

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Grega Pompe

Priporočilni sistem za izletniške točke

DIPLOMSKO DELO
VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

Ljubljana, 2011



Št. naloge: 00112/2011

Datum: 05.04.2011

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **GREGA POMPE**

Naslov: **SISTEM ZA PRIPOROČANJE IZLETNIŠKIH TOČK**
A SYSTEM FOR RECOMMENDATION OF TRIP DESTINATIONS

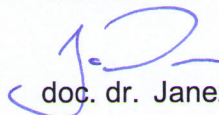
Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Z razvojem socialnih omrežij se na spletu pojavlja vedno več storitev, ki uporabniku predlagajo različne produkte, aktivnosti, spletne strani in podobno na podlagi priporočil, ki jih dajejo drugi uporabniki. Za Slovenijo sicer obstaja več spletnih mest z opisi različnih krajev in izletov, noben od njih pa ne premore dobre sistema, ki bi obiskovalcem strani priporočal izlete glede na njegove preference in na odzive uporabnikov, ki so posamezni izlet izbrali v preteklosti.


V okviru diplomskega dela predlagajte možen način delovanja takšne aplikacije in jo implementirajte. V delu se osredotočite predvsem na uporabniško izkušnjo in poskusite zasnovati sistem, ki bi se utegnil dejansko uveljaviti v praksi.

Mentor:


doc. dr. Janez Demšar



Dekan:


prof. dr. Nikolaj Zimic

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Grega Pompe

Priporočilni sistem za izletniške točke

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

Mentor: doc. dr. Janez Demšar

Ljubljana, 2011

IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani **Grega Pompe**

z vpisno številko **63060351**

sem avtor diplomskega dela z naslovom: Priporočilni sistem za izletniške točke

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal/-a samostojno pod mentorstvom doc. dr. Janez Demšar
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne _____

Podpis avtorja:

Zahvala

Zahvaljujem se mentorju na Fakulteti za računalništvo in informatiko v Ljubljani doc. dr. Janezu Demšarju za dobre ideje, pomoč in podporo pri izdelavi diplomske naloge.

Zahvaljujem se tudi staršem, bratu Anžetu, sestri Neži in puncu Katji, ki so me ves čas študija podpirali in spodbujali.

Kazalo

| | |
|---|----|
| Povzetek | 1 |
| 1 Uvod | 2 |
| 2 Uporabljene tehnologije in orodja | 3 |
| 2.1 PHP | 3 |
| 2.2 Simple XML | 5 |
| 2.3 MySQL | 5 |
| 2.4 Adobe Dreamweaver | 6 |
| 2.5 Eclipse + PHP Development Tools | 8 |
| 2.6 Google Maps API Family | 9 |
| 3 Razvoj in uporaba spletne aplikacije | 12 |
| 3.1 Zbiranje podatkov | 12 |
| 3.2 Baza podatkov | 14 |
| 3.3 Ocenjevanje predlogov za izlet | 16 |
| 3.3.1 Splošna ocena izleta | 16 |
| 3.3.2 Ujemanje lastnosti izleta z interesi uporabnika | 17 |
| 3.3.3 Razdalja do izleta | 18 |
| 3.3.4 Poizvedba SQL | 18 |
| 3.4 Uporabniški vmesnik | 20 |
| 3.5 Primeri uporabe aplikacije | 22 |
| 3.5.1 Gosti | 22 |
| 3.5.2 Registrirani uporabniki | 23 |
| 3.6 Spletni urejevalnik vsebine | 25 |
| 4 Sklepne ugotovitve | 27 |
| Kazalo slik | 28 |
| Kazalo tabel | 28 |
| Literatura in viri | 29 |

Seznam uporabljenih kratic in simbolov

- SQL – Structured query language, programski jezik za delo s podatkovnimi bazami
- PHP – odprtokodni programski jezik, ki se uporablja za razvoj dinamičnih spletnih strani
- AJAX – asinhroni JavaScript in XML je skupina medsebojno povezanih spletnih razvojnih tehnik, uporabljenih za ustvarjanje interaktivnih spletnih aplikacij
- CSS - Cascading Style Sheets (kratica CSS) so predloge, ki določajo izgled spletnih strani
- HTML - Hyper Text Markup Language je označevalni jezik za izdelavo spletnih strani
- XML - Extensible Markup Language, razširljiv označevalni jezik, ki nam omogoča format za opisovanje strukturiranih podatkov
- IDE - integrated development environments, okolje za pisanje in razhroščevanje kode
- URL - enolični krajevnik vira (angleško Uniform Resource Locators) je naslov spletnih strani v svetovnem spletu
- GPS - Global Positioning System (sistem globalnega določanja lege)
- WYSIWYG - What you see is what you get (kar vidiš to dobiš) običajno označuje razvojna okolja, ki znajo grafično prikazati rezultat kode in omogočajo vizualno urejanje

Povzetek

Cilj diplomskega dela je bil izdelati sistem, ki na podlagi ocen in lokacije uporabnikov, na čim bolj avtomatiziran način in s čim manjšo obremenitvijo uporabnika, najde primeren predlog za izlet. Na spletu namreč obstajajo spletni portali z dolgim seznamom različnih izletov, pri katerih je potrebno, da uporabnik vsaj približno ve, kaj si želi. Pri izdelavi spletne aplikacije je uporabljen programski jezik PHP v povezavi z bazo podatkov MySQL. Ocenjevalni sistem temelji na splošnih ocenah izletov, ki nastanejo na podlagi pozitivnih in negativnih ocen uporabnikov, razdalji med uporabnikom in izletom ter ujemanjem uporabnikovih interesov z lastnostmi izleta. Rezultat diplomskega dela je uporabniku prijazna spletna aplikacija, ki uporabnika spoznava, ocenjuje izlete na podlagi izkušenj uporabnikov in nudi različne predloge glede na lastnosti in zgodovino uporabnika.

1 Uvod

Gibanje je ena izmed glavnih potreb našega telesa in duha, saj pripomore k zdravju in dobremu počutju. Ker ljudje vedno radi združujemo prijetno s koristnim, so izleti v naravo že od nekdaj pomemben del našega življenja. Poleg tega, da izleti pripomorejo k dobri telesni pripravljenosti, blagodejno delujejo tudi na naše psihično počutje. Kljub temu, da ljudje na izletih uživamo, se nanje ne odpravljamo tako pogosto, kot bi si mogoče želeli. Razlogov za to je več. Izletov v bližnjo okolico smo se že naveličali, izletov kam dlje pa ne poznamo. Obstaja kar nekaj knjig na tematiko izletov po Sloveniji, vendar po njih ne posežemo pogosto, saj se redkim ljubi prebrskati cele knjige, da bi našli primeren izlet.

