

## UNIVERSITATEA DE VEST DIN TIMIȘOARA

**FACULTATEA DE MATEMATICĂ ŞI INFORMATICĂ PROGRAMUL DE STUDII DE LICENȚĂ:**

**LUCRARE DE LICENȚĂ**

**COORDONATOR ȘTIINȚIFIC: ABSOLVENT:**

Conf. Univ. Dr. Victoria Iordan și Asist. Univ. Drd. Sebastian-Aurelian Ștefănigă.

Mureșan Darius Alin

## UNIVERSITATEA DE VEST DIN TIMIȘOARA

**FACULTATEA DE MATEMATICĂ ŞI INFORMATICĂ PROGRAMUL DE STUDII DE LICENȚĂ:**

Sistem de votare online sub formă de aplicație web

**COORDONATOR ȘTIINȚIFIC: ABSOLVENT:**

Conf. Univ. Dr. Victoria Iordan și Asist. Univ. Drd. Sebastian-Aurelian Ștefănigă.

Mureșan Darius Alin

**CUPRINS**

1. Introducere
   1. Rezumat pag. 4

1.2 Motivație ……………………………………………………………………………. pag. 5

1.3 Obiective ……………………………………………………………………………. pag. 6

1. Metodolgie

2.1. Concepte …………………………………………………………………………… pag. 7

2.2. Algoritmi …………………………………………………………………………… pag. 9

3. Aplicația …………………………………………………………………………………… pag. 10

4. Concluzii ………………………………………………………………………………….. pag. 29 5. Direcții de dezvoltare …………………………………………………………………….. pag. 30

6. Bibliografie ……………………………………………………………………………….. pag. 32

**Rezumat**

Avand in vedere evolutia tehnologica din ultimii ani si cresterea numarului de votanti in tara noastra si in diaspora, am ales sa gandesc un concept de sistem de votare online care ar putea sa fie implementat in curand pentru alegerile din viitor.

Fiecare cetatean cu drept vot poate sa se inregistreze pentru sistemul de vot online trimitand CNP-ul si o poza cu un act de identitate valabil, preferabil buletinul. Cand cererea este aprobata acesta primeste un cod unic pe care trebuie sa il stocheze

intr-un loc cunoscut numai de el deoarece cu ajutorul acestuia va putea sa voteze.

Aplicatia contine:

-Sistem de inregistrare cont de vot cu ajutorul CNP-ului

-Sistem de logare cu ajutorul CNP-ului si codului unic

-Sistem de votare cu ajutorul CNP-ului

-Posibilitatea de a analiza rezultatele alegerilor

-Posibilitatea de a vedea optiunile de vot ale utilizatorului

**Motivație**

Avand in vedere evolutia tehnologica din ultimii ani si cresterea numarului de votanti in tara noastra si in diaspora, am ales sa gandesc un concept de sistem de votare online care ar putea sa fie implementat in curand pentru alegerile din viitor.

Cu toate ca alegerile online ar reprezenta totusi o provocare nu doar din punct de vedere tehnologic, avand in vedere ca nici un sistem tehnologic nu poate fi 100% protejat, lasand altfel loc de frauda, de asemenea incapatanarea unor anumiti indivizi de a adopta noile tehnologii sau a unor dispozitive trebuie sa incercam sa implementam un sistem cat mai eficient si scalabil din toate punctele de vedere deoarece avantajele sunt totusi exponentiale iar cheltuielile aferente organizarii unor alegeri ar putea fi reduse exponential, banii economisiti putand fi folositi ulterior pentru dezvoltarea tarii.

**Obiective**

Obiectivul acestui proiect este de a construi un prototip de sistem de votare online care sa demonstreze aceste concept si care sa fie, eventual, implementat in viitor.



**Aplicatia**

**Tehnologii folosite**

**HTML**

HyperText Markup Language (HTML) este un limbaj de marcare utilizat pentru crearea paginilor web ce pot fi afișate într-un browser (sau navigator). Scopul HTML este mai degrabă prezentarea informațiilor – paragrafe, fonturi, tabele ș.a.m.d. – decât descrierea semanticii documentului.

Specificațiile HTML sunt dictate de World Wide Web Consortium (W3C).

HTML este o formă de marcare orientată către prezentarea documentelor text pe o singura pagină, utilizând un software de redare specializat, numit agent utilizator HTML, cel mai bun exemplu de astfel de software fiind browserul web. HTML furnizează mijloacele prin care conținutul unui document poate fi adnotat cu diverse tipuri de metadate și indicații de redare. Indicațiile de redare pot varia de la decorațiuni minore ale textului, cum ar fi specificarea faptului că un anumit cuvânt trebuie subliniat sau că o imagine trebuie introdusă, până la scripturi sofisticate, hărți de imagini și formulare. Metadatele pot include informații despre titlul și autorul documentului, informații structurale despre cum este împărțit documentul în diferite segmente, paragrafe, liste, titluri etc. și informații cruciale care permit ca documentul să poată fi legat de alte documente pentru a forma astfel hiperlink-uri (sau web-ul).

HTML este un format text proiectat pentru a putea fi citit și editat de oameni utilizând un editor de text simplu. Totuși scrierea și modificarea paginilor în acest fel solicită cunoștințe solide de HTML și este consumatoare de timp. Editoarele grafice (de tip WYSIWYG) cum ar fi Macromedia Dreamweaver, Adobe GoLive sau Microsoft FrontPage permit ca paginile web sa fie tratate asemănător cu documetele Word, dar cu observația că aceste programe generează un cod HTML care este de multe ori de proastă calitate.

HTML se poate genera direct utilizând tehnologii de codare din partea serverului cum ar fi PHP, JSP sau ASP. Multe aplicații ca sistemele de gestionare a conținutului, wiki-uri și forumuri web generează pagini HTML.

HTML este de asemenea utilizat în e-mail. Majoritatea aplicațiilor de e-mail folosesc un editor HTML încorporat pentru compunerea e-mail-urilor și un motor de prezentare a e-mail-urilor de acest tip. Folosirea e-mail-urilor HTML este un subiect controversat și multe liste de mail le blochează intenționat.

HTML este prescurtarea de la Hyper Text Mark-up Language si este codul care sta la baza paginilor web.

Paginile HTML sunt formate din etichete sau tag-uri și au extensia „.html” sau „.htm”. În marea lor majoritate aceste etichete sunt pereche, una de deschidere <eticheta> și alta de închidere </eticheta>, mai există și cazuri în care nu se închid, atunci se folosește <eticheta />. Navigatorul web interpretează aceste etichete afișând rezultatul pe ecran. HTML-ul este un limbaj care nu face deosebire între litere majuscule și minuscule.

Pagina principala a unui domeniu este fisierul „index.html” respectiv „index.htm”. Această pagină este setată a fi afișată automat la vizitarea unui domeniu.

De exemplu la vizitarea domeniului www.nume.ro este afișată pagina www.nume.ro/index.html.

Unele etichete permit utilizarea de atribute care pot avea anumite valori:

<eticheta atribut="valoare"> ... </eticheta>

**CSS**

CSS (Cascading Style Sheets) este un standard pentru formatarea elementelor unui document HTML. Stilurile se pot atașa elementelor HTML prin intermediul unor fișiere externe sau în cadrul documentului, prin elementul <style> și/sau atributul style. CSS se poate utiliza și pentru formatarea elementelor XHTML, XML și SVGL.

CSS este unul dintre limbajele principale ale webului deschis și este standardizat în browsere conform specificațiilor W3C. Dezvoltat pe niveluri, CSS1 este acum depășit, se recomandă CSS2.1, iar CSS3, acum împărțit în module mai mici, este în curs de standardizare.

CSS3 reprezintă un upgrade ce aduce câteva atribute noi și ajută la dezvoltarea noilor concepte in webdesign.

Unele dintre cele mai importante segmente (module) noi adăugate acestui standard pentru formatarea elementelor HTML aduc un plus considerabil in dezvoltarea activități webdesign.

Mai jos sunt prezente in listă cele mai importante modulele adăugate in CSS3:

-Selectors

-Box Model

-Backgrounds and Borders

-Image Values and Replaced Content

-Text Effects

-2D/3D Transformations

-Animations

-Multiple Column Layout

-User Interface

Deși au apărut unele deficiente de compatibilitate intre browsere, majoritatea proprietăților CSS3 au fost implementate cu succes in variantele browserelor noi.

**Javascript**

JavaScript (JS) este un limbaj de programare orientat obiect bazat pe conceptul prototipurilor.[5] Este folosit mai ales pentru introducerea unor funcționalități în paginile web, codul JavaScript din aceste pagini fiind rulat de către browser. Limbajul este binecunoscut pentru folosirea sa în construirea siturilor web, dar este folosit și pentru accesul la obiecte încapsulate (embedded objects) în alte aplicații. A fost dezvoltat inițial de către Brendan Eich de la Netscape Communications Corporation sub numele de Mocha, apoi LiveScript, și denumit în final JavaScript.

În ciuda numelui și a unor similarități în sintaxă, între JavaScript și limbajul Java nu există nicio legătură. Ca și Java, JavaScript are o sintaxă apropiată de cea a limbajului C, dar are mai multe în comun cu limbajul Self decât cu Java.

Până la începutul lui 2005, ultima versiune existentă a fost JavaScript 1.5, care corespunde cu Ediția a 3-a a ECMA-262, ECMAScript, cu alte cuvinte, o ediție standardizată de JavaScript. Versiunile de Mozilla începând cu 1.8 Beta 1 au avut suport pentru E4X, care este o extensie a limbajului care are de a face cu XML, definit în standardul ECMA-357. Versiunea curentă de Mozilla, 1.8.1 (pe care sunt construite Firefox și Thunderbird versiunile 2.0) suportă JavaScript versiunea 1.7.

Cea mai des întâlnită utilizare a JavaScript este în scriptarea paginilor web. Programatorii web pot îngloba în paginile HTML script-uri pentru diverse activități cum ar fi verificarea datelor introduse de utilizatori sau crearea de meniuri și alte efecte animate.

Browserele rețin în memorie o reprezentare a unei pagini web sub forma unui arbore de obiecte și pun la dispoziție aceste obiecte script-urilor JavaScript, care le pot citi și manipula. Arborele de obiecte poartă numele de Document Object Model sau DOM. Există un standard W3C pentru DOM-ul pe care trebuie să îl pună la dispoziție un browser, ceea ce oferă premiza scrierii de script-uri portabile, care să funcționeze pe toate browserele. În practică, însă, standardul W3C pentru DOM este incomplet implementat. Deși tendința browserelor este de a se alinia standardului W3C, unele din acestea încă prezintă incompatibilități majore, cum este cazul Internet Explorer.

O tehnică de construire a paginilor web tot mai întâlnită în ultimul timp este AJAX, abreviere de la „Asynchronous JavaScript and XML”. Această tehnică constă în executarea de cereri HTTP în fundal, fără a reîncărca toată pagina web, și actualizarea numai anumitor porțiuni ale paginii prin manipularea DOM-ului paginii. Tehnica AJAX permite construirea unor interfețe web cu timp de răspuns mic, întrucît operația (costisitoare ca timp) de încărcare a unei pagini HTML complete este în mare parte eliminată.

**Jquery**

jQuery este o platformă de dezvoltare JavaScript, concepută pentru a ușura și îmbunătăți procese precum traversarea arborelui DOM în HTML,[2] managementul inter-browser al evenimentelor, animații și cereri tip AJAX. jQuery a fost gândit să fie cât mai mic posibil, disponibil în toate versiunile de browsere importante existente, și să respecte filosofia "Unobtrusive JavaScript". Biblioteca a fost lansată in 2006 de către John Resig [3].

jQuery se poate folosi pentru a rezolva următoarele probleme specifice programării web:

-selecții de elemente în arborele DOM folosind propriul motor de selecții open source Sizzle, un proiect născut din jQuery [4]

-parcurgere și modificarea arborelui DOM (incluzând suport pentru selectori CSS 3 și XPath simpli)

-înregistrarea și modificarea evenimentelor din browser

-manipularea elementelor CSS

-efecte și animații

-cereri tip AJAX

-extensii

-utilităti - versiunea browser-ului, funcția each.

**AJAX**

AJAX este o metodologie de a trimite date catre server fara a reinitializa pagina web actuala. Aceasta este o extensie asincrona javascript dar se poate folosi si cu ajutorul jquery.



