Universitatea Tehnică „Gheorghe Asachi”, Iaşi
Facultatea de Automatică şi Calculatoare

Programarea Clientului Web

Proiect “Travel guide”

Îndrumător: Tiberius Dumitriu

Student : Dragoș Geantău
Grupa: 1406A

Student : Bogdan Cărpușor
Grupa: 1406B

Cuprins

Capitolul 1. Introducere 3
Capitolul 2. Fundamentare teoretică 4
Capitolul 3. Proiectarea aplicației 5
Capitolul 4. Implementarea aplicației 6
Capitolul 5. Concluzii 7
Capitolul 6. Bibliografie 8

Capitolul 1. Introducere

Aplicația își propune preluarea de pe Facebook a locațiilor vizitate de utilizator și afișarea acestora pe o harta Google intr-un formal simplu și interactiv. Pe baza datelor preluate aplicația poate genera diverse rapoarte precum orașele cele mai vizitate sau perioada preferată de călătorie a utilizatorului.

Pentru partea de front-end aplicația folosește următoarele tehnologii:

- HTML5
- CSS
- Bootstrap
- Javascript
- jQuery
- AngularJS
- NodeJS
- Grunt
- Bower
- Google maps API
- Facebook API

Pentru partea de back-end aplicația folosește următoarele tehnologii:

- Oracle DB
- Rails -v 4.2.5
- Ruby -v 2.1.6
- Bundle

Capitolul 2. Fundmentare teoretică

HTML5 este un limbaj de markup folosit pentru structurarea și prezentarea conținutului de pe World Wide Web, finalizat și publicat în Octombrie 2014 de către Consorțiu World Wide Web.

CSS (Cascading Style Sheets) este un limbaj (style language) care definește "layout-ul" pentru documentele HTML. CSS acoperă culori, font-uri, margini (borders), linii, înalțime, lățime, imagini de fundal, poziții avansate și multe alte opțiuni.

Bootstrap este un framework HTML, CSS și Javascript dezvoltat de Twitter ce permite realizarea de site-uri web responsive, care se adapteaza la orice rezoluție de dispozitiv: desktop, tablete și telefoane mobile.

Javascript este un limbaj de programare orientat obiect bazat pe conceptul prototipurilor. Este folosit mai ales pentru introducerea unor funcționalități în paginile web, codul Javascript din aceste pagini fiind rulat de către browser.

jQuery este o platformă de dezvoltare JavaScript, concepută pentru a ușura și îmbunătăți procese precum traversarea arborelui DOM în HTML, 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.

AngularJS este un framework open-source folosit în principal pentru dezvoltarea aplicațiilor web single-page.

Node.js este o platformă software folosită pentru a construi aplicaţii de reţea scalabile (mai ales server-side). Node.js utilizează JavaScript ca limbaj de scripting și conține intern o bibliotecă server HTTP, ceea ce face posibilă rularea unui server de web fără utilizarea de software extern, cum ar fi Apache sau Lighttpd, și permiţând astfel un control mai mare asupra felului în care serverul web intern lucrează.

Ruby este un limbaj de programare generic, reflexiv, dinamic și orientat pe obiecte: fiecare tip de date este un obiect, inclusive clasele și tipurile pe care multe alte limbaje le consideră primitive (cum ar fi tipul întreg, boolean și “nil”). Fiecare funcție reprezintă o metodă. Variabilele desemnează referințe la obiecte, nu obiectele în sine. Ruby suportă moștenirea, dar nu moștenirea multiplă, totuși clasele pot importa module.

Ruby on Rails (RoR sau **Rails)**, este o platformă de dezvoltare web pentru limbajul de programare Ruby care facilitează construirea de aplicații web într-un mod mai accelerat.

Capitolul 3. Proiectarea aplicației

Aplicația este practic împarțită în două aplicații distincte ce comunica una cu alta prin intermediul unui API.

Pe partea de back-end se regăsește o aplicație de tip rails-api, structurată după template-ul MVC.