Na spletu obstaja več kvalitetnih spletnih strani, ki obsegajo obširne kataloge izletov in kvalitetne opise poti s komentarji. Na različnih forumih se odpirajo teme, kjer uporabniki sprašujejo za nasvet, kam na izlet. To ljudje počnejo tudi v realnem življenju, in sicer tako, da znance povprašajo za nasvet. Katalogi izletov so seveda zelo koristni, a v prvi fazi, ko človek nima ideje kam bi šel na izlet, so lahko, podobno kot knjige, preobsežni in neodločenemu uporabniku ne znajo pomagati. Manjka torej aplikacija, ki v zelo hitrem času uporabniku ponudi idejo za izlet, ki je za uporabnika relevantna. Namen diplomske naloge je razviti spletno aplikacijo, ki spozna uporabnika z minimalno količino direktnih vprašanj in s pomočjo teh podatkov predlagati čim boljše ideje za posameznega uporabnika.

Spletno stran smo zastavili zelo usmerjeno na poglavitno funkcijo t.j. ponuditi uporabniku idejo za izlet. Uporabniku so ponujene osnovne informacije o izletu, za kaj več pa ima povezave na zunanje vire. Nabor izletov smo si z dovoljenjem sposodili z uradnega slovenskega turističnega informacijskega portala www.slovenia.info, ki pa se bo s časom lahko dopolnjeval. Predlagani izleti imajo praviloma fotografijo, kratek opis, preračunano razdaljo in pot do izleta. Prednost naše spletne aplikacije je torej preprostost uporabe. S tem, ko uporabniku zmanjšamo količino podanih informacij, mu olajšamo izbiro, kar pa je pogosto bistveno pri odločanju.

2 Uporabljene tehnologije in orodja

Spletna stran je zasnovana v programskem jeziku PHP v navezi z bazo podatkov v MySQL-u. Gre za preverjeno odprtokodno kombinacijo, ki je brezplačna in deluje zanesljivo. Uporabljen je tudi JavaScript z AJAXom za boljšo uporabniško izkušnjo. Za oblikovanje je uporabljen izključno CSS, HTML pa za vsebino. Slikovno gradivo in grafike so narejene s pomočjo Adobe Photoshopa. Za pisanje in urejanje kode smo uporabljali razvojno okolje Eclipse, za oblikovni del pa Adobe Dreamweaver. Testno okolje in na koncu tudi javni strežnik smo postavili s pomočjo orodja XAMP, ki v paketu namesti PHP, MySQL in HTTP strežnik Apache.

2.1 PHP

PHP[1] je zelo razširjen odprtokodni spletnih programski jezik, ker je preprost in ima veliko podporo. Zanj je napisanih veliko dodatnih knjižnic, ki pa jih zaradi odličnega osnovnega nabora pri splošnih nalogah niti ne potrebujemo. Primerjamo ga lahko z Microsoftovim ASP in Microsystemovim JSP. Njegova sintaksa je podobna ostalim strukturiranim jezikom.

Sintaksa uporablja več posebnih znakov. Na primer Dolar (\$) označuje začetek imena spremenljivke. V HTML dokumentu začnemo pisati kodo z `<?php` značko, zaključimo pa jo z `?>`. Spremenljivke se tipično začnejo z malo črko. V primeru, da je ime spremenljivke sestavljeno iz več besed, pa se le-te združijo s podčrtajem ali pa so od druge besede dalje velike začetnice. Koda PHP znotraj praznega HTML dokumenta, ki primerja dve spremenljivki in izpiše enakost, zglada takole:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>PHP test enakosti</title>
</head>
<body>
<?php
    $a = 12
    $b = 15
    if ( $a == $b ) {
        echo "Spremenljivki sta enaki!";
    } else {
        echo "Spremenljivki sta različni!";
    }
?>
</body>
</html>
```

PHP je bil razvit leta 1994, kot dodatne skripte za programski jezik Perl. Njegova popularnost je začela naraščati z verzijo PHP 3, ki je bila uradno izdana leta 1998. Njegov razvoj je še danes aktiven, zadnja stabilna izdana verzija PHP 5 pa je uporabljena v naši diplomski nalogi.

Za hitrejši in preglednejši razvoj smo uporabili tudi svoje funkcije. Ena od njih je `sql($query)`, ki sprejme niz znakov, ki predstavljajo poizvedbo SQL. Funkcija izvede poizvedbo in vrne prvo polje prve vrstice. V primeru da poizvedba ne vrne rezultata ali pa vrne napako, funkcija vrne negativno (`false`). Funkcija predvideva, da so podatki o povezavi že navedeni v spremenljivki `$conn`.

```
function sql($query) {
    global $conn;
    $res = mysql_query($query, $conn);
    if(!$row = mysql_fetch_array($res))
        return $row[0];
    return false;
}
```

Funkcija `getCoordinates($naslov)` vrne tabelo s podatki o geografski višini in širini točke ter nizom znakov, ki označuje lokacijo (običajno poštni naslov). Če lokacije ne najde, vrne 0. Funkcija najprej spremeni podan naslov v niz primeren za poizvedbo, tako da presledke zamenja za pluse (+). Ta niz nato vstavi v URL naslov, ki je pravzaprav klic google maps APIja, ki vrne XML datoteko z geografskimi podatki. Funkcija `simplexml_load_file()` vrnjeno datoteko XML pretvori v objekt, s katerim lažje upravljamo in se sprehajamo po strukturi. Iz rezultata nato izlušči zelene informacije, shrani v tabelo in tabelo vrne.

```
function getCoordinates($naslov){
    $coordinates = array();
    $searchstring = str_replace(" ", "+", $naslov);
    $xml = simplexml_load_file("http://maps.googleapis.com/maps/
        api/geocode/xml?address=$searchstring,slovenija&sensor=false&re
        gion=sl&language=sl");
    if($xml->status != "ZERO_RESULTS")
    {
        $coordinates["lat"] = $xml->result->geometry->location->lat;
        $coordinates["lng"] = $xml->result->geometry->location->lng;
        $coordinates["address"] = $xml->result->formatted_address;
        //print $coordinates["address"];
        return $coordinates;
    }
    else return 0;
}
```

2.2 Simple XML

Za sintakso analizo XML dokumentov smo uporabili PHPjev Simple XML [2], ki je del jedra PHP 5. Uporaba je preprosta in omogoča dostop do večine informacij s par vrsticami kode. Pri zahtevnejših strukturah hitro naletimo na omejitve, vendar je za prebiranje dobro definiranih preprostih struktur, kot se pojavljajo v tej diplomski nalogi zelo primeren. Simple XML dokument XML pretvori v objekt sestavljen iz hierarhije elementov, ki jih je našel v dokumentu. Za primer uporabe bomo uporabili dokument XML, ki predstavlja sporočilo SMS.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SMS>
<od>041238297</od>
<za>031289132</za>
<dolžina>23</dolžina>
<vsebina>Lepe pozdrave iz morja!</vsebina>
</SMS>
```

Iz podanega XMLja želimo izpisati imena polj in njihove vrednosti v človeku prijaznejši obliki. To storimo na sledeč način:

```
<?php
$xml = simplexml_load_file("sms.xml");
echo <h1>$xml->getName() . "</h1>";
foreach($xml->children() as $child)
{
    echo $child->getName() . ": " . $child . "<br />";
}
?>
```

