Proiect "Nikola"

- concurs: InfoEducatie, editia 2017
- categorie: Soft educational
- realizator: Iosif George Andrei

I. Introducere

Nikola este un proiect educational de tip "website", vizand categoria concursului "Soft educational". A fost dezvoltat cu scopul de a ajuta elevii si profesorii ce predau informatica, plecand de la urmatoarele cerinte:

- elevii sa se poata loga fara a tine minte alt cont
- elevii sa aiba access la o platforma portabila
- elevii sa creeze programe in diverse limbaje de programare
- elevii sa rezolve probleme, avand propuse metode eficiente de rezolvare
- profesorii sa verifice progresul elevilor in rezolvarea problemelor
- profesorii sa gestioneze rapid datele aflate pe website
- profesorii sa modifice problemele existente sau sa adauge probleme noi

Astfel, s-a dezvoltat aplicatia Nikola, care ajuta la imbunatatirea invatamantului romanesc si al invatarii individuale.

Numele platformei, Nikola, este un omagiu adus inventatorului si fizicianului sarb-american Nikola Tesla, care a contribuit definitiv la dezvoltarea tehnologiei.

II. Utilizabilitate, originalitate si inovatie

Platforma Nikola poate fi accesata ca website.

Odata accesata aplicatia, elevul este intampinat cu o pagina de start, in care este invitat sa se logheze cu contul sau de Github, platforma foarte populara in randul tinerii pasionati de IT. Dupa confirmarea logarii, el este redirectionat catre dashboard.

Dashboard-ul reprezinta aplicatia propriu-zisa. In partea stanga a aplicatiei se afla un meniu cu urmatoarele butoane:

- "Compileaza", cu un buton atasat, care, odata actionat, va compila programul introdus de user, folosindu-se de datele de intrare, si va returna datele de iesire si informatii despre compilarea programului
- "Problema actuala", care, actionat, va actualiza recomandarile profesorilor, aflate in dialog box-ul din coltul sus-dreapta, si va afisa in fereasta de mai jos cerinta problemei si format-ul datelor de intrare si de iesire; langa acesta apare un buton care va trimite datele de iesire returnare de program si le va compara cu cele corecte, introduse de profesori
- "Informatii", care, odata actionat, va afisa datele preluate de pe contul personal de Github al utilizatorului; langa acesta apare un buton folosite pentru delogare

Compilarea se poate realiza in mai multe limbaje de programare, a caror selectare poate fi motivata de gama larga de utilizarea a acestora:

- C: aplicatii desktop, sistem
- C#: aplicatii desktop, client-side, server-side, web
- C++: aplicatii desktop, sistem, educatie
- Java: aplicatii desktop, client-side, server-side, web, aplicatii mobile
- Javascript: client-side, server-side, web
- Pascal: aplicatii desktop, educatie
- Perl: aplicatii desktop, scripting, web
- PHP: server-side, web
- Python: apicatii desktop, web, scripting, artificial intelligence
- Swift: aplicatii mobile

In continuarea acestor butoane, se afla o fereasta folosita pe post de frame, in care se vor afisa diverse informatii. Un form de selectare permite utilizatorilor sa selecteze limbajul de programare. Widget-urile din partea de jos a meniului afiseaza:

- ora
- vremea de afara
- timpul petrecut pe platforma
- recomandarea platformei, prin care elevul este motivat sa fie productiv sau anuntat sa ia o pauza, in cazul in care a trecut o ora de cand se afla pe website

In coltul sus-dreapta al paginii este prezent un dialog box, in care este afisata recomandarea profesorului cu privire la rezolvarea problemei. Pe langa acestea, pagina contine si 4 editoare de text:

- editorul, in care se va introduce programul; acesta prezinta o evidentiere a sintaxei programului, specifica limbajului selectat, cat si asa-numitii "workeri", care afiseaza in timp real erorile, prezenti insa pentru un numar redus de limbaje de programare
- informatii de compilare, in care sunt afisare detalii despre compilarea programului, precum: status-ul, result-ul, timpul de compilare, spatiul ocupat si erorile
- datele de intrare
- datele de iesire, respectiv rezultatul trimiterii problemei

Odata ce elevul trimite solutia unei probleme, el va primi un rezultat al acesteia. Daca solutia este valida, este anuntat si ii va fi recomandata alta problema. Daca nu este valida, va fi indemnat sa mai incerce. In cazul in care solutia este valida, dar s-a epuizat baza de date a problemelor, elevul va fi anuntat ca solutia este valida si ii este recomandat sa incerce mai tarziu, dupa ce vor fi adaugate noi probleme.

Elevul are posibilitatea de a salva programul realizat de el prin scrierea codului, selectarea limbajului de programare si prin folosirea shortcut-ului "CTRL+S".

