea unei pagini de profil unde acesta își poate vedea și modifica datele personale cât și informațiile politice despre el.

Concluzia lucrării

Această lucrare de licență prezintă principalele motive pentru care odată cu dezvoltarea tehnologiei trebuie aduse îmbunătățiri și într-o activitate precum votarea, care la o primă vedere pare simplă. Dar în contradicție cu simplitatea acestei activități, stă importanța ei, deoarece asigurându-se o prezență mai mare la vot prin această idee de votare, acest lucru poate schimba viitorul unei națiuni.

Contribuția personală în această aplicație constă în migrarea sistemului de vot clasic, unde se alege un candidat de pe buletinul de vot, într-un mediu online. Această migrare vine și cu o serie de lucruri pozitive. Principalele beneficii care trebuie amintite ar fi îmbunătățirea prezenței la vot, deoarece oricine are acces la internet poate intra pe aplicație să își exercite dreptul. Acest fapt oferă oamenilor posibilitatea de a nu se mai deplasa până la secțiunile de votare. Un alt beneficiu major va fi forța de muncă ce trebuie mobilizată pentru a putea organiza o sesiune de votare clasică, ce trebuie să se ocupe de buna desfășurare a alegerilor, implicând oameni ce aduc urnele de vot, cei ce împart buletine de vot, dar și cei care numără voturile. Prin folosirea acestei aplicații, toate aceste costuri vor fi reduse, deoarece doar administratorul aplicației va deschide și va închide o sesiune, iar pe parcursul sesiunii de votare, voturile vor fi monitorizate automat în timp real.

Baza oferită de această aplicație poate fi ușor extinsă pentru a oferi o gamă mai mare de funcționalități, precum configurarea pe mai multe tipuri de alegeri. De asemenea, pe partea de UI există multe posibilități de îmbunătățire a experienței utilizatorului prin adăugarea de elemente interactive, adăugarea unei imagini în prezentarea unui candidat, dar și opțiunea de editare a profilului unui utilizator.

În concluzie, s-a dovedit că această procedură de votare, se poate implementa și într-o aplicație din mediul online, putând să substituie cu succes clasică metodă de votare, dar și aducând un plus în privința organizării alegerilor.

Bibliografie

[1] Aplicații asemănătoare

ElectionBuddy <https://electionbuddy.com/>

[2] Tesseract <https://www.c-sharpcorner.com/article/ocr-using-tesseract-in-C-Sharp/>

[3] Documentație Git <https://teaching.alexcoman.com/resurse/tutorial/git/notiuni-introductive/>

[4] Documentație Arhitectură pe mai multe nivele(n-tier) - <https://stackify.com/n-tier-architecture/>

[5] Documentație Asp.Net Core 2.1 <https://docs.microsoft.com/ro-ro/aspnet/core/web-api/?view=aspnetcore-2.1>

[6] Documentație React

Cum funcționează React-<https://www.talkingdotnet.com/create-react-app-visual-studio-2017-and-asp-net-core-2-2/>

Tutorial ChartPie <https://medium.com/@rkhatia89/how-to-make-pie-charts-in-react-with-chart-js-64edd03fc922>

[7] Algoritm verificare cnp <http://www.dgnconsulting.ro/articole-validare-cnp.html>

[8] Tutorial pentru construire a navbar-ului și sidebar-ului
<https://www.youtube.com/watch?v=l6nmysZKHFU>

[9] Exemple toast <https://github.com/fkhadra/react-toastify>

Link-uri sursa

[1] Pagina oficială a aplicației ElectionBuddy <https://electionbuddy.com/process>

[2] Diagrama pe 3-nivele https://www.google.com/search?q=n-tier+architecture&rlz=1C1GCEA_enRO843RO843&source=lnms&tbm=isch&sa=X&ved=0ahUKewju76-M0vfiAhXCpo8KHx4EBI8Q_AUIECgB&biw=1920&bih=888#imgsrc=iClR9Ky5hcrxrM:

Anexa 1

Structura unui proiect React

La generarea unui proiect în React putem observa structura și componentele din care aceasta este formată.

- Subdirectorul **ClientApp** care este standard pentru o aplicație React
- Folderul **src** unde găsim componentele ce alcătuiesc interfața cu utilizatorul
- Principala componentă este **App.js** unde fiecare componentă este înregistrată
- **Index.js** oferă intrarea în aplicație și lansează serviciul registerServiceWorker ce are grijă de memorarea în cache și actualizarea fișierelor pentru utilizatorul final.

Structura simplă a codului ajută în dezvoltarea aplicațiilor deoarece partea de UI este separată în componente. În general vorbind, pentru a scrie aplicații React se scriu componente React care corespund diferitelor elemente de interfață, apoi acestea sunt organizate în componente de nivel superior ce vor forma structura aplicației dorite.