**Mongo DB**

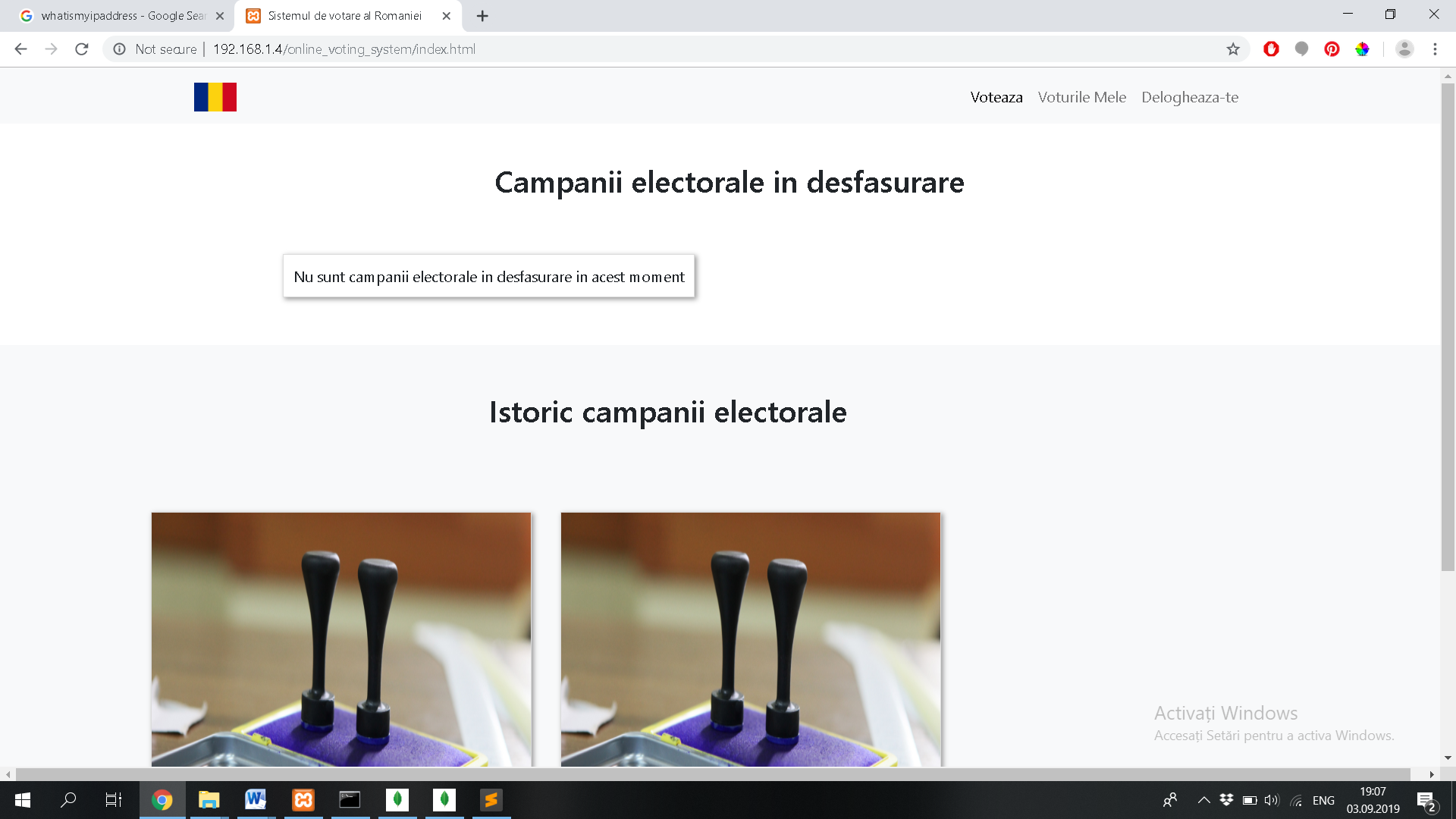
MongoDB este o bază de date NoSQL open-source orientată pe documente. Acestă bază de date beneficiază de suport din partea companiei 10gen. MongoDB face parte din familia de sistemelor de baze de date NoSQL. Diferența principală constă în faptul că stocarea datelor nu se face folosind tabele precum într-o bază de date relațională, MongoDB stochează datele sub formă de documente JSON cu scheme dinamice.

MongoDB este o bază de date open-source NoSQL scrisă în C++. Aceasta poate conține mai multe baze de date, colecții și indecși. În unele cazuri (baze de date și colecții ) aceste obiecte pot fi create implicit. Odată create, ele se găsesc în catalogul sistemului db.systems.collection, db.system.indexes. Colecțiile conțin documente (BSON). Aceste documente conțin la rândul lor mai multe câmpuri. În MongoDB nu există câmpuri predefinite spre deosebire de bazele de date relaționale, unde există coloanele care sunt definite în momentul în care tabelele sunt create. Nu există schemă pentru câmpurile dintr-un document, acestea precum și tipurile lor pot varia. Astfel nu există operația de „alter table” pentru adăugare de coloane. În practică este obișnuit ca o colecție să aibă o structură omogenă, deși nu este o cerință, colecțiile putând avea structuri diferite. Această flexibilitate presupune ușurință în migrarea și modificarea imaginii de ansamblu asupra datelor.

**Structura Aplicatiei**

**Index.html ( pagina principala )**

**Descriere**



Pe pagina principala se afla informatiile despre campaniile electorale in desfasurare si

campaniile incheiate, de asemenea in partea superioara a ecranului avem bara de

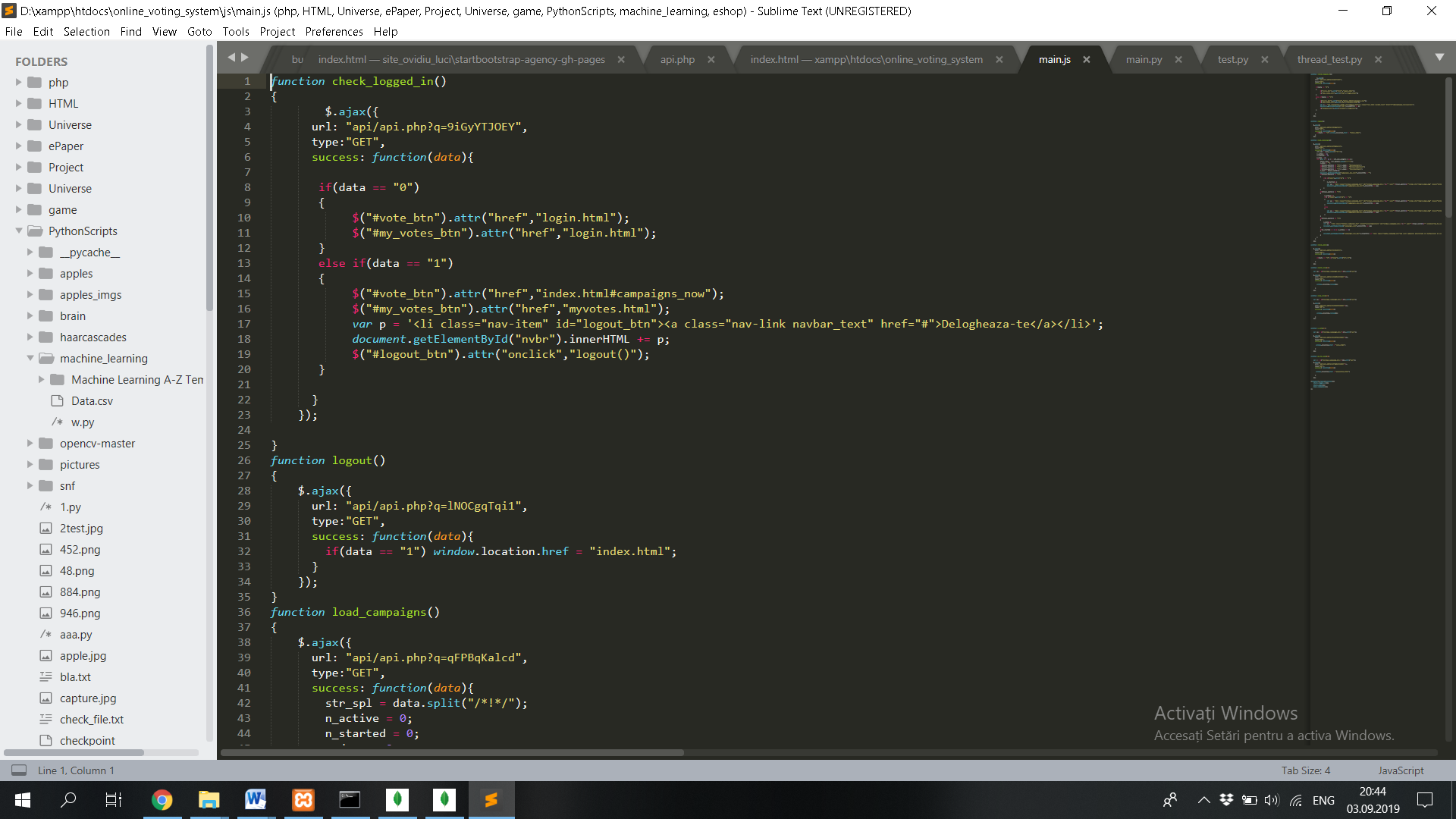
navigatie ( „navbar” ) pe care avem butonul „Voteaza” care ne va duce catre

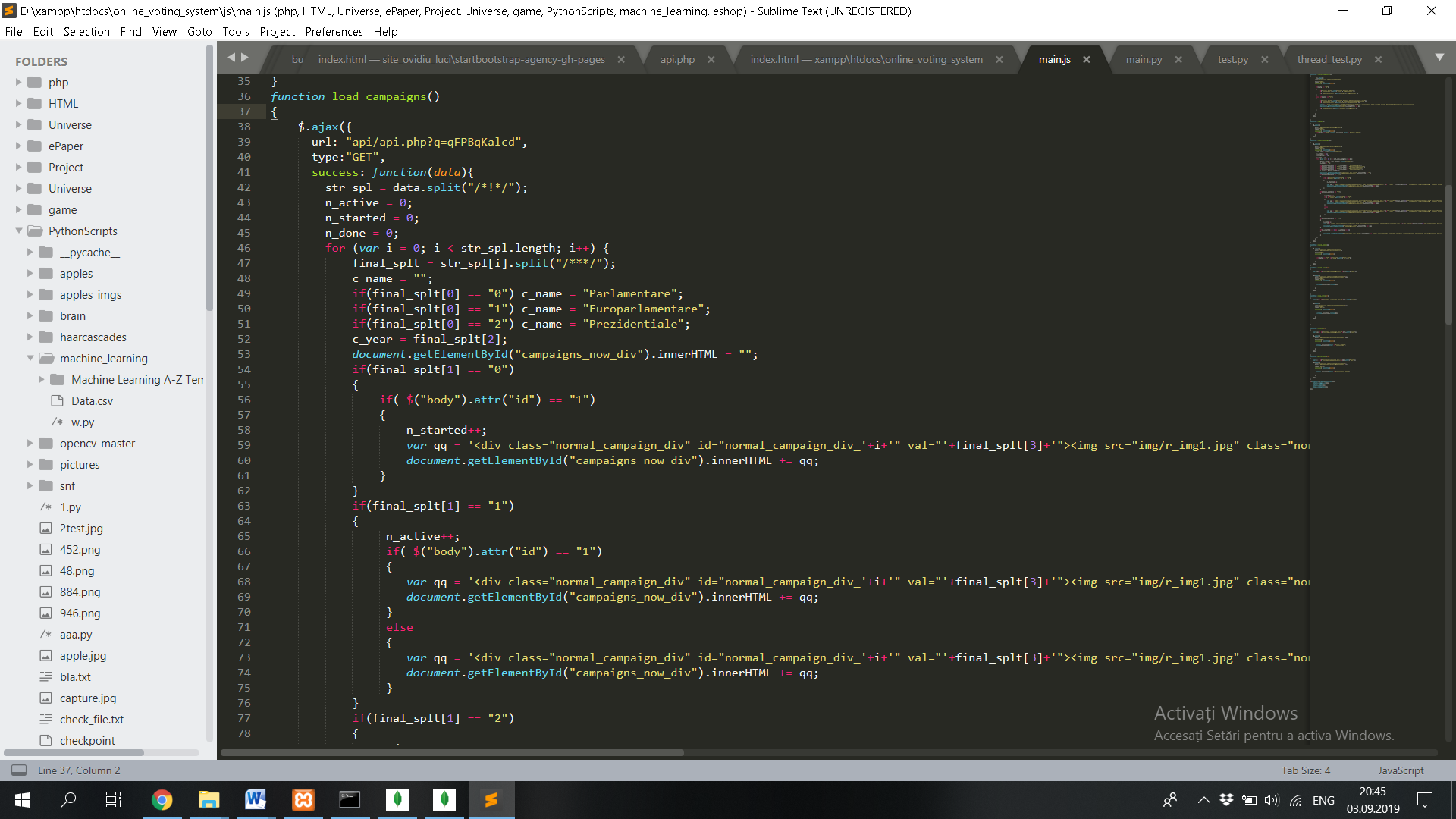
sectiunea cu campaniile de votare in desfasurare, butonul „Voturile mele” care ne va