Rezultat zgornje kode bi bil:

SMS

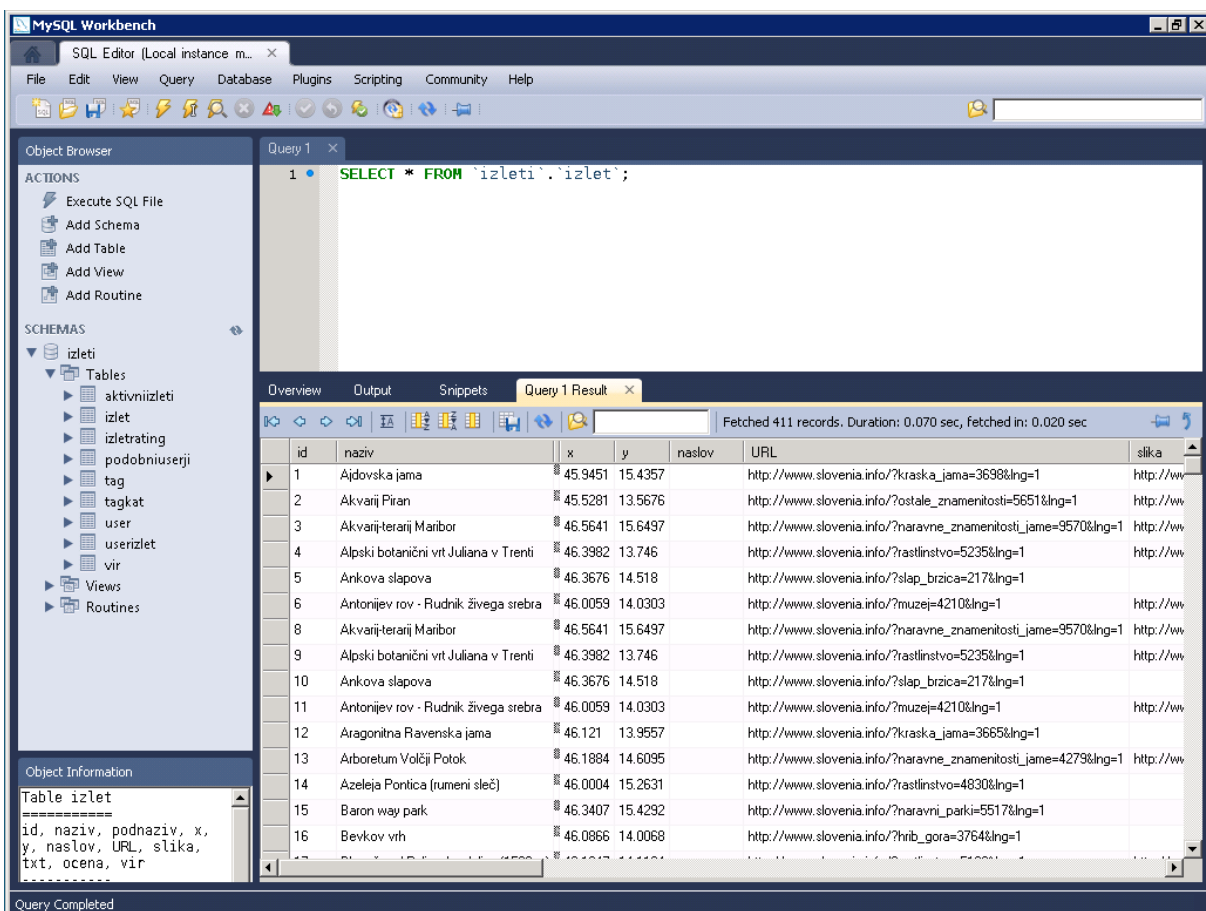
```
Od: 041238297
za: 031289132
dolžina:23
vsebina: Lepe pozdrave iz morja!
```

2.3 MySQL

MySQL [3] je odprtokoden sistem za upravljanje s podatkovnimi bazami. Deluje po principu odjemalec – strežnik. Podpira večino operacij, ki jih podpirajo večje in plačljive podatkovne baze in v hitrosti ne zaostaja veliko, če sploh. Napisan je v jeziku C in C++ in podpira več

programskih niti oz procesorjev. Je brezplačen in ima dobra orodja za urejanje in upravljanje. Vse te lastnosti ga naredijo idealnega za manjše projekte. Podpirajo ga vsi bolj razširjeni operacijski sistemi, uporabljajo pa različne zelo popularne spletne strani kot so YouTube in Wikipedia.

Za manipuliranje z bazo smo uporabili orodje MySQL Workbench, ki omogoča skoraj vse, kar razvijalec potrebuje za upravljanje baze. S pomočjo MySQL Workbencha smo kreirali bazo, ustvarili tabele in relacije s pomočjo vizualnega povezovanja. Ta program poleg poizvedb (prikazuje Slika 1), omogoča tudi urejanje podatkov v tabelah, spreminjanje lastnosti tabel, prikazuje hitrost izvajanja posameznih poizvedb, ustvarjanje pogledov (views),...



Slika 1: Uporabniški vmesnik MySQL Workbencha

2.4 Adobe Dreamweaver

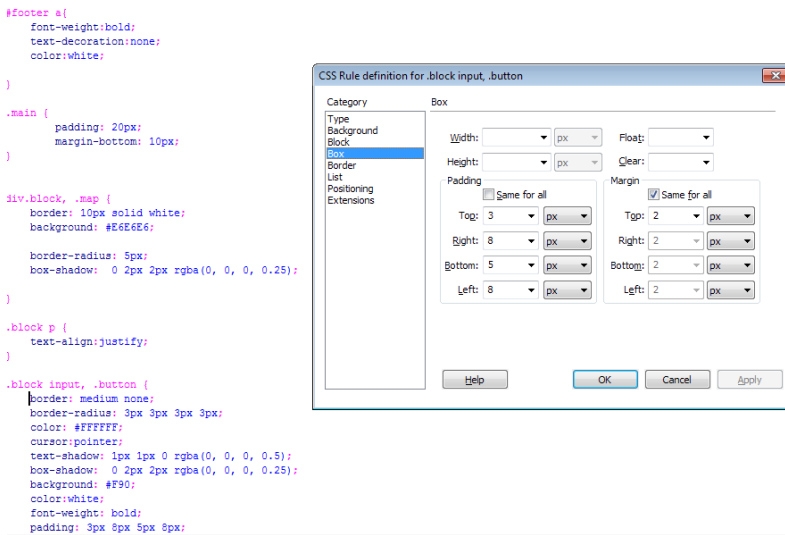
Dreamweaver [4] je program za spletno oblikovanje in postavitvev HTML strani, razvijanje spletnih aplikacij in urejanje CSS predlog. Je eden izmed najbolj popularnih HTML urejevalnikov. Prvotno je bil razvit za sisteme Macintosh, takrat še pod lastništvom

Macromedije, a je kmalu dobil verzijo za PCje. Ko je Adobe leta 2000 program kupil, ga je tesneje povezal s svojimi ostalimi programi, kot je Adobe Photoshop, in s tem je postal še bolj učinkovit in uporaben.