Spre deosebire de predecesorii săi, React nu lucrează direct cu Document Object Model(DOM) din browser, ci se folosește de un virtual DOM. React Virtual DOM se afla în întregime în memorie și este o reprezentare a DOM-ului din browser. Din acest motiv, când se scrie o componentă React, aceasta nu este scrisă direct în DOM ci se scrie o componentă pe care React o va transforma în DOM.

```
class HelloWorld extends React.Component {  
  render() {  
    return (  
      <h1 className='large'>Hello World</h1>  
    );  
  }  
}
```

Fig.22 : Structura de baza a unei componente în React

După cum se poate observa în figura 2, în componenta React, se afla o funcție render ce specifică ce output HTML va avea componenta noastră.

JavaScript eXtension sau mai bine cunoscută sub numele de JSX, este o extensie a React-ului, ce permite scrierea de cod JavaScript ce va arăta ca și cod HTML. Componenta prezentată mai sus, arată ca și cum ar returna cod HTML, dar de fapt este JSX.

Configurarea proiectului cu Entity Framework

Pentru a se putea folosi EF în aplicație, este necesară instalarea a două pachete din Nuget Packages (Microsoft.EntityFrameworkCore.SqlServer și Microsoft.EntityFrameworkCore). Apoi, va trebui creată o clasă context ce va fi extinsă din DbContext class ce ține de EF. Sub această clasă, se va înregistra câte fiecare model, ce va reprezenta o tabelă din baza de date.

```
public class ApplicationDbContext : DbContext
{
    public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options) : base(options)
    {
    }

    public DbSet<City> Cities { get; set; }
    public DbSet<User> Users { get; set; }
    public DbSet<VotingHistory> VotingHistory { get; set; }
    public DbSet<VotingSession> VotingSessions { get; set; }
}
```

Fig.23: Clasa context a proiectului

Această imagine descrie clasa context ce conține un DbSet pentru fiecare model utilizat în aplicație(City, User, VotingHistory, VotingSession). Acest context reprezentând baza de date.

Pentru a duce până la final configurarea proiectului cu EF, mai este necesară adăugarea în Startup.cs a stringului de conectare la baza de date. Aplicația creată se folosește de un server SQL local, iar stringul de conectare va fi găsit în appsettings.json și va indica către o baza de date locală. Atunci când se va dori modificarea structurii bazei de date, EF utilizează migrări. Când se execută o modificare în model, vom adăuga o migrare. Aceasta va genera codul SQL necesar modificării structurii.

Comenzile necesare unui update la BD sunt:

- Add-Migration NumeMigrare ce va genera codul SQL necesar update-ului
- Update-Database - în urma rulării acestei comenzi se va modifica structura bazei de date.

Crearea unui repository nou

Pașii ce au fost urmați pentru a crea un nou repository:

1. Crearea unui cont de github la următorul link: <https://github.com/>
2. Crearea unui nou repository. Se poate face atât online de pe linkul anterior cât și de la linia de comandă, local. Dacă se optează pentru o variantă creată local va trebui în prealabil git-ul să fie instalat apoi, va trebui să se creeze un folder unde se va dori să se țină codul sursa, iar apoi, de la linia de comandă va trebui rulată comanda git init Nume-Repository.
3. Instalarea git de la următorul link: <https://git-scm.com/downloads>
4. După ce repository-ul s-a creat cu succes va trebui să configurăm contul nostru de github prin introducerea următoarelor comenzi
 - **git config --global user.email nume@exemplu.com**
 - **git config --global user.name „nume”**

Odată cu crearea unui repository se va începe și lucrul la aplicație, acest fapt ducând la dorința de a salva codului pe GitHub. Pentru a salva fișierele modificate vor fi câteva comenzi ce vor trebui rulate în Command Prompt sau Git Bash în interiorul repository-ului.

- **git status** - atunci când se va dori verificarea fișierelor modificate de la ultimul commit.
- **git add nume-fișier** când se va dori adăugarea unui fișier sau **git add .** pentru adăugarea tuturor fișierelor modificate de la ultimul commit, în staging pentru un viitor commit.
- **git commit -m „NumeComit”** pentru a adăuga schimbările permanent în repository însoțit de parametrul -m dacă se va dori să se ofere respectivului commit un mesaj dorit.

Într-un repository de git există și posibilitatea de a lucra pe mai multe branches(ramuri)

cu scopul lucrului în paralel la două versiuni diferite. De exemplu, puteți crea un branch "de testare" și un branch "de producție" pentru a urmări modificările pentru diferite versiuni. În această aplicație, s-a lucrat pe un singur branch, de development, deoarece o singură persoană a lucrat în dezvoltarea aplicației și nu a fost necesar lucrul pe branch separat.