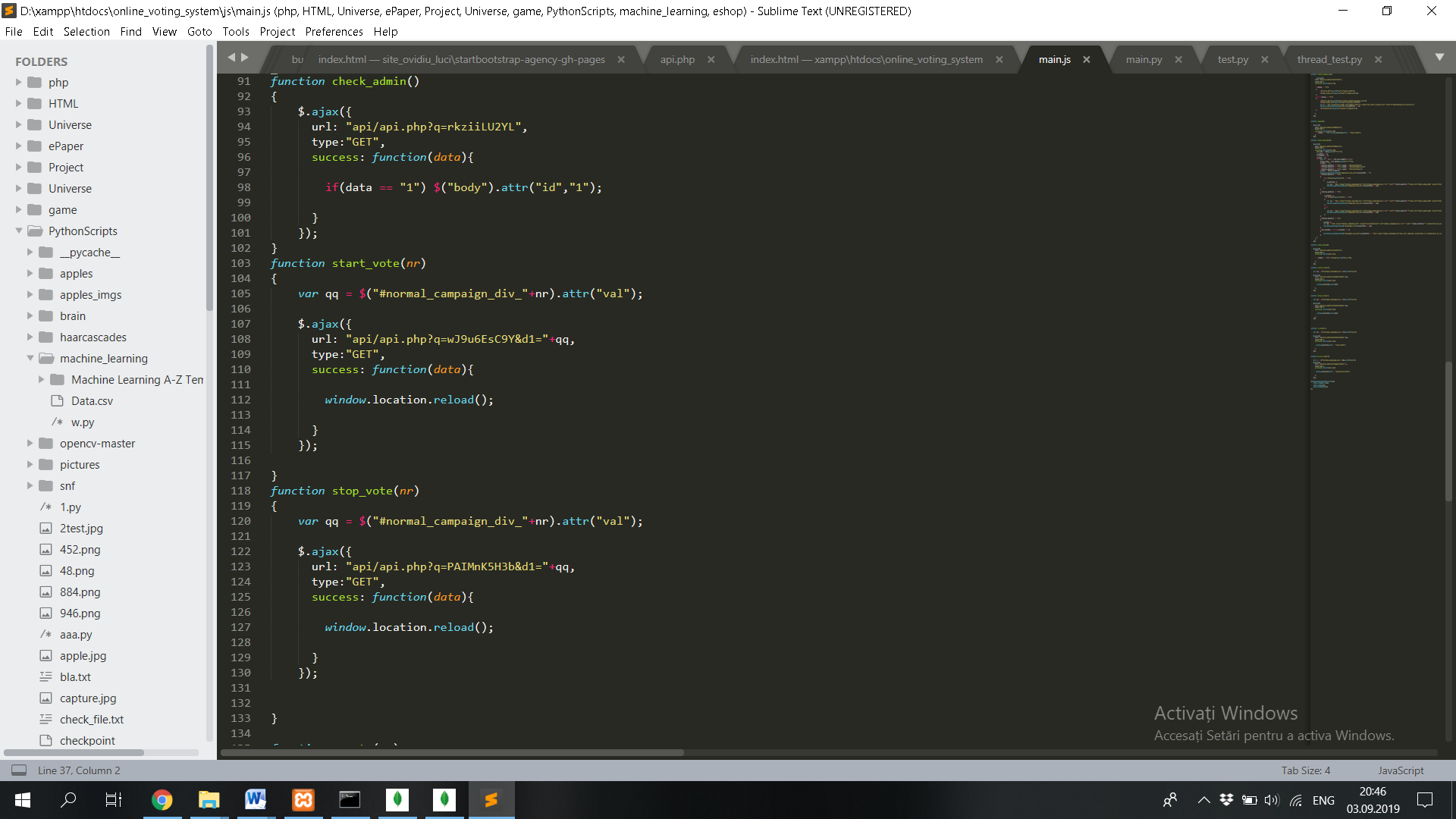
duce la pagina „myvotes.html”, pagina care afiseaza istoricul voturilor utilizatorului.

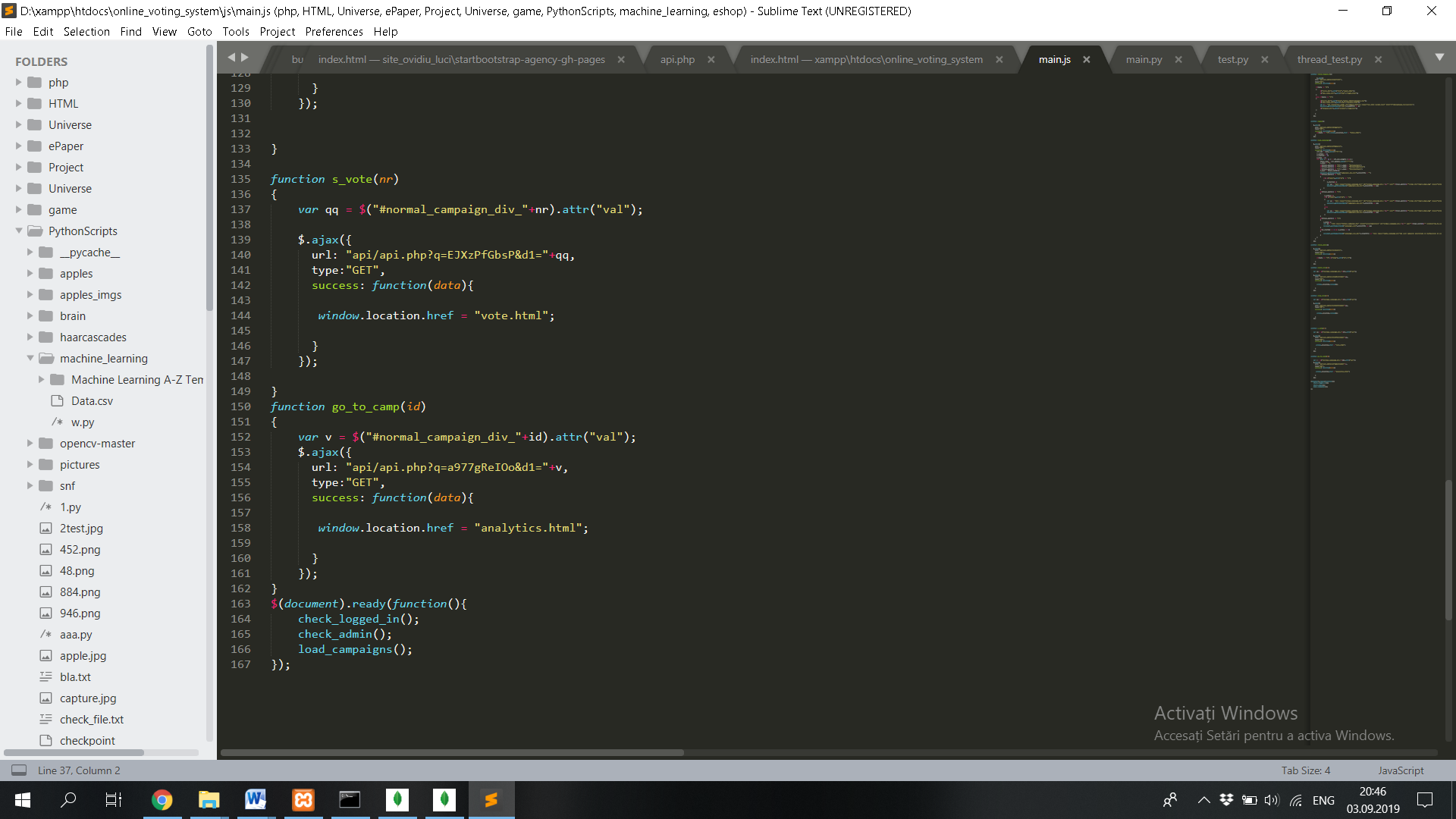
De asemenea daca este logat, utilizatorul va putea sa vada si butonul „Delogheaza-te”.

**Structura codului**









Atunci cand documentul este incarcat se apeleaza automat cu ajutorul jQuery 3 functii:check\_logged\_in();check\_admin();load\_campaigns();

function check\_logged\_in()

{

**$.ajax({**

**url: "api/api.php?q=9iGyYTJOEY",**

**type:"GET",**

**success: function(data){**

**if(data == "0")**

**{**

**$("#vote\_btn").attr("href","login.html");**

**$("#my\_votes\_btn").attr("href","login.html");**

**}**

**else if(data == "1")**

**{**

**$("#vote\_btn").attr("href","index.html#campaigns\_now");**

**$("#my\_votes\_btn").attr("href","myvotes.html");**

**var p = '<li class="nav-item" id="logout\_btn"><a class="nav-link navbar\_text" href="#">Delogheaza-te</a></li>';**

**document.getElementById("nvbr").innerHTML += p;**

**$("#logout\_btn").attr("onclick","logout()");**

**}**

**}**

**});**

**}**

Aceasta functie trimite un request Ajax de tipul GET catre API ( reprezentat prin pagina api.php ) si primeste ca raspuns „1” daca variabila $\_SESSION[’ cnp’] e diferita de „none” si „0” in caz diferit. Daca primeste „1” adauga la navbar butonul „Delogheaza-te”.

**function check\_admin()**

**{**

**$.ajax({**

**url: "api/api.php?q=rkziiLU2YL",**

**type:"GET",**

**success: function(data){**

**if(data == "1") $("body").attr("id","1");**

**}**

**});**

**}**

Aceasta functie trimite un request de tip GET catre API pentru a verifica daca contul logat este administrator, daca primeste raspunsul „1” atunci se face un update la tag-ul body, caruia i se adauga id-ul 1,acest indicativ semnifica faptul ca sesiunea curenta are detaliile administratorului.

**function load\_campaigns()**

**{**

**$.ajax({**

**url: "api/api.php?q=qFPBqKalcd",**

**type:"GET",**

**success: function(data){**

**str\_spl = data.split("/\*!\*/");**

**n\_active = 0;**

**n\_started = 0;**

**n\_done = 0;**

**for (var i = 0; i < str\_spl.length; i++) {**

**final\_splt = str\_spl[i].split("/\*\*\*/");**

**c\_name = "";**

**if(final\_splt[0] == "0") c\_name = "Parlamentare";**

**if(final\_splt[0] == "1") c\_name = "Europarlamentare";**

**if(final\_splt[0] == "2") c\_name = "Prezidentiale";**

**c\_year = final\_splt[2];**

**document.getElementById("campaigns\_now\_div").innerHTML = "";**

**if(final\_splt[1] == "0")**

**{**

**if( $("body").attr("id") == "1")**

**{**

**n\_started++;**

**var qq = '<div class="normal\_campaign\_div" id="normal\_campaign\_div\_'+i+'" val="'+final\_splt[3]+'"><img src="img/r\_img1.jpg" class="normal\_campaign\_div\_img"><div style="margin-top: 30px;"></div><h4 style="text-align: center;">Alegeri '+c\_name+" "+c\_year+'</h6><div style="margin-top: 30px;"></div><button class="btn btn-success c\_form\_btn" onclick="start\_vote('+i+')">Start vot</button></div>';**

**document.getElementById("campaigns\_now\_div").innerHTML += qq;**

**}**

**}**

**if(final\_splt[1] == "1")**

**{**

**n\_active++;**

**if( $("body").attr("id") == "1")**

**{**

**var qq = '<div class="normal\_campaign\_div" id="normal\_campaign\_div\_'+i+'" val="'+final\_splt[3]+'"><img src="img/r\_img1.jpg" class="normal\_campaign\_div\_img"><div style="margin-top: 30px;"></div><h4 style="text-align: center;">Alegeri '+c\_name+" "+c\_year+'</h6><div style="margin-top: 30px;"></div><button class="btn btn-danger c\_form\_btn" onclick="stop\_vote('+i+')">Stop vot</button></div>';**

**document.getElementById("campaigns\_now\_div").innerHTML += qq;**

**}**

**else**

**{**

**var qq = '<div class="normal\_campaign\_div" id="normal\_campaign\_div\_'+i+'" val="'+final\_splt[3]+'"><img src="img/r\_img1.jpg" class="normal\_campaign\_div\_img"><div style="margin-top: 30px;"></div><h4 style="text-align: center;">Alegeri '+c\_name+" "+c\_year+'</h6><div style="margin-top: 30px;"></div><button class="btn btn-success c\_form\_btn" onclick="s\_vote('+i+')" >Voteaza</button></div>';**

**document.getElementById("campaigns\_now\_div").innerHTML += qq;**

**}**

**}**

**if(final\_splt[1] == "2")**

**{**

**n\_done++;**

**var qq = '<div class="normal\_campaign\_div" style="cursor:pointer;" id="normal\_campaign\_div\_'+i+'" val="'+final\_splt[3]+'" onclick="go\_to\_camp('+i+');";><img src="img/r\_img1.jpg" class="normal\_campaign\_div\_img"><div style="margin-top: 30px;"></div><h4 style="text-align: center;">Alegeri '+c\_name+" "+c\_year+'</h6><div style="margin-top: 30px;"></div></div>';**

**document.getElementById("campaigns\_old").innerHTML += qq;**

**}**

**if(n\_started == 0 && n\_active == 0)**

**{**

**document.getElementById("campaigns\_now\_div").innerHTML += '<div class="empty\_campaign\_div">Nu sunt campanii electorale in desfasurare in acest moment</div>';**

**}**

**}**

**}**

**});**

**}**

Aceasta functie, dupa cum semnifica si numele, incarca campaniile existente din baza de date, face un request catre API si primeste inapoi un string cu toate informatiile necesare sub forma „detaliu\_campanie\_0[0]/\*\*\*/detaliu\_campanie\_0[1] /\*\*\*/..../\*!\*/ detaliu\_campanie\_1[0]/\*\*\*/detaliu\_campanie\_1[1] /\*\*\*/..../\*!\*/....”.