Slika 2: Prikaz hkratnega urejanja kode s tekstovnim urejevalnikom in z grafičnim prikazom v Dreamweaverju

Ima zelo dober vizualni urejevalnik, prikazan na Sliki 2, ki v realnem času prikazuje spremembe v kodi in obratno. Zelo uporabna funkcija, ki jo ima Dreamweaver, je napredno iskanje in popravljanje besedila oz. kode z regularnimi izrazi. Pozna različno barvanje kode in zna ponuditi kodo, ki jo verjetno želite napisati (inteli-sense), vendar žal samo pri nekaterih jezikih. Pri PHPju je ta podpora zelo osnovna, zato pri razvijanju PHP kode v tem projektu ni bil uporabljen. Dreamweaver je bil uporabljen predvsem za urejanje slogov CSS in grafično postavitev spletne strani. Slika 3 prikazuje urejanje slogov CSS.

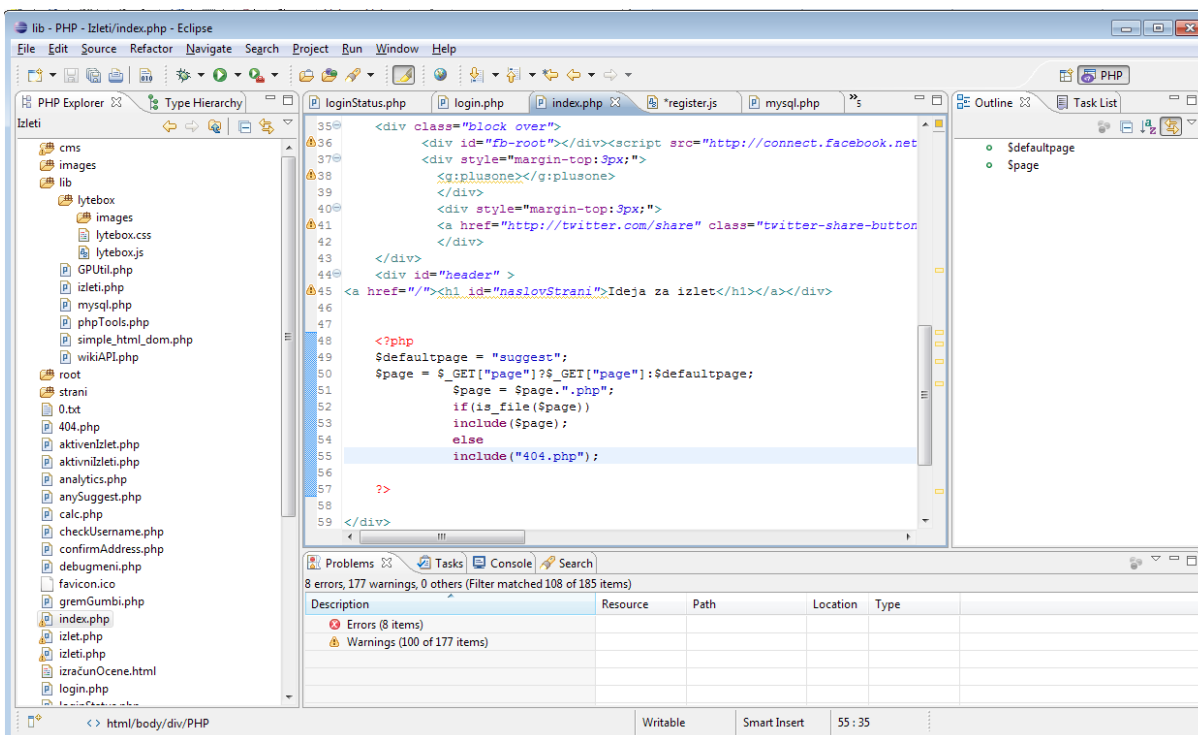


Slika 3: Urejanje slogov CSS v Dreamweaverju

2.5 Eclipse + PHP Development Tools

Eclipse [5] je integrirano razvojno okolje, napisano v Javi, in je v osnovi namenjen razvijanju programov v Javi, a z vtičniki podpira več programskih jezikov vključno s PHP in JavaScript. Je brezplačen, odprtokoden in podprt na vseh popularnih operacijskih sistemih. Nudi izdatno podporo pri pisanju kode, pozna vse funkcije in spremenljivke znotraj projekta, zna hitro prikazati opis funkcije (če je podan), označi sintaktične napake in podobno. Omogoča zelo dobro prilagodljivost okolja, tako, da si lahko vsak uporabnik ustvari okolje, v katerem je najbolj učinkovit. PHP Development Tools [6] je vtičnik za Eclipse, ki zajema vsa orodja in doda podporo za PHP.

Kombinacijo Eclipse in PHP Development Tools smo uporabili za pisanje PHP kode, zaradi zmogljivega inteli-sensa in odličnega okolja. Primer urejanja dokumenta PHP prikazuje Slika 4. Žal ta brezplačna rešitev ne ponuja razhroščevanja, tako, da je bila večina iskanja napak opravljena preko izpisov spremenljivk na ekran, kar je razmeroma zamuden proces. Program je bil uporabljen tudi za pisanje kode v JavaScriptu.



Slika 4: Uporabniški vmesnik razvojnega okolja Eclipse

2.6 Google Maps API Family

Google Maps API Family vsebuje več koristnih vmesnikov, ki ponujajo delo z zemljevidi in geografskimi podatki. V diplomski smo uporabili Maps API Web service [7], ki je zbirka HTTP vmesnikov za pridobivanje teh podatkov. Zajema sledeče vmesnike:

- Directions API – Načrtovanje poti
- Distance Matrix API – Razdalje med točkami
- Elevation API – Nadmorska višina
- Geocoding API – Geografske koordinate
- Places API – Zanimive točke

Za našo diplomsko nalogo sta bila najbolj zanimiva Geocoding API in Directions API. Zahteva se poda v obliki URL naslova s podanimi parametri, strežnik pa vrne (v našem primeru) XML datoteko z zahtevanimi podatki. Vsi ti podatki so za našo aplikacijo zelo pomembni, ker je eden od faktorjev ocene izleta tudi razdalja med lokacijo uporabnika in ciljem izleta.

Primer uporabe v aplikaciji za naslov »Ljubljana«. Funkcija kliče URL

`http://maps.googleapis.com/maps/api/geocode/xml?address=Ljubljana, Slovenija&sensor=false®ion=sl&language=sl`

Zgornji URL je sestavljen iz več delov:

Naslov: <http://maps.googleapis.com/maps/api/geocode/xml>

Parametri:

- `Sensor = false`, obvezen parameter, ki pove, ali zahtevek prihaja iz naprave z GPS senzorjem
- `Region = sl`, neobvezen podatek, ki vmesniku pomaga pri iskanju
- `Language = sl`, obvezen podatek, ki vmesniku nastavi jezik, v katerem želi odgovor
- `Address = Ljubljana, Slovenija`, obvezen parameter, ki opiše kraj katerega koordinate iščemo.