Pagina de admin este disponibila doar profesorilor, care pot edita in timp real informatiile de pe paginile platformei, pot verifica progresul elevilor cu ajutorul adresei de email, cat si edita sau adauga probleme. In partea de sus a paginii sunt prezente butoane cu care se pot accesa mai usor paginile platformei. Sub acestea, apar input-uri de tip text si butoane pentru actualizare. Pentru a parcurge lista de probleme, se vor folosi sagetile. In momentul in care se va depasi numarul de probleme aflate in baza de date, se va afisa un form pentru adaugarea unei noi probleme.

Din punct de vedere al inovatiei si al originalitatii, platforma este unica prin posibilitatea compilarii de pe orice dispozitiv avand instalat un browser. Website-ul satisface un numar relativ mare de nevoi ale elevilor si ale profesorilor. Logarea cu contul de Github nu impovareaza elevul cu alt cont specific aplicatiei Nikola, ci, mai degraba, favorizeaza folosirea website-ului Github, necesar unui viitor dezvoltator in domeniu.

Publicul-tinta este reprezentat de catre elevii care doresc sa invete programarea intr-un program si sa renunte la metodele clasice folosite in scolile romanesti.

III. Interfata si continut

Nikola poseda o interfata intuitiva, usor de utilizat si adaptiva pe majoritatea browserelor si sistemelor de operare. Aspectul este placut, prin utilizatrea unei compinatii de albastru, alb si negru. Au fost folosite icon-uri specifice pentru butoane si efecte de hover, care confera o nuanta de modernitate interfetei.

Continutul platformei este corect din punct de vedere gramatical si stiintific. El poate fi modificat in timp real de catre profesori cu ajutorul paginii de admin.

IV. Arhitectura si programare

Pentru a realiza proiectul s-au utilizat tehnologiile: HTML, CSS, Javascript, Node.js si MongoDB. Structura folderelor este urmatoarea:

- node_modules: modulele NPM
- others/Database: o copie a bazei de date NoSQL
- others/Documentatie: documentatia de fata
- others/Graphic/Favicon: proiectul grafic al faviconului
- others/Graphic/Github: proiectul grafic al logo-ului pentru Github
- others/ Graphic/Poster: proiectul grafic al posterului
- others/Screenshots: screenshot-uri ale aplicatiei
- server/config: datele de configurare ale server-ului Node.js
- server/controllers: controalele rutelor MVC
- server/models: modelele rutelor MVC
- server/my_modules: modulele create pentru a usura dezvoltarea
- server/public/css: fisierele .css generate cu ajutorul Gulp.js
- server/public/img: imaginile utilizate pe platforma
- server/public/js: script-urile Javascript pe partea de frontend

- server/public/less: fisierele .less folosite pentru a genera fisierele .css
- server/public/lib: librariile .css sau .js utilizate pe platforma
- server/public/views: fisierele .pug folosite pentru a genera fisierele .html, views pentru rutele MVC
- server/sockets: configurarea WebSockets-urilor
- .git: contine fisierele specifice repository-ului de Github

Logica server-ului se afla in fisierul app.js. In interiorul acestuia, sunt luate configuratiile din fisierul JSON config/config.json, de la port la contul de admin. Sunt preluate modulele NPM in prima parte a fisierului. Cu ajutorul mongo, se creeaza o legatura cu baza de date MongoDB si modelele necesare preluarii datelor. In dezvoltarea server-ului, s-a folosit Express, pe care au fost setate diferite middlewares precum cors si hemlet. Pug este folosit ca template engine pentru a usura scrierea HTML. Pentru a stabili o conexiune constanta cu user-ul, am folosit socket.io.

Au fost folosite trei API-uri:

- Cloud Compiler API: compilarea programelor utilizatorului
- IP-API: preluarea locatiei in functie de IP-ul utilizatorului
- Open Weather Map: preluarea datelor meteorologice in functie de locatie

La un request, user-ul primeste un fisier .html, care reprezinta de fapt un fisier .pug, in care au fost introduse datele preluate cu ajutorul mongo. Fisierele .css sunt compilate din fisierele .less.

S-au utilizat urmatoarele librarii CSS si Javascript:

- Font Awesome: icons
- Ace.js: framework pentru realizarea editoarelor de text
- Mousetrap.js: crearea shortcut-ului CTRL+S
- Socket.io: WebSockets pe partea de client
- Vue.js: framework front-end pentru Javascript

S-au folosit urmatoarele module instalate cu NPM:

- body-parser: modul folosit pentru a parsa body-urile request-urilor inainte de handlere
- browser-sync: modul folosit in Gulp.js pentru live reload
- cookie-parser: modul folosit pentru a parsa cookie-urile
- cors: modul folosit pentru a activa CORS
- express: modul folosit ca framework pe server-ul Node.js
- express-partials: modul folosit in Express pentru partials
- express-session: modul folosit pentru a seta sessions pe server
- gulp: modul de gestionare a task-urilor
- gulp-less: modul folosit pentru a crea un task pentru a compila fisierele .less
- helmet: modul folosit pentru a securiza aplicatia Node.js
- less: modul folosit pentru a compila fisierele .less
- method-override: modul pentru a activa verbele HTTP, precum PUT si DELETE, unde clientul nu suporta
- mongo: modul pentru pentru MongoDB
- passport: modul pentru a autentificarea in Node.js
- passport-github2: modul pentru autentificarea cu Github
- pug: modul folosit ca template engine, compliland fisierele .pug in .html
- request: modul folosit pentru a faca request-uri in Node.js
- socket.io: modul pentru a utiliza WebSockets pe partea de server

In fisierul gulpfile.js sunt precizate task-urile Gulp.js. Acest modul a fost esential in dezvoltarea aplicatiei deoarece am automatizat munca prin live reload si compilarea fisierelor less. Tot pentru dezvoltare, am folosit editorul Atom, browserele Crome si Firefox, GUI-ul pentru mongo Robo3T, iar pentru versionare Github, asigurandu-se caracterul de open-source al platformei. Pentru design-ul grafic, am folosit Adobe Illustrator.

Codul sursa al aplicatiei este foarte bine structurat si este documentat cu ajutorul comentariilor, cu scopul de a putea fi inteles de posibili colaboratori. Design-ul este responsive, insa pe dispozitivele mobile este recomandata utilizarea landscape. Testarea a fost realizata pe browserele Chrome, Firefox si pe sistemele de operare Windows si Ubuntu, proiectul neprezentand erori pe partea de cliend-side si server-side.

Securitatea aplicatiei este asigurata prin modulul helmet, prin protejarea aplicatie de majoritatea atacuri specifice Node.js si prin verificarea datelor utilizatorului. Socket-urile au fost protejate prin crearea unui middleware care verifica daca realizatorul request-ului este autorizat.

V. Dezvoltare ulterioara

In viitor, aplicatia va fi prezentata in mod oficial unor profesori si elevi, pentru a primi un feedback dupa care echipa sa isi dea seama daca aplicatia starneste interes si daca ar trebui sa continue dezvoltarea, caz in care se pot aduce urmatoarele imbunatatiri:

- posibilitatea elevilor de a-si putea customiza lista de limbaje de programare
- adaugarea a mai multe metode de login cu ajutorul retelelor de socialziare
- posibilitatea profesorilor de a crea sesiuni de live-coding, la care elevii sa participe online

VI. Ghid de instalare si de configurare

Platforma Nikola poate fi testata local cu ajutorul Node.js si a MongoDB, urmand pasii:

- instalare Node.js
- instalare si rulare MongoDB
- instalare Robo3T
- importare in Robo3T a bazei de date NoSQL, aflata in folder-ul /database
- instalare dependente proiect cu ajutorul NPM
- configurarea variabilelor specifice cu ajutorul fisierului config.json aflat in folder-ul /config
- rulare app.js cu ajutorul Node.js sau nodemon

Instalarea Robo3T este optionala in cazul in care se va folosi sheel-ul mongo, instalat odata cu MongoDB. In cazul in care intampinati probleme cu instalarea modulelor de pe NPM, va invit sa clonati repository-ul privat de pe Github: https://github.com/iosifache/Nikola.git.

VII. Opinie

Justificarea tehnologiilor alese: am ales Node.js ca runtime environment deoarece acesta prezinta numeroase avantaje. Fiind un proiect relativ tanar fata de celelate alternative folosite pentru a crea un server, Node.js este bazat de Javascript, folosindu-se astfel acelasi limbaj de programare pentru frontend si pentru backend. Folosind engine-ul V8 al celor de la Google si asynchronous I/O, viteza cu care ruleaza este foarte mare. Dispune de un manager de module, numit NPM, si este recomandat sa se foloseasca cu acest tip de aplicatii baze de date NoSQL, implementate si in aplicatia de fata cu ajutorul MongoDB.

Opinia despre ideea de baza a proiectului si despre utilitate: suntem toti consitenti ca invatamantul din Romania nu este cel mai eficient, in comparatie cu sistemele din strainatate, si de faptul ca ca o aplicatie web nu va revolutiona tot sistemul de invatamant. Am incercat sa imbunatatim predarea si invatarea unei materii: informatica. Aceasta materie reprezinta, din punctul meu de vedere, temelia generatiilor urmatoare care vor trai intr-un viitor mult mai tehnologizat. Prin utilitatea aplicatiei, ce se doresc a fi folosita in scolile romanesti, se starneste interesul elevilor pentru Informatica, cu scopul final de a avea in viitor, poate in urmatorii 5-10 ani, tineri mult mai pregatiti in domeniul IT.