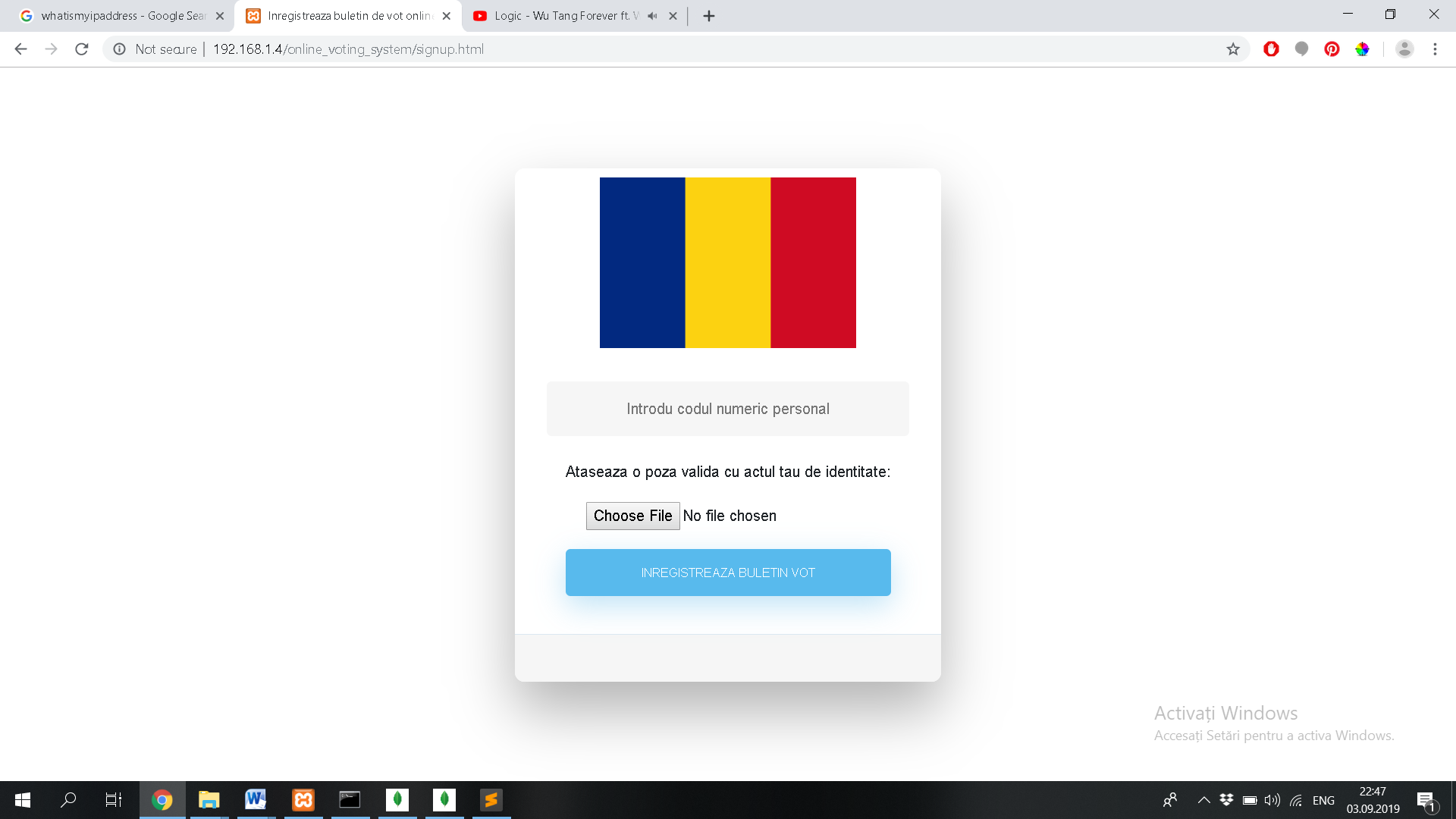
Am ales aceasta metoda de a transmite datele deoarece este mai sigura decat varianta JSON, acest string poate fi encryptat pentru o securitate sporita, de asemenea putem alege orice combinatie pentru a trimite datele cum dorim, din nou aceasta caracteristica sporind securitatea trimiterii datelor.

Dupa ce au fost primite, datele sunt „sparte” intr-un array de stringuri fiind separate cu ajutorul functii „split()” din Javascript, pentru fiecare campanie se itereaza fiecare string din array si se mai „sparge” inca o data cu ajutorul „split()” pentru a prelua detaliile campaniilor, apoi pentru fiecare campanie este updatat div-ul cu id-ul „campaigns\_now\_div” cu un nou div care contine toate detaliile campaniei.

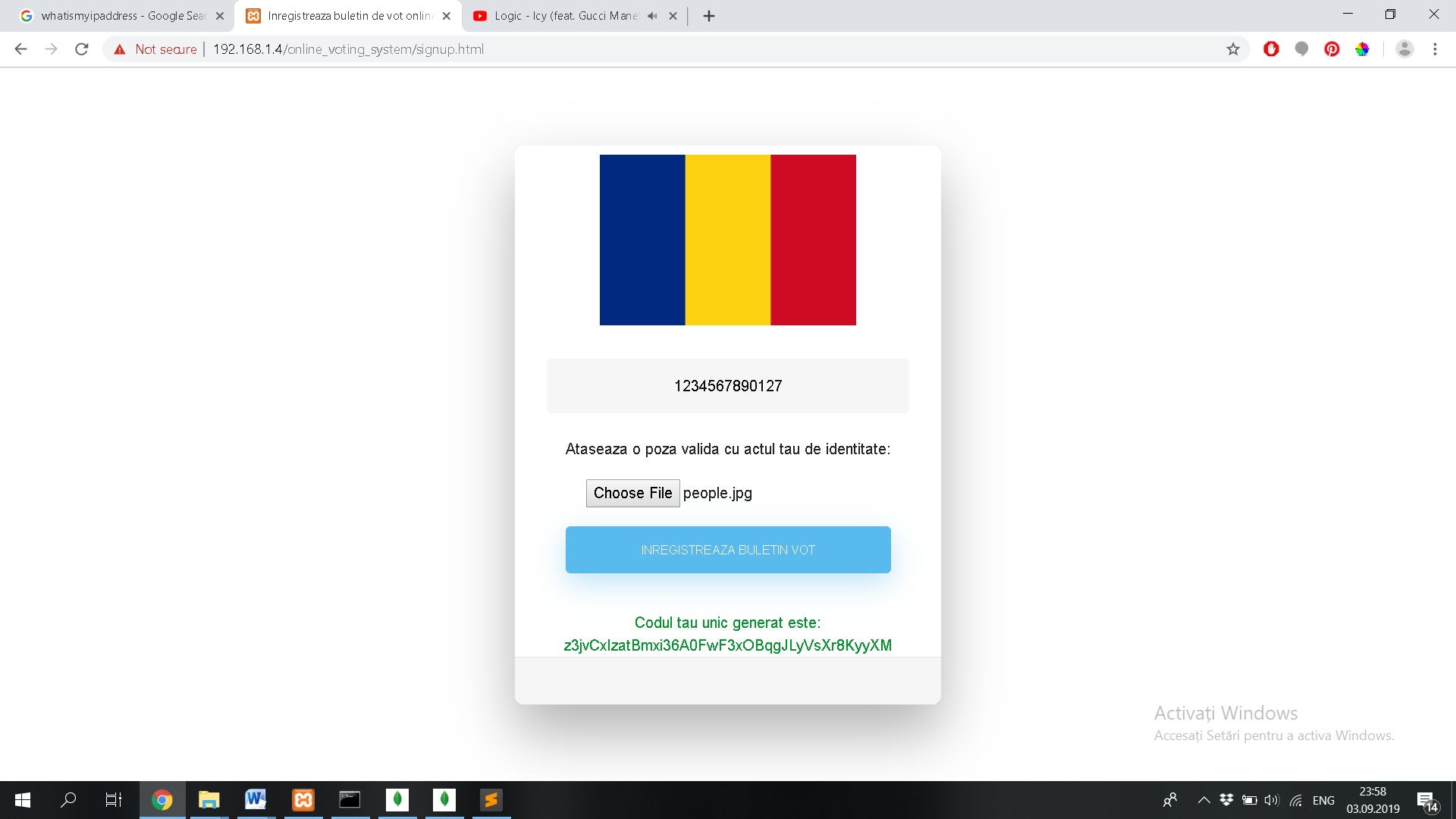
De asemenea ca si metoda de transmitere a datelor intre paginile web am ales, pentru o securitate mai buna , variabilele $\_SESSION din php. De exemplu daca dorim sa incarcam detaliile unei campanii de votare metoda clasica folosind variabile de tip GET ar fi analytics.html?campaign=campanie\_de\_incarcat, dar aceasta metoda poate fi vulnerabila in cazul in care configuram gresit anumite lucruri in PHP si aplicatia poate deveni foarte vulnerabila, de aceea folosim variabilele session deoarece ele sunt server-side si nu sunt vizibile utilizatorului( daca setam acest lucru in setarile serverului ). Deci, in concluzie, pentru a incarca o anumita campanie, vom modifica cu ajutorul AJAX printr-un request catre API variabila $\_SESSION[’ target\_campaign’] cu ’numele\_campaniei\_de\_incarcat’ si ne vom duce la pagina analytics.html care printr-un request automat catre API va citi variabila session si va stii campania care trebuie incarcata.

**signup.html ( pagina de inregistrare )**

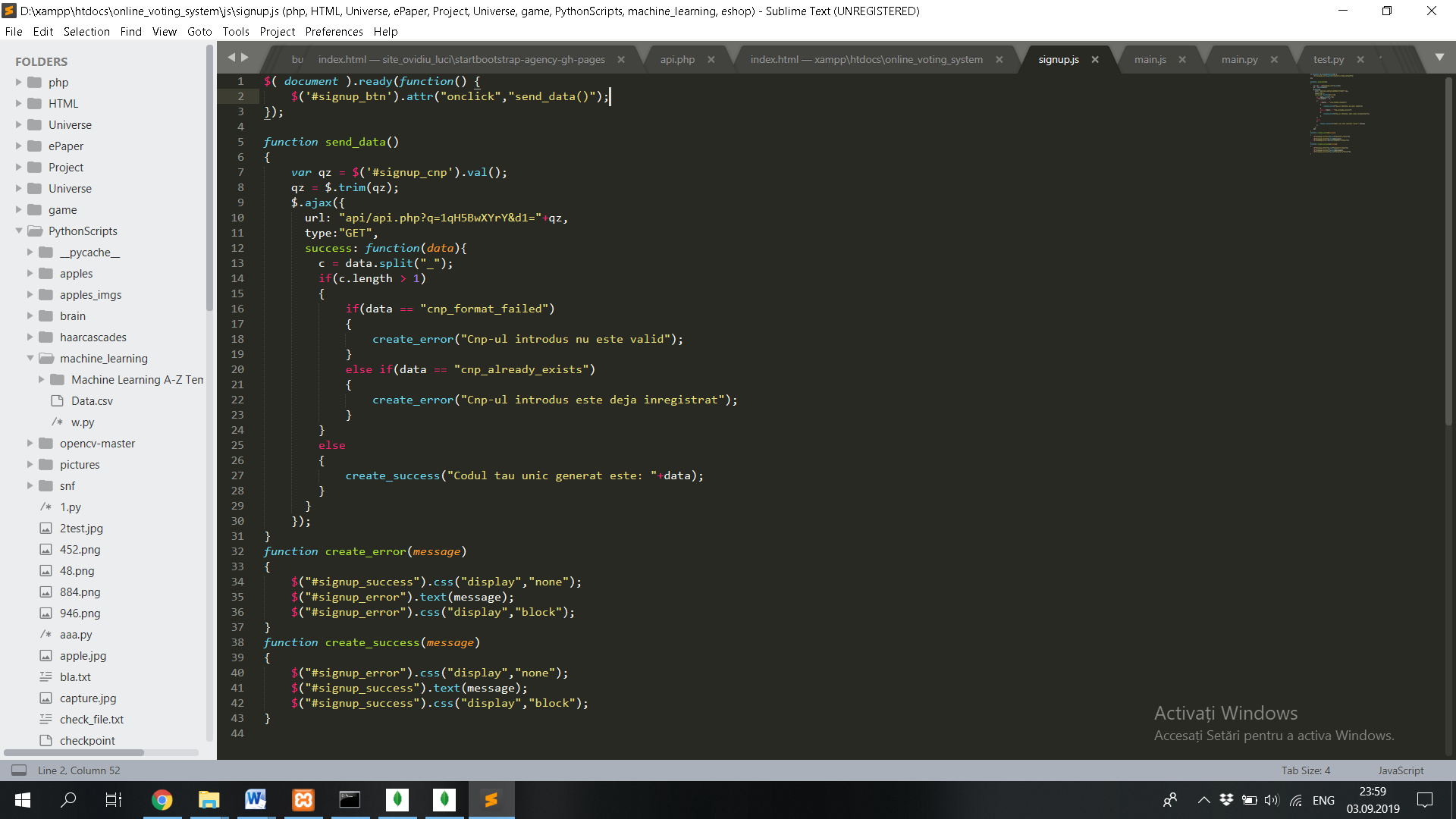
**Descriere**



Sistemul de inregistrare consta in introducerea unui cnp valid ( sistemul executa verificari daca numarul introdus este valid ) si o poza ( teoretic aceste date ar trebui trimise catre o autoritate care ar confirma aceste date dar noi presupunem ca poza corespunde cu cnp-ul ) si apasarea pe butonul de inregistrare. Dupa ce cnp-ul este confirmat utilizatorului ii va fi generat un cod unic cu care acesta se va putea loga pentru a vota.