API vrne sledeč rezultat v obliki XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<GeocodeResponse>
  <status>OK</status>
  <result>
    <type>locality</type>
    <type>political</type>
    <formatted_address>Ljubljana, Slovenija</formatted_address>
    <address_component>
      <long_name>Ljubljana</long_name>
      <short_name>Ljubljana</short_name>
      <type>locality</type>
      <type>political</type>
    </address_component>
    <address_component>
      <long_name>Slovenija</long_name>
      <short_name>SI</short_name>
      <type>country</type>
      <type>political</type>
    </address_component>
    <geometry>
      <location>
        <lat>46.0514263</lat>
        <lng>14.5059655</lng>
      </location>
      <location_type>APPROXIMATE</location_type>
    </geometry>
    <viewport>
      <southwest>
        <lat>45.9740608</lat>
        <lng>14.4086264</lng>
      </southwest>
      <northeast>
        <lat>46.1459915</lat>
        <lng>14.7553364</lng>
      </northeast>
    </viewport>
    <bounds>
      <southwest>
        <lat>45.9740608</lat>
        <lng>14.4086264</lng>
```

```
</southwest>  
<northeast>  
  <lat>46.1459915</lat>  
  <lng>14.7553364</lng>  
</northeast>  
</bounds>  
</geometry>  
</result>  
</GeocodeResponse>
```

Med podatki, ki jih vrnen XML zajema, so status, ki opiše morebitne napake, ime točke v treh oblikah, koordinate sredine najdenega območja in koordinate skrajnih točk območja v vse štiri smeri sveta.

3 Razvoj in uporaba spletne aplikacije

Spletna stran je bila načrtovana z mislijo na uporabnika. Na podlagi pridobljenih informacij iz pogovorov z znanci in naših pričakovanj smo si zamislili spletno stran, ki deluje sveže, hitro opravi svoj namen in od uporabnika ne pričakuje, da se bo na njej dalj časa zadrževal. Od uporabnika se pričakuje, da jo bo obiskal, ko mu zmanjka idej, ali pa želi še kakšno alternativo.

Ker lahko uporabniku predlagamo precej bolje, če o njem kaj vemo, smo usmerili glavni del strani na prijavljene uporabnike. Ker pa uporabniki radi stvari preizkusijo, preden se prijavijo, je stran odprta za vse, vendar pri neregistriranih uporabnikih nima polne funkcionalnosti.

3.1 Zbiranje podatkov

Za pravo uporabnost aplikacije je zelo pomembno, da so v bazi že v osnovi kvalitetni predlogi, med katerimi lahko potem aplikacija izbira in ponudi najboljši predlog za posameznika. Ker različni kriteriji in filtri hitro zmanjšajo število primernih izletov, je zelo pomembno tudi število različnih izletov. Za spodnjo mejo smo si postavili 300 izletov. Namen te diplomske naloge ni preučevanje izletov in subjektivno ocenjevanje le-teh. So le delovno sredstvo, s katerim operira umetna inteligenca aplikacije.

Kot potencialne vire smo si zamislili spletne strani z različnimi idejami za izlete in knjige. Prednost knjig je njihova urejenost in konsistentnost podatkov, slabost pa ročno vnašanje podatkov v bazo aplikacije. Prednost spletnih strani je v možnosti elektronskega prenosa podatkov v bazo, slabost pa, da so večinoma slabše urejene in nepopolne. Ponekod manjkajo podatki, opisi strani so zelo različni, itd...

Po pregledu kar nekaj knjig smo prišli do zaključka, da med izbrano literaturo žal ni knjige, ki bi imela dovolj predlogov in pravih informacij o izletih. Več knjig skupaj bi sicer ponudilo dovolj zanimivih idej, a so si opisi, količina informacij in stopnja podrobnosti med seboj tako različni, da bi jih težko poenotili. Tudi dela s prepisovanjem bi bilo enostavno preveč, zato bi se lahko manj posvetili bistvenemu delu aplikacije. Pregledali smo več spletnih portalov, med drugimi <http://www.kam.si/>, <http://www.kamnaizlet.si/>, <http://www.hribi.net/>, <http://www.naizlet.si/>, <http://www.slovenia.info/>, <http://www.gremoven.com/>, <http://www.planinec.si/>, <http://www.kodkam.si/>. Večino spletnih strani smo hitro izločili, ker so ponujale premalo izletov. Nekatere so imele samo planinske poti, pri drugih so bili podatki nepopolni.

Kljub vsemu smo uspeli najti zelo primerno stran, in sicer Uradni slovenski turistični informacijski portal (<http://www.slovenia.info/>), ki ima praktično vse, kar smo iskali: dovolj velik nabor izletov, dokaj konsistentne opise izletov, zadovoljive fotografije izletov in geografske koordinate lokacije. Za potrebe diplomske naloge so nam dovolili uporabiti njihove vsebine, seveda pod pogojem, da navedemo vir.

Uradni slovenski turistični informacijski portal v osnovi ne omogoča izvoza podatkov, sprotno poizvedovanje pa zaradi hitrosti nikakor ne pride v poštev, tako, da smo morali za zajem in prenos podatkov poskrbeti sami. Podatke bi lahko kopirali ročno, a bi bilo to pri taki količini podatkov nesmotrno. Ker portal ne ponuja samo izletov, ampak tudi druge turistične informacije in članke, smo najprej pripravili seznam naslovov URL primernih izletov. S pomočjo raznih seznamov interesnih točk (npr. Jezera - <http://www.slovenia.info/si/Jezera.htm?jezero=0&lng=1>), ki so na strani, smo hitro prišli do seznama več kot 400 izletov. S pomočjo urejevalnika Dreamweaver smo iz dokumentov HTML dobili povezave in sestavili seznam naslovov URL.

Za delovanje strani smo potrebovali podatke o izletih v svoji bazi, zato smo napisali skripto, ki se sprehodi čez seznam naslovov URL izletov iz spletnih strani in sestavi poizvedbo SQL, ki vpiše podatke v bazo. Koda spletne strani je lepo urejena. Naslovi so znotraj značk <h1>, fotografija izleta vedno znotraj značke <div> z istim IDjem in koordinate vedno takoj za enakim tekstom. Le opisi niso standardizirani, tako, da smo jih uspeli zajeti le v bolj grobi obliki (preveč podatkov). Opise smo nato ročno pregledali in jih uredili tako, da so vsi enako oblikovani in vsebujejo podobno količino podatkov.

Vpiši URLje