Ca și modele avem **User**: ce ține informații cu privire la utilizator, **Location**: ce conține informațiile tuturor locațiilor vizitate de utilizator, **Photo**: ce conține informații cu privire la pozele facute de utilizator într-o anumită locație.

Aplicația conține un singur controller (**UserController**) ce are ca scop captarea datelor și introducerea acestora în baza de date, dar și servirea lor la cerere sub forma unui JSON.

Aceasta se conectează la o baza de date Oracle ce rulează pe o mașină virtuală la adresa 192.168.1.4:1542. Aplicația rails-api ruleaza pe un subdomeniu localhost și anume **http://api.localhost:3000.**

Aplicația pune la dispoziție urmatoarele metode precum **login**, **get_data, get_locations, save_locations, get_most_traveled_period** etc.

Prima metoda apelata va fi cea de login, în urma căreia se returnează id-ul utilizatorului, necesar pentru a putea apela celelalte metode. Spre exemplu pentru a putea obține locurile vizitate de utilizator se va face un request de forma **http://api.localhost:3000/v1/user/:user_id/locations.**

Pe partea front-end se găsește o aplicație de tip AngularJS care este structurată după template-ul MVC. Aceasta are două view-uri principale și două controllere atașate view-urilor respective. Se folosește de 4 servicii pentru a comunica și a transmite către client datele furnizate de api-ul REST.

Primul controller are ca scop livrarea conținutului static către clientul neautentificat și aplearea serviciului de autentificare. Autentificarea se face cu ajutorului Facebook-API obținând un JSON ce conține numele utilizatorului, e-mail-ul și orașul natal, datele fiind ulterior trimise prin serviciul REST către baza de date prin metoda HTTP POST.

Al doilea controller comunică cu Google maps API pentru randarea hărții și a locațiilor obtinute din back-end. Acestuia îi sunt atașate 2 servicii, unul care se ocupă de logica din spatele hărților și unul care înglobează acțiuni asupra utilizatorului.

Pe partea de manipularea DOM-ului și interacțiune cu utilizatorul aplicația se folosește de directivele Angular care setează șabloane și funcții specifice

pentru blocuri importante din cadrul view-ului. Aplicația rulează pe **http:// localhost:9000.**

Capitolul 4. Implementarea aplicației

Aplicația front-end a fost dezvoltată cu ajutorului mediului de dezvoltare WebStorm, iar cea de back-end a fost dezvoltată utilizând RubyMine, ambele fiind produse de JetBrains.

Fiind două aplicații separate, am întâmpinat probleme majore în realizarea comunicației dintre acestea. Mai exact, am întampinat probleme de CROS-Origin-Request. Soluția găsită a fost modificarea headerelor trimise, astfel încât sa fie acceptate request-uri de la orice sursă.

Interfața cu utilizatorul se realizează cu ajutorul aplicației de front-end, acesta având parte de o interacțiune simplă și facilă.

Capitolul 5. Concluzii

Inițial am dorit a realiza o singură aplicație. Din motive legate de modul de lucru, dar și de învățarea unor noi tehnologii, am decis a separa conceptele, astfel ajungând la forma actuală.

Aplicația este structurată astfel încât să permită cu ușurință exitinderea acesteia. Ar putea fi implementate module pentru captarea pozelor și pentru generarea mai multor rapoarte cu privire la călătoriile utilizatorului. De asemenea, ar putea fi integrate alte API-uri pentru călătorii astfel încât, învățând comportamentul utilizatorului și preferințele acestuia, să i se poată sugera eventuale trasee turistice.

Capitolul 6. Bibliografie

1. AngularJS API Reference https://docs.angularjs.org/api
2. DevDocs AngularJS Documentation http://devdocs.io/angular/
3. Ruby on Rails Documentation http://rubyonrails.org/documentation/
4. Google maps Javascript API Documentation https://developers.google.com/maps/documentation/javascript/places
5. Facebook Developers Documentation https://developers.facebook.com/docs/