**Structura codului**



Atunci cand pagina se incarca butonului cu id-ul „signup\_btn” i se adauga atributul „onclick=send\_data()”, adica la apasarea acestuia functia Javascript send\_data() va fi apelata.

**function send\_data()**

**{**

**var qz = $('#signup\_cnp').val();**

**qz = $.trim(qz);**

**$.ajax({**

**url: "api/api.php?q=1qH5BwXYrY&d1="+qz,**

**type:"GET",**

**success: function(data){**

**c = data.split("\_");**

**if(c.length > 1)**

**{**

**if(data == "cnp\_format\_failed")**

**{**

**create\_error("Cnp-ul introdus nu este valid");**

**}**

**else if(data == "cnp\_already\_exists")**

**{**

**create\_error("Cnp-ul introdus este deja inregistrat");**

**}**

**}**

**else**

**{**

**create\_success("Codul tau unic generat este: "+data);**

**}**

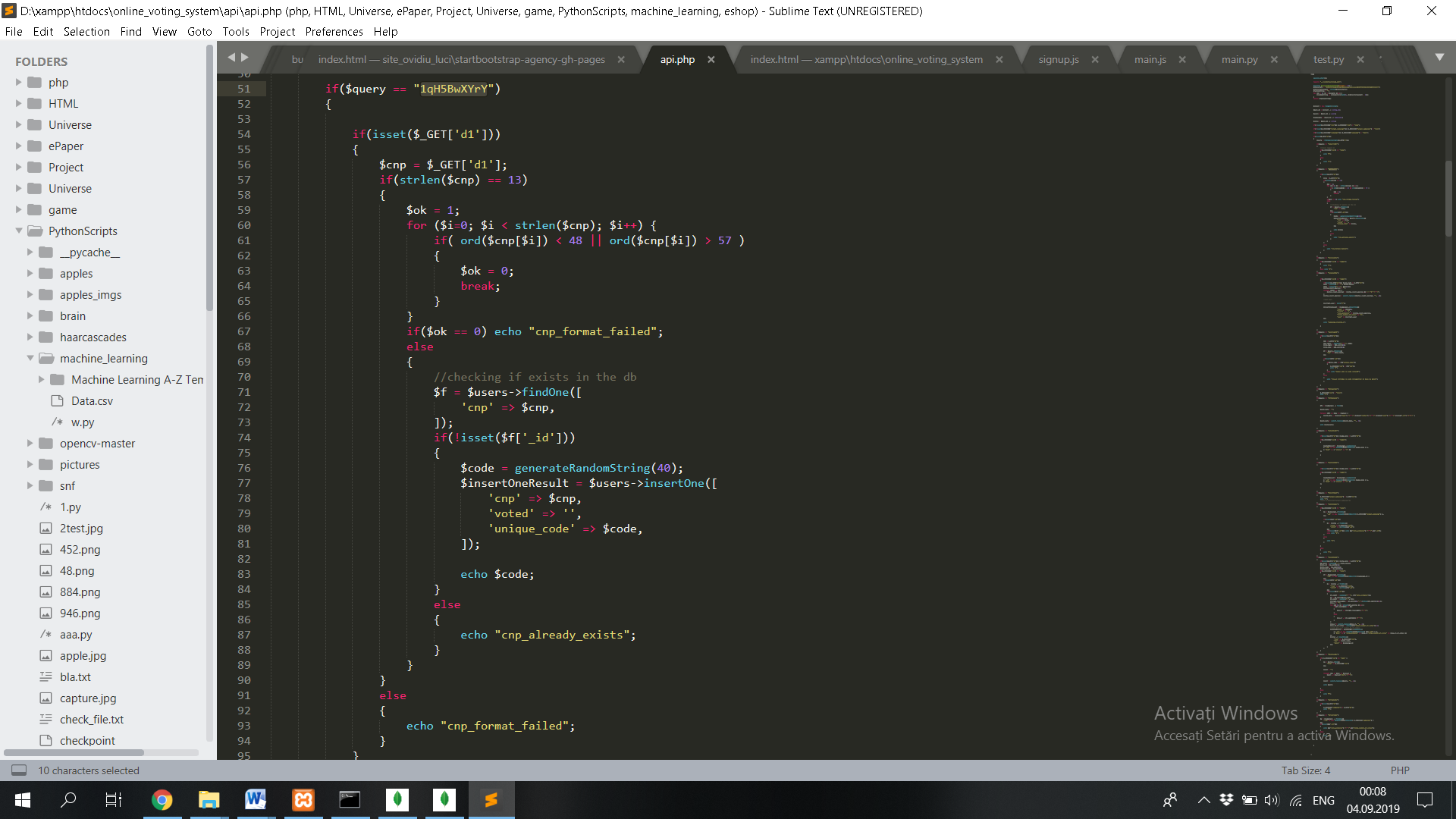
**}**

**});**

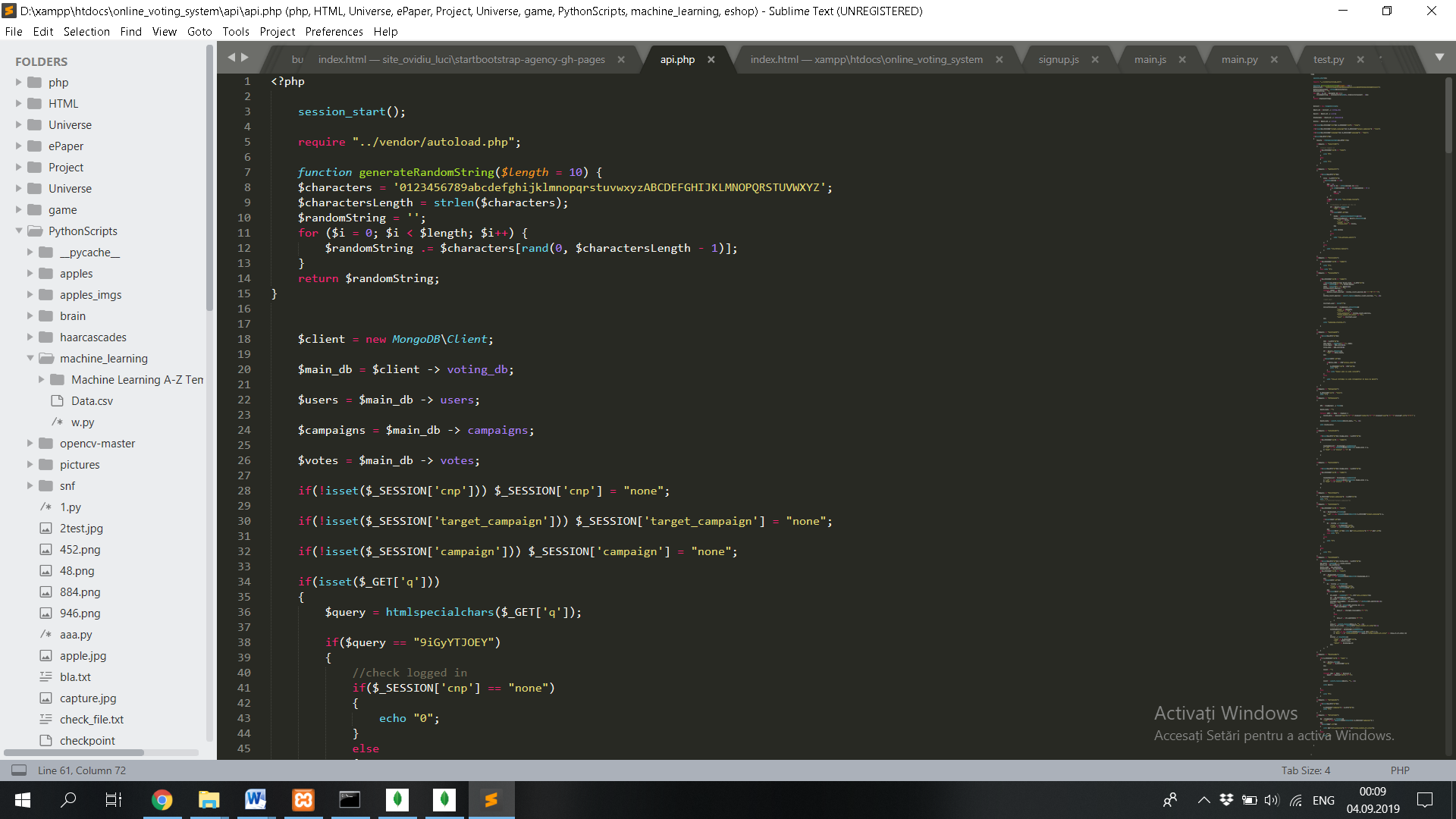
**}**

Functia send\_data() trimite catre API cnp-ul si imaginea, acesta verifica datele si trimite inapoi un raspuns, acesta poate sa fie un cod generat, in acest caz totul a mers in regula si contul a fost inregistrat sau un mesaj care va fi interpretat de functie si se va afisa eroarea in cazul respectiv.

Un snippet cu codul din API:

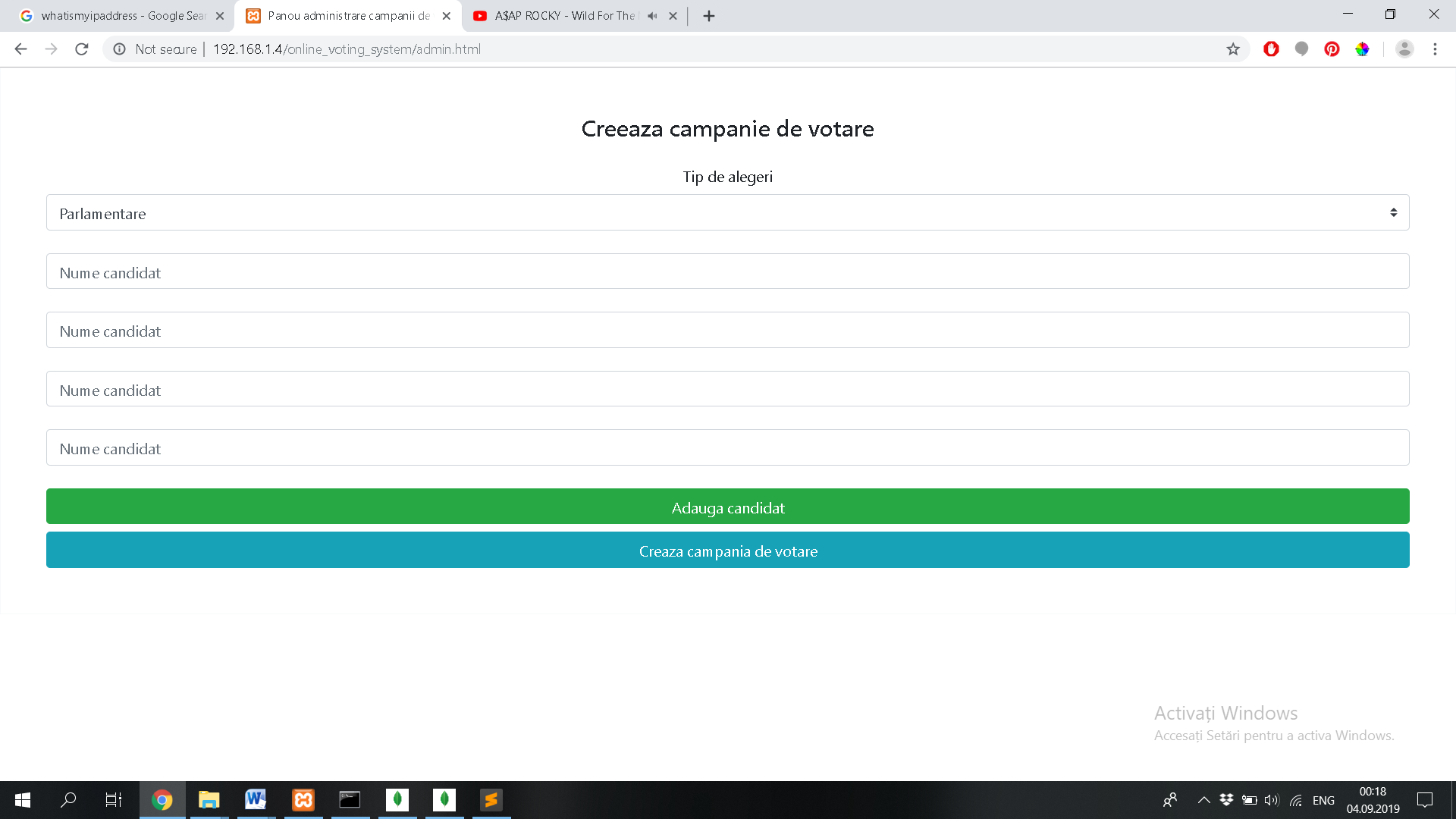


Si functia care genereaza codul pentru buletinul online „generateRandomString(...)” :