```

/si/naravni-parki/Bohinjsko-jezero.htm?naravni_parki=1745&lng=1
/si/naravni-parki/Divje-jezero.htm?naravni_parki=1533&lng=1
/si/naravni-parki/Kozjanski-park.htm?naravni_parki=7514&lng=1
/si/naravni-parki/Krajinski-park-Goričko.htm?naravni_parki=4732&lng=1
/si/naravni-parki/Krajinski-park-Logarska-dolina.htm?naravni_parki=4203&lng=1
/si/naravni-parki/Krajinski-park-Sečoveljske-soline.htm?naravni_parki=4263&lng=1
/si/naravni-parki/Notranjski-regijski-park.htm?naravni_parki=7554&lng=1
/si/naravni-parki/Park-Škocjanske-jame,-Slovenija.htm?naravni_parki=3857&lng=1
/si/naravni-parki/Triglavski-narodni-park.htm?naravni_parki=4993&lng=1
/si/naravni-parki/Ledenica-na-Dolu.htm?naravni_parki=4296&lng=1
/si/naravni-parki/Smrekova-draga-pri-Predmeji.htm?naravni_parki=4184&lng=1
/si/naravni-parki/Velika-planina-(1400-1666-m).htm?naravni_parki=3421&lng=1
/si/naravni-parki/Ribnik-Vrbje.htm?naravni_parki=3990&lng=1
/si/naravni-parki/Alpski-botanični-vrt-Juliana-v-Trenti.htm?naravni_parki=5206&lng=1
/si/naravni-parki/Baron-way-park.htm?naravni_parki=5517&lng=1
/si/naravni-parki/Boč-rastišče-velikonočnice.htm?naravni_parki=5922&lng=1
/si/naravni-parki/Dravograd,-Krajinski-park-Košenjaka-Velka.htm?naravni_parki=6750&lng=1
/si/naravni-parki/Grajski-park.htm?naravni_parki=5920&lng=1

```

Submit

Slika 5: Vnosno polje za seznam URL naslovov izletov

Z namenom čim manj obremenjevati spletno stran slovenija.info, smo vse izlete s pomočjo skripte najprej prenesli na lokalni disk. Preko preprostega obrazca »Vpiši URLje« (prikazan na Sliki 5) smo preko metode POST funkciji poslali seznam naslovov URL (brez domene in protokola). Skripta se sprehodi čez seznam in vsako spletno stran shrani v svojo datoteko na disk. Izloči tudi morebitne podvojene izlete, ki so se lahko pojavili v več seznamih.

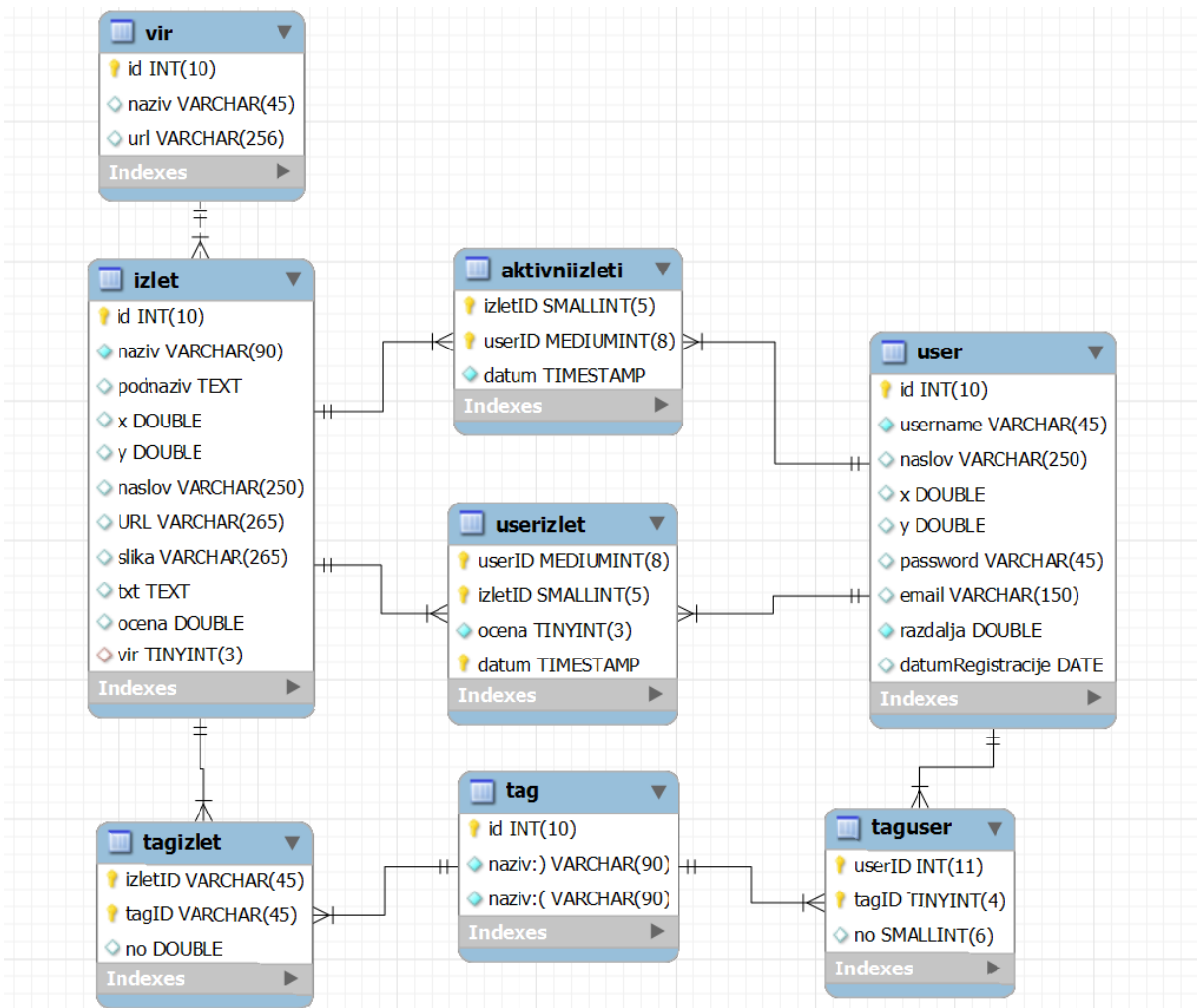
Druga skripta nato vsako datoteko posebej analizira in izpiše vstavljalne stavke SQL. Analizo dokumenta opravi s pomočjo knjižnice simple HTML DOM. Deluje na podoben način kot simple XML. Datoteko najprej naloži v spremenljivko `$html`, nato pa izlušči potrebne podatke. Koordinato `x` pridobi tako, da poišče `<div>` z IDjem `wpContent`, ki vsebuje opis. Znotraj besedila najde začetek besedila, ki govori o koordinati `x`. Celoten opis nato skrajša na znake od začetka niza "GPS Northing (N)" + 19 (kolikor je dolg niz s presledki) do naslednjih 7 znakov, kolikor znakov je dolga koordinata. Za združljivost z Google maps APIjem zamenja vejico za piko. Za konec doda še ničle, tako, da so vsi zapisi enako dolgi.