**admin.html ( pagina de administrare campanii )**

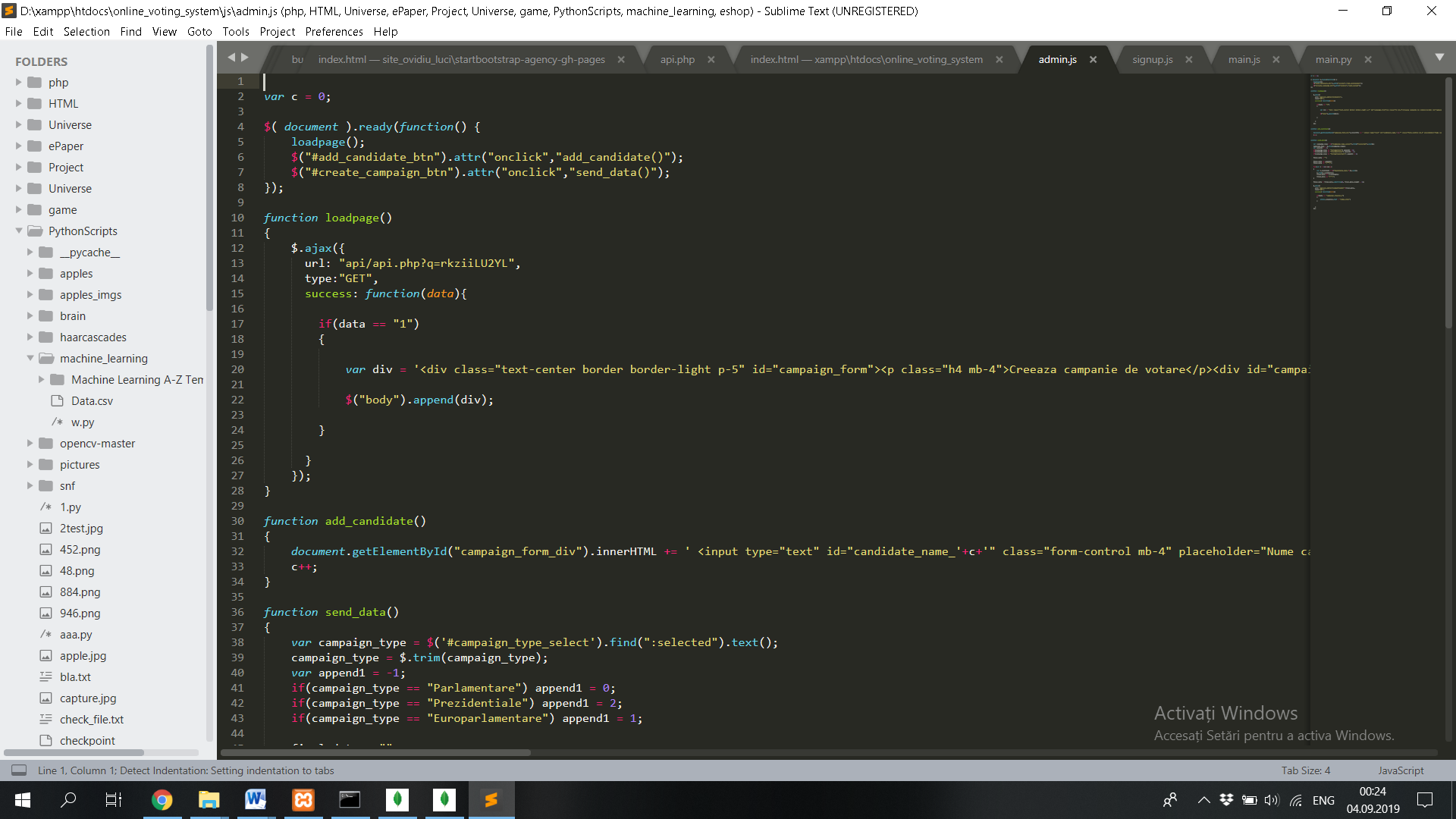
**Descriere**

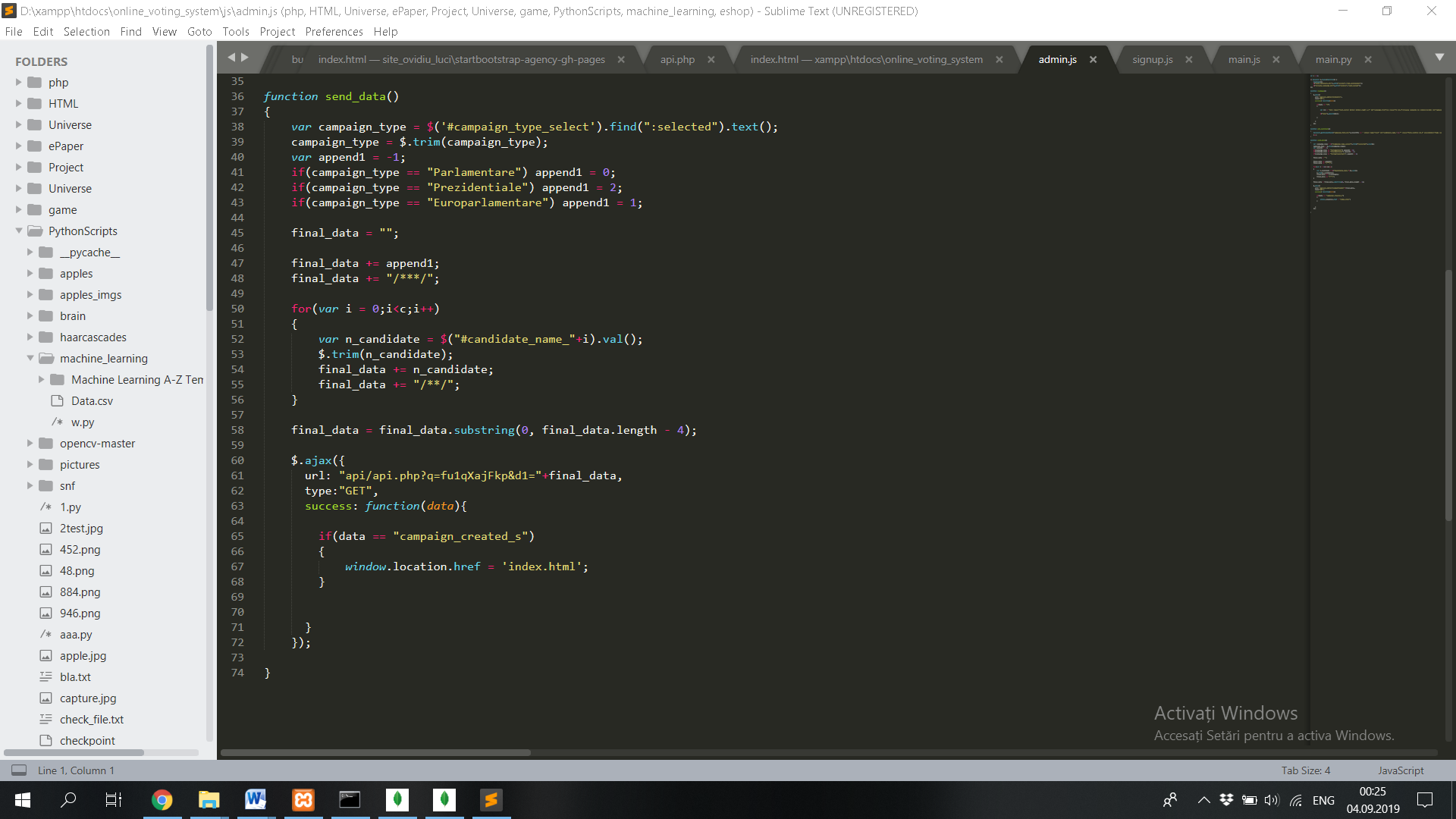


Pagina admin.html este vizibila doar contului de administrator, daca utilizatorul logat nu este administratorul la deschiderea paginii admin.html i se va afisa o pagina alba, goala.

Pe aceasta pagina administratorul poate sa aleaga tipul de campanie pe care vrea sa o initieze si sa adauge candidati. Cand totul este pregatit se apasa pe butonul „Creaza campania de votare”.

**Structura codului**





La incarcarea paginii se apeleaza functia loadpage().

**function loadpage()**

**{**

**$.ajax({**

**url: "api/api.php?q=rkziiLU2YL",**

**type:"GET",**

**success: function(data){**

**if(data == "1")**

**{**

**var div = '<div class="text-center border border-light p-5" id="campaign\_form"><p class="h4 mb-4">Creeaza campanie de votare</p><div id="campaign\_form\_div"><label>Tip de alegeri</label><select class="browser-default custom-select mb-4" id="campaign\_type\_select"><option>Parlamentare</option><option>Europarlamentare</option><option>Prezidentiale</option></select></div><button class="btn btn-success btn-block" id="add\_candidate\_btn" onclick="add\_candidate()">Adauga candidat</button><button class="btn btn-info btn-block" id="create\_campaign\_btn" onclick="send\_data()">Creaza campania de votare</button></div>';**

**$("body").append(div);**

**}**

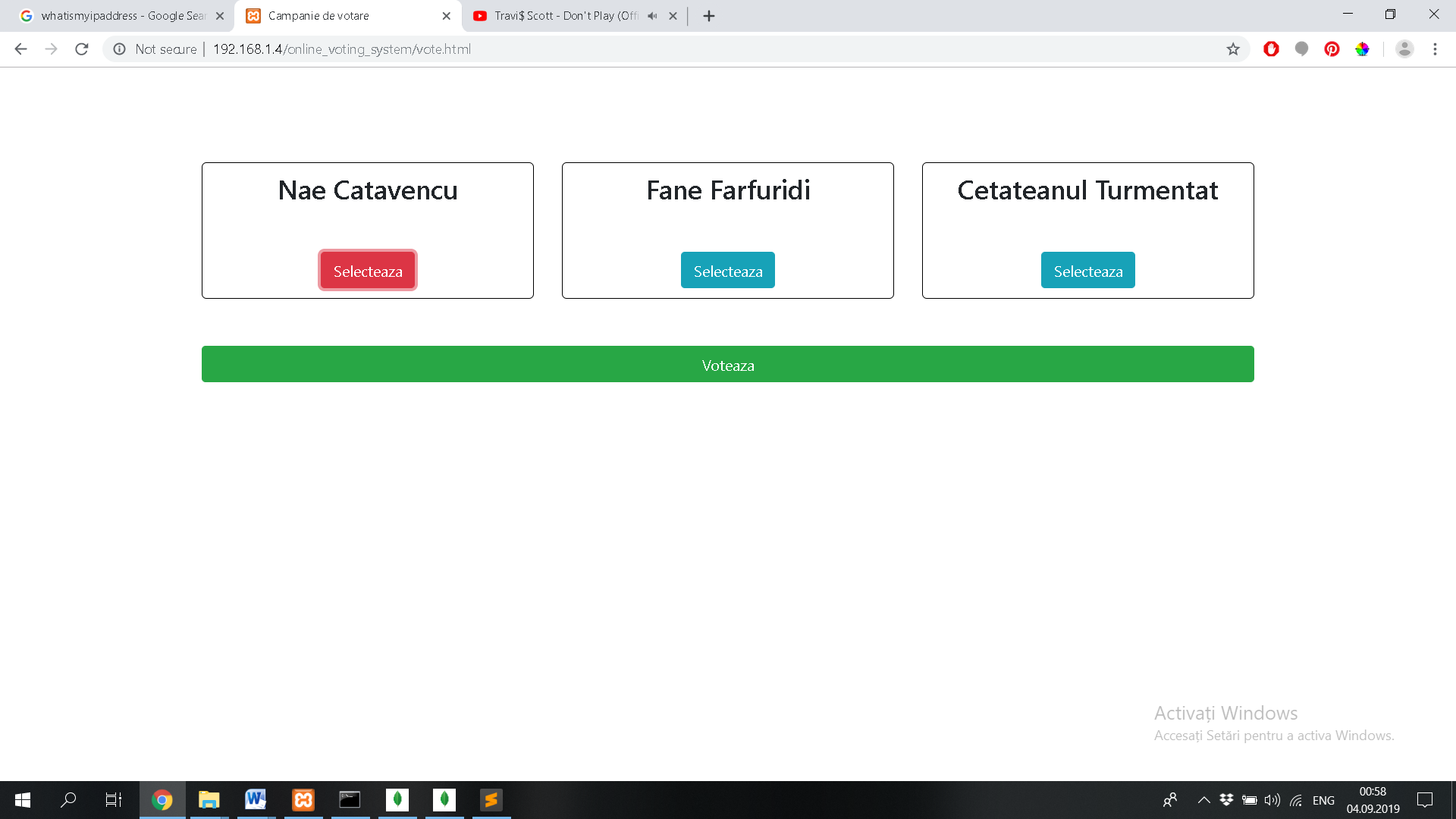
**}**

**});**

**}**

Aceasta functie trimite un GET request catre API pentru a verifica daca utilizatorul logat este administratorul, in cazul in care primeste raspunsul „1” inseamna ca acesta este admin-ul si incarca butoanele pentru adaugarea campaniilor, in cazul contrar nu incarca nimic.

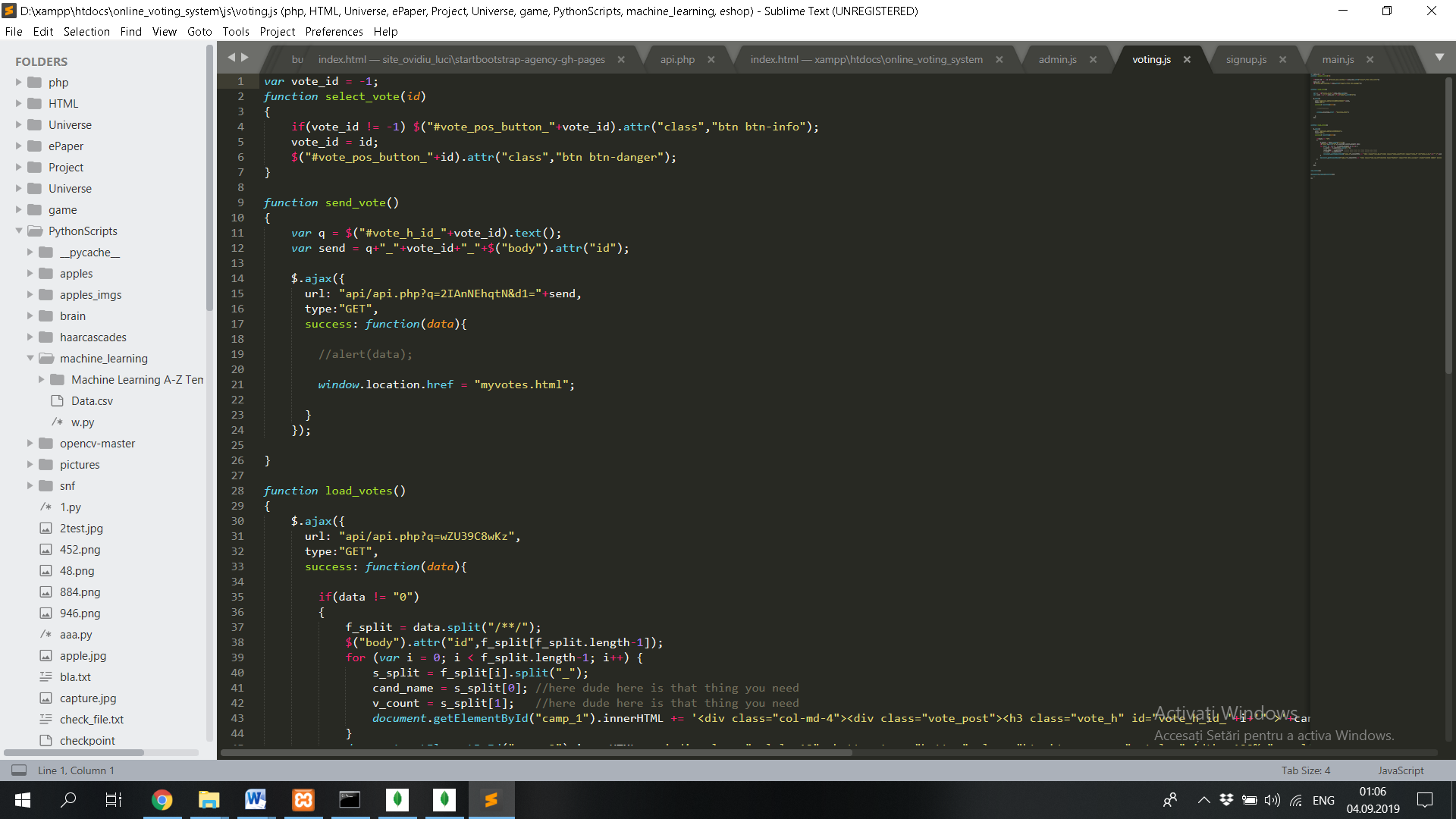
**vote.html ( pagina de votare )**

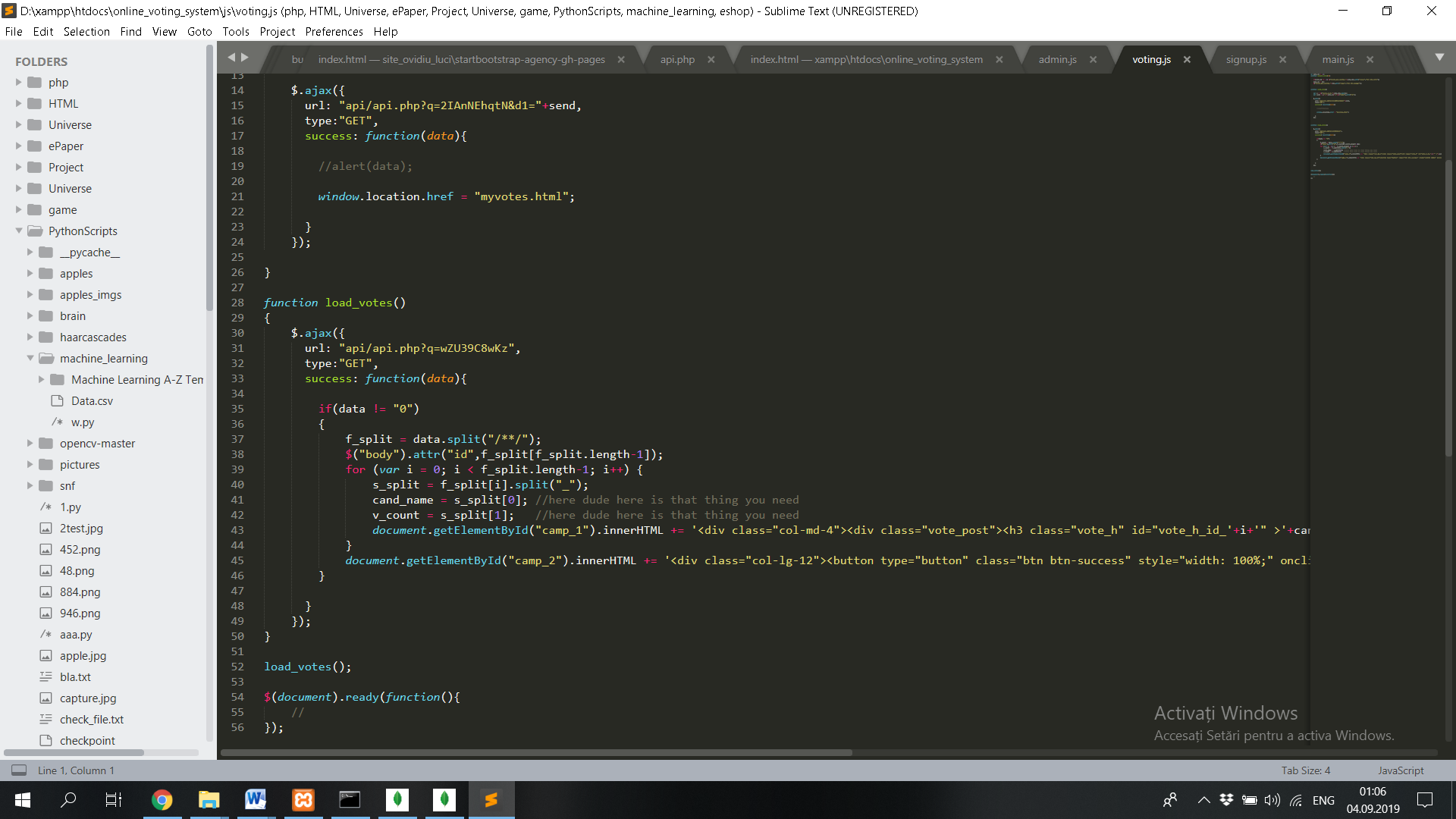


Aceasta pagina este pagina de votare, imaginea de sus este un exemplu de campanie de votare in desfasurare. Selectam candidatul si apasam voteaza, acest template seamana cu buletinul din realitate, numai ca este transpus intr-un format digital.

Desigur daca utilizatorul nu este logat cu un cnp inregistrat nu se va afisa nimic.

**Structura codului**





Atunci cand pagina se incarca este apelata functia „load\_votes()” care incarca toate detaliile legate de campanie si verifica daca utilizatorul este logat cu un cnp inregistrat.

**function load\_votes()**

**{**

**$.ajax({**

**url: "api/api.php?q=wZU39C8wKz",**

**type:"GET",**

**success: function(data){**

**if(data != "0")**

**{**

**f\_split = data.split("/\*\*/");**

**$("body").attr("id",f\_split[f\_split.length-1]);**

**for (var i = 0; i < f\_split.length-1; i++) {**

**s\_split = f\_split[i].split("\_");**

**cand\_name = s\_split[0]; //here dude here is that thing you need**

**v\_count = s\_split[1]; //here dude here is that thing you need**

**document.getElementById("camp\_1").innerHTML += '<div class="col-md-4"><div class="vote\_post"><h3 class="vote\_h" id="vote\_h\_id\_'+i+'" >'+cand\_name+'</h3><button type="button" class="btn btn-info" id="vote\_pos\_button\_'+i+'" onclick="select\_vote('+i+');">Selecteaza</button></div></div>';**

**}**

**document.getElementById("camp\_2").innerHTML += '<div class="col-lg-12"><button type="button" class="btn btn-success" style="width: 100%;" onclick="send\_vote()">Voteaza</button></div>';**