```
$element = $html->find('div.wpContent', 0)->plaintext;
$x = strpos($element, "GPS Northing (N)");
$x = str_replace(",", ".", substr($element, $x+19, 7));
$x = preg_replace('/^[^0-9.]/', '0', $x);
```

Podobno naredi za `y`. Fotografijo najde znotraj značke ``, ki je znotraj `diva wpContent`. Vzame prvi element. Za opis zajame celoten `div wpContent`, nato pa ga nekoliko očisti. Odstrani vse slike in spremeni vse lokalne povezave v absolutne. Ročno urejanje se je izvedlo skozi spletni urejevalnik vsebine.

3.2 Baza podatkov

Vsi podatki, s katerimi spletna stran operira, so shranjeni v podatkovni bazi MySQL. Shema kaže Slika 6. Smiselno so razporejeni v več tabel, ki so med seboj povezane s ključi. Podatki se ne podvajajo in vsi podatki, ki izhajajo iz teh podatkov, se preračunajo sproti, ko se kliče določena poizvedba. Ustvarili smo tudi več pogledov za boljšo preglednost poizvedb. Vse osnovne tabele imajo za primarni ključ polje »id«, povezovalne pa običajno 2 ali več tujih ključev. Kot tuj ključ smo običajno uporabili ime tabele na katero se sklicuje + ID. Polja imajo določeno velikost na podlagi predvidene uporabe in so čim manjša.



Slika 6: Shema ureditve podatkovne baze

Tabela **user** hrani glavne podatke o uporabniku, ki jih je navedel ob registraciji, poleg teh pa sta še polji x in y, ki se ob registraciji avtomatično zapišeta glede na naslov, ki ga je uporabnik vnesel. Polje razdalja predstavlja uporabnikovo povprečno razdaljo izletov, katerih se je udeležil in so mu bili všeč. Predstavljala naj bi uporabnikovo najljubšo oddaljenost za predlog izleta (bližnje morda že vse pozna, preveč oddaljeni pa ga ne zanimajo). Posodobi se ob vsaki prijavi. Za evidenčne potrebe se hrani tudi datum registracije.

Tabela **izlet**, podobno kot tabela user za uporabnike, hrani tako opisne kot tehnične podatke o izletih. X in Y sta preračunana iz podanega naslova, ali vpisana ročno (cilj veliko izletov je namreč v naravi, ki nima poštnega naslova).

Tabela **tag**. Vsak tag oziroma označba ima dva naziva odziva uporabnika. Naziv:) oz pozitiven odziv se izpiše, ko uporabnik klikne, da mu je bil izlet všeč in ga vpraša, kaj mu je bilo všeč, naziv:(pa v primeru, da mu izlet ni bil všeč.

Tabela **vir** vsebuje podatke o viru, s katerega so bili pridobljeni podatki.

Tabela **aktivniizleti** hrani aktualne podatke o izletih, pri katerih je uporabnik kliknil, da gre na izlet. Ko se vrne z izleta in ga oceni, se zapis izbriše. Za potrebe razvrščanja in morebitnih opozoril, se hrani še datum izleta.

Tabela **userizleti** je povezovalna tabela, ki hrani odnos od uporabnikov do izletov. Primarni ključ sestavljajo userID, izletID in čas (datum). Polje ocena hrani ocene in zavrnitve. V tabelo se zapišejo vsi klikli »mogoče drugič«, ki so pomembni zato, da stran ne predlaga enakega izleta.

Tabela **taguser** hrani odnos uporabnika do označb (tag-ov). Prišteva (in odšteva) število klikov, glede na to, kar je bilo uporabniku pri izletih všeč (ni bilo všeč). Na podlagi teh podatkov program lahko sklepa o uporabnikovih interesih in to uporabi pri ocenjevanju predlogov za izlet.

Tabela **tagizlet** je podobna tabela kot taguser. Hrani odnos izletov do označb, na podlagi česar lahko sklepamo o lastnostih izleta.

3.3 Ocenjevanje predlogov za izlet

Ocena predloga je kriterij, po katerem razvrstimo izlete in jih nato uporabniku po vrsti predlagamo. A tu ne gre za statično oceno izleta. Gre za izračun, ki se izvede ob vsaki prijavi uporabnika in je prilagojen samo njemu. Ocene so podane v odstotkih, za boljšo preglednost in primerljivost. Ocene so vselej skrite uporabniku, odražajo se le v vrstnem redu ponujenih izletov.

Čeprav govorimo o ocenah izletov, se je potrebno zavedati, da tu dejansko ocenjujemo predlog in ne sam izlet. Na primer izlet v Bohinj je odličen, a verjetno za marsikoga nezanimiv predlog, ker ga že pozna.

Ocena se v grobem deli na 3 ocene: splošna ocena izleta (30%), ujemanje lastnosti izleta z interesi uporabnika (40%) in razdalja do izleta (30%). Skupaj tvorijo končno oceno primernosti izleta za prijavljenega uporabnika. Izleti, ki dosežejo višjo oceno, so za uporabnika bolj primerni.

3.3.1 Splošna ocena izleta

Na splošno oceno izleta vplivajo poročila uporabnikov, kako jim je bil izlet všeč. Vsak klik na »Všeč mi je bilo« prinese izletu točko, vsak klik na »Ni mi bilo všeč« pa izletu odbije 2 točki.

To se v bazo v tabelo userizlet zapiše v obliki id trenutnega uporabnika, id izleta, ki ga ocenjuje, trenutni čas, datum in pa ocena (1 ali -2).

Če v tabeli združimo izlete in jim seštejemo ocene dobimo rezultate v smislu:

Bled 83

Kranjska Gora 65

Soline -12

Ljubljanski grad 15

Te ocene same po sebi omogočajo efektivno razvrščanje in so uporabljene v primeru, ko uporabnik ni prijavljen (torej je gost) in mu sistem predlaga izlet le na podlagi te ocene.

Če pa želimo to oceno narediti združljivo z drugima dvema ocenama, jo je potrebno postaviti v isti interval med 0 in 1. Oceno 1 ima najboljši izlet, 0 najslabši, vmesni izleti so razporejeni linearno glede na primerjavo z najboljšim in najslabšim. To storimo tako, da oceni odštejemo najmanjšo oceno in to delimo z razponom ocen (najvišja ocena – najnižja ocena). Ocene torej postanejo nekaj takega:

| | | |
|------------------|------------------------|-----|
| Minimalna ocena | | -12 |
| Maksimalna ocena | | 83 |
| Razpon | $83 - (-12) = 83 + 12$ | 95 |

Tabela 1: Razpon ocen

| Izlet | Ocena | Račun | Preračunana ocena |
|---------------------|-------|-----------------------------|-------------------|
| Bled | 83 | $83 - (-12) / 95 = 95/95$ | 1,00 |
| Kranjska Gora 65 | 65 | $65 - (-12) / 95 = 77/95$ | 0,81 |
| Soline -12 | -12 | $"-12 - (-12) / 95 = 0/95"$ | 0,00 |
| Ljubljanski grad 15 | 15 | $15 - (-12) / 95 = 27/95$ | 0,28 |

Tabela 2: Izračun ocen

3.3.2 Ujemanje lastnosti izleta z interesi uporabnika

Uporabniki pri ocenjevanju izleta (lahko) podajo še mnenje, zakaj jim je bil določen izlet všeč. Ko pri izletu ocenijo, da jim je bilo všeč npr. kultura in razgled, povedo, da ima ta izlet lep razgled in zanimive kulturne spomenike, muzeje... To sistem zabeleži v tabelo tagizlet tako, da poveča število glasov za eno pri pravi kombinaciji izetID in tagID oz. ustvari novo in ji da število 1. Pri negativnih ocenah podobno, le da se odšteje 2. Vsak glas ne potrebuje svoje vrstice, ker je za računanje ocene pomembno samo število glasov. Na ta način ima tabela vedno maksimalno št. izletov krat št. različnih označb zapisov. To omogoča hitrejše izvajanje in manjšo porabo prostora. Poleg tega, da uporabnik pove, katere lastnosti se mu zdijo

zanimive pri izletu, pove, da te lastnosti pri izletu zanimajo tudi njega v splošnem. Seveda pri prvih nekaj ocenah to morda ne drži najbolj, a se že po nekaj ocenah lahko pokaže dokaj realna slika. Podatki se hranijo na isti način kot pri izletih, tokrat v tabeli taguser.

Ko imamo vse te podatke, lahko enostavno ugotovimo, kaj je uporabniku všeč. Sistem na podlagi ocen izbere tri najbolj priljubljene označbe in ostale pusti na strani, ker so za tega uporabnika manj pomembne. Iz seznama tagizlet izbere samo tiste zapise, ki se nanašajo na označbe, ki so uporabniku všeč in njihove ocene združi. S tem dobimo seznam vseh izletov z ocenami na podlagi ujemanja. Te rezultate podobno kot pri splošni oceni transformiramo v območje 0 – 1.

3.3.3 Razdalja do izleta

Tretji del ocene predstavlja razdalja uporabnika do izleta. Ker najbližji izlet ni nujno najboljši, kar se tiče razdalje, sistem za izhodno točko oz krožnico vzame »idealno« razdaljo uporabnika. Izleti, ki so uporabniku preblizu, so namreč lahko nezanimivi, ker izlet, ki je tako blizu, sploh ni pravi izlet, ali pa je izlet zanimiv, pa ga zelo verjetno že pozna, tako da je samo ideja slaba (ocenjuje se namreč ideja, ne sam izlet). »Idealno« razdaljo do izleta sistem izračuna tako, da vzame razdalje izletov, ki so bili uporabniku všeč in izračuna srednjo vrednost.

Pri prikazu izleta stran uporabniku sicer poda informacijo o poti in razdalji po cesti, a je odzivnost te poizvedbe prepočasna, da bi se opravila za vse izlete, ko se računajo ocene. Lahko bi vnaprej izračunali razdalje za vsakega uporabnika, a bi s tem dobili veliko tabelo, ki bi jo bilo treba posodabljati v primeru menjave naslova uporabnika, dodajanju izletov in morda še kdaj. Vse to bi bilo izvedljivo, a sistem uporabi rešitev, ki je pri rezultatih le malo manj natančna, vendar veliko preprostejša in hitrejša. Po Pitagorovem izreku izračuna zračno razdaljo med točkama podanima s koordinatama. Ta operacija je dovolj preprosta in hitra, da se izvede vedno, ko se računajo ocene. Na ta način dobimo razdalje med uporabniki in izleti. Ker za računanje ocene potrebujemo razdajo od »idealne krožnice«, od razdalje odštejemo »idealno« razdaljo in vzamemo absolutno vrednost rezultata. Tako dobimo tabelo z razdaljami od uporabnikove idealne razdalje do izletov, ki jih enako kot pri drugih ocenah transformiramo v interval med 0 in 1 ter dobimo ocene razdalje. Ker so izleti s krajšimi razdaljami od idealne razdalje seveda boljši, oceno pa imajo nižjo, ocene še odštejemo od 1 in dobimo pravilne ocene.

3.3.4 Poizvedba SQL

Spodnja poizvedba vrne seznam izletov, urejenih po vrstnem redu od najboljše ideje za trenutnega uporabnika do najslabše. Spremenljivka \$id predstavlja id številko trenutnega

uporabnika. Poizvedba se deli podobno, kot opisano v prejšnjih straneh, na 3 dele, ki se na koncu v podanem razmerju združijo. Iz rezultata na koncu izloči neprimerne ideje. To so izleti, na katerih je uporabnik trenutno (jih še ni ocenil), je v zadnjih 30 dneh kliknil »Mogoče drugič« in jim s tem dal oceno 0 ali jih je ocenil z negativno v zadnjih 270 dneh.