**}**

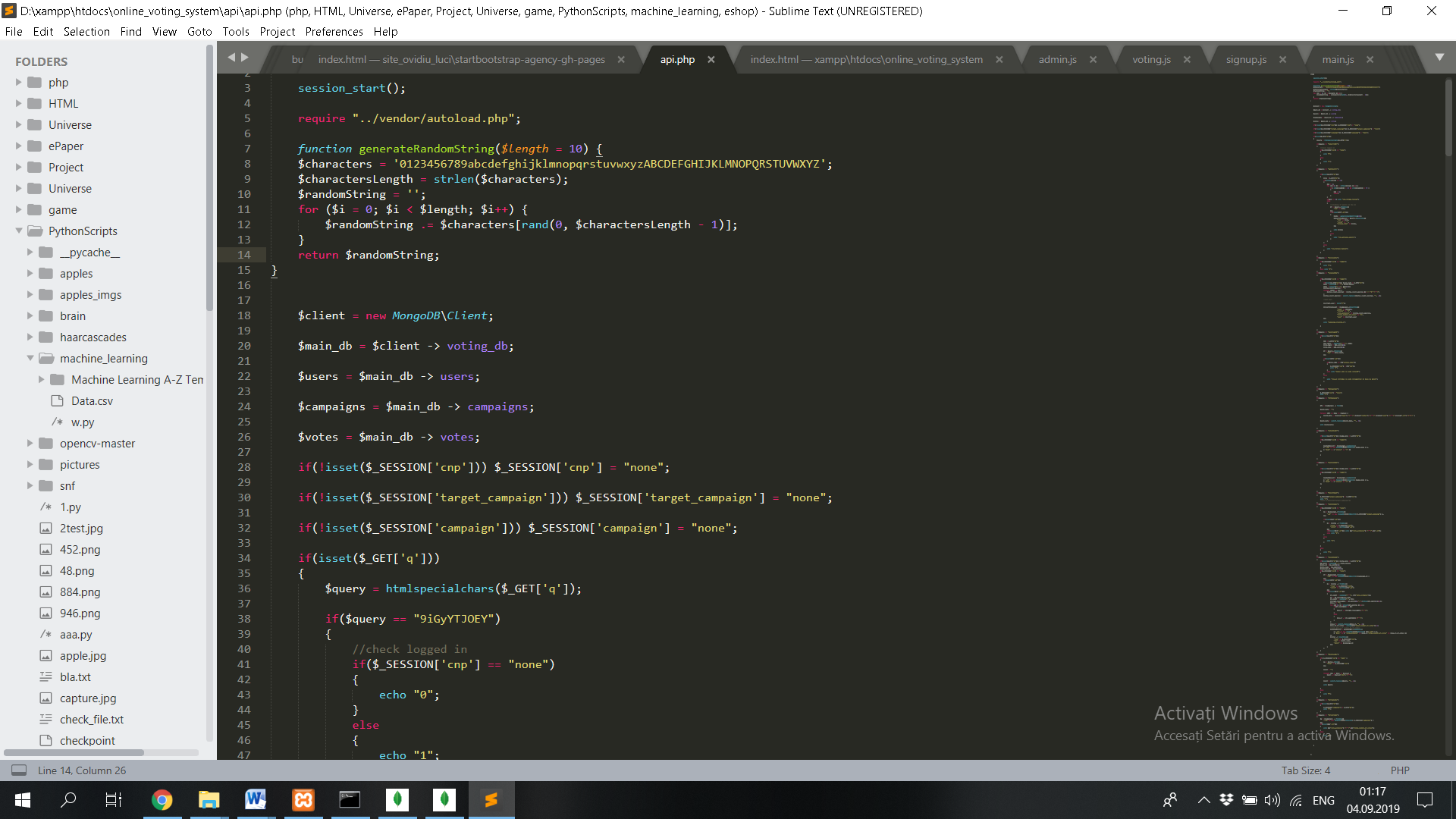
**}**

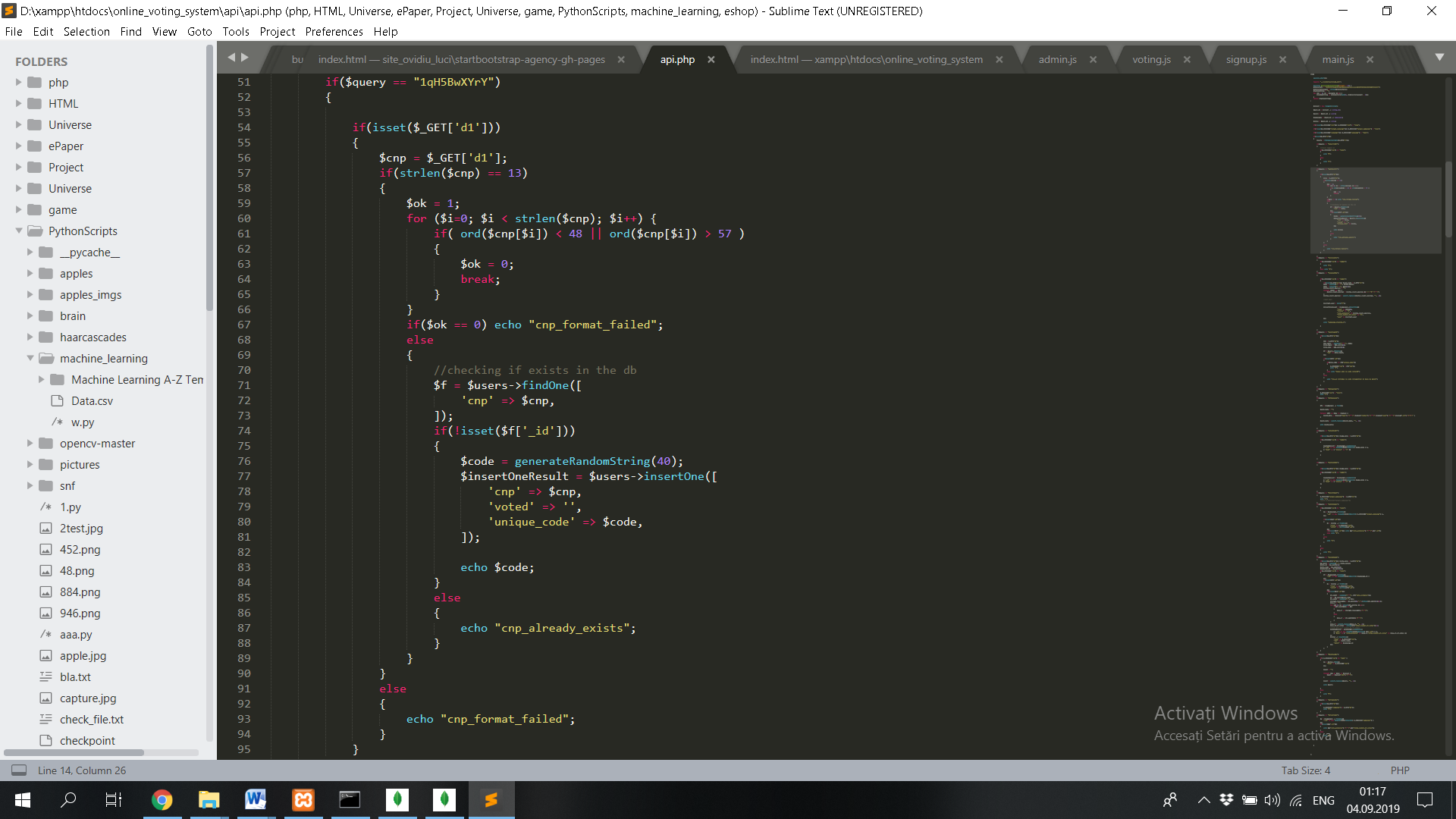
**});**

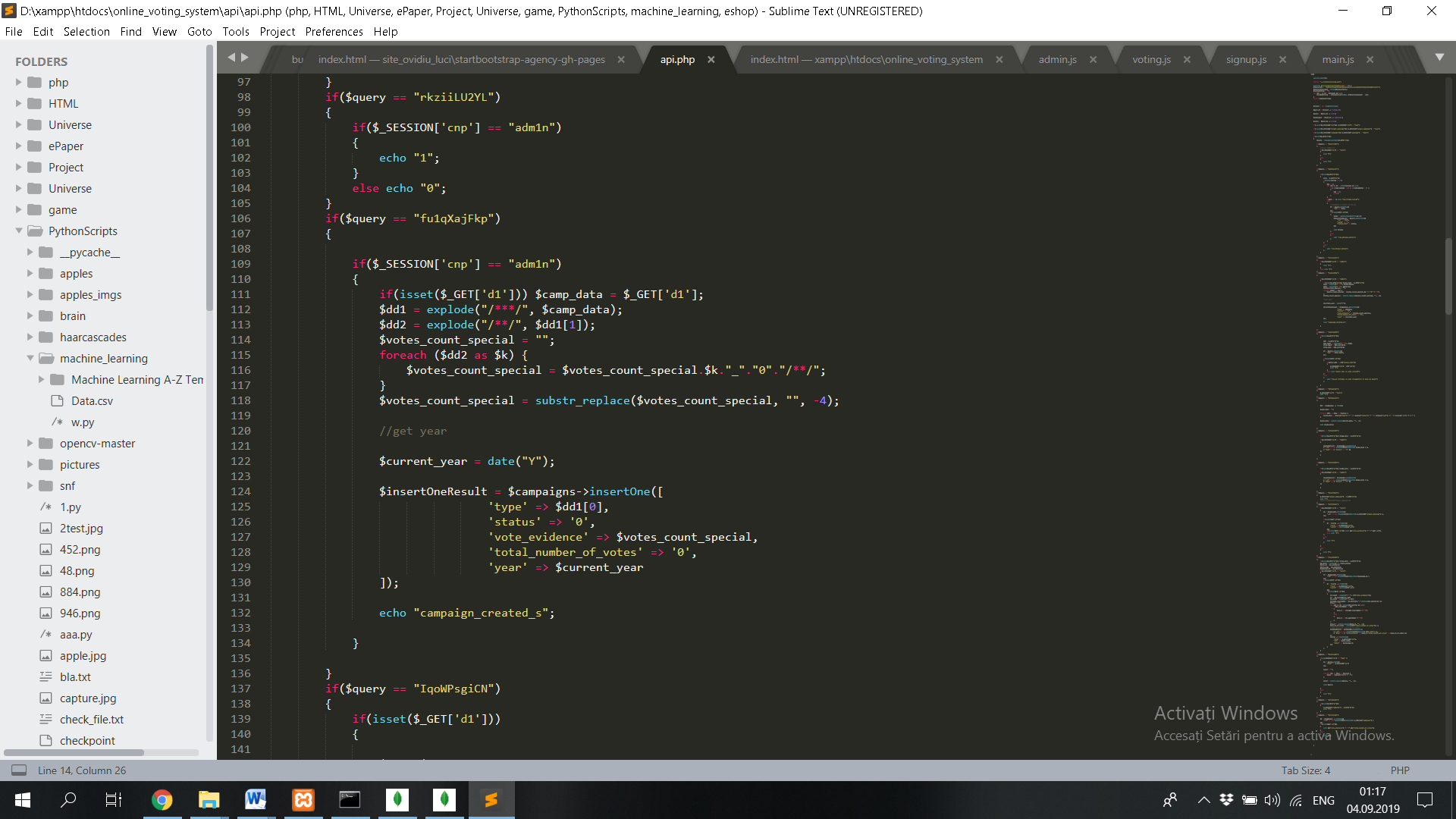
**}**

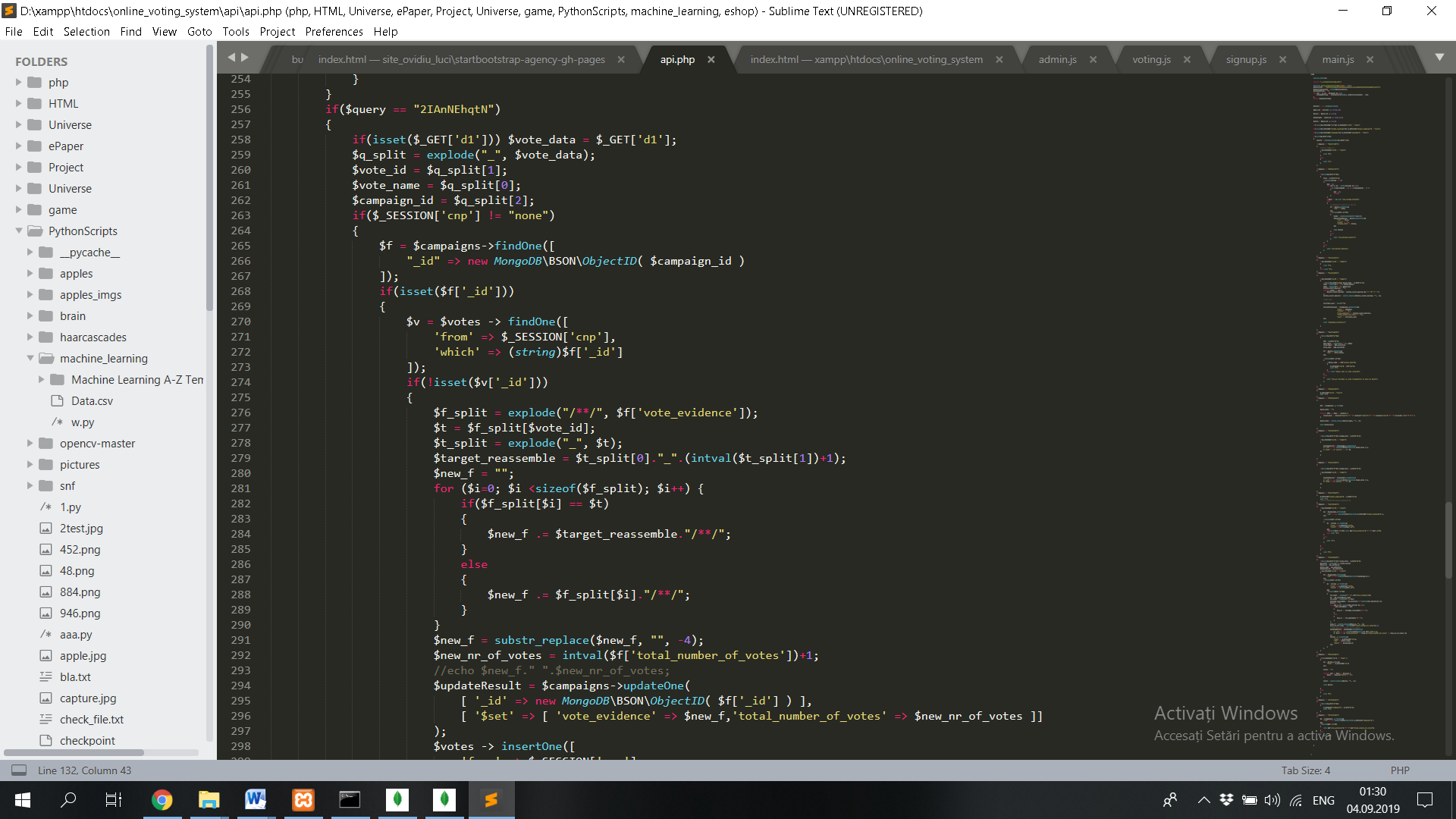
**Structura API-ului**

Pentru aceasta aplicatie am ales un API RESTful, comunicarea cu API-ul se face prin intermediul requesturilor AJAX de tipul GET, fisierul principal al API-ului este api.php . Template-ul requestului este ../api.php?q=’cod query encriptat’&d1=’data1’&d2=’data2’... etc.









**Directii de dezvoltare**

Cu siguranta ca acest sistem poate fi imbunatatit, el reprezinta doar un prototip functional pentru un eventual sistem de vot online. In primul rand se pot adauga niste optiuni mai avansate la sistemul de creare de campanii de vot, cum ar fi o poza la fiecare candidat, o scurta descriere etc.

De asemenea, pentru a spori securitatea se pot encrypta toate mesajele trimise intre API si aplicatie.

In viitor acest sistem ar putea fi rafinat si imbunatatit pentru a putea fi folosit in realitate , cu toate ca votul online reprezinta o dificultate semnificativa nu din punct de vedere tehnic ci mai mult din cauza tentativelor de frauda, prezente in numar ridicat in tara noastra.

**Bibliografie**

<https://www.wikipedia.org/>

<https://www.w3schools.com/>

<https://stackoverflow.com/>

<https://www.mongodb.com/>

<https://www.php.net/>

<https://www.google.com/>

[https://superuser.com](https://superuser.com/)