```

SELECT razdalja.izletID as id, razdalja*0.3 + IFNULL(ocena, 0)*0.3 +
IFNULL(tag, 0)*0.4 AS skupaj

FROM
(
  # Ocena razdalje
  SELECT i.id AS izletID, 1-(abs(POW(u.x-i.x, 2)-POW(u.y-i.y, 2) -
u.razdalja)- minrazdalja) / ( maxrazdalja - minrazdalja) AS razdalj

  FROM izlet i,
  (
    SELECT i.id AS id, abs(POW(u.x-i.x, 2)-POW(u.y-i.y, 2) -
u.razdalja) AS maxrazdalja
    FROM izlet i
    LEFT JOIN user u ON u.id = $id

    ORDER BY maxrazdalja DESC
    LIMIT 1
  ) AS max,
  (
    SELECT i.id AS id, abs(POW(u.x-i.x, 2)-POW(u.y-i.y, 2) -
u.razdalja) AS minrazdalja
    FROM izlet i
    LEFT JOIN user u ON u.id = $id

    ORDER BY minrazdalja ASC
    LIMIT 1
  ) AS min
  LEFT JOIN user u ON u.id = $id
) razdalja

LEFT JOIN
(
  # Splošna ocena izleta
  SELECT u.izletID, (SUM(nemara)*2 + SUM(vsec) - minocena) /
(maxocena - minocena) AS ocena
  FROM userocenasum u,
  (
    SELECT izletID, SUM(nemara)*2 + SUM(vsec) AS maxocena
    FROM userocenasum u
    GROUP BY izletID
    ORDER BY maxocena DESC
    LIMIT 1
  ) AS max,
  (
    SELECT izletID,
SUM(nemara)*2 + SUM(vsec) AS minocena
    FROM userocenasum u
    GROUP BY izletID
    ORDER BY minocena ASC
    LIMIT 1
  ) AS min
  GROUP BY u.izletID
)

```

```

) ocene ON razdalja.izletID = ocene.izletID

LEFT JOIN
(
  # Ujemanje lastnosti izleta z interesi uporabnika
  SELECT ti.izletID, (SUM(no)-minno)/(maxno-minno) AS tag
  FROM tagizlet ti,
  (
    SELECT izletID, SUM(no) AS maxno
    FROM tagizlet
    WHERE tagID IN
      (SELECT tagID FROM usertop3tagi WHERE userID = $id)
    GROUP BY izletID
    ORDER BY maxno DESC
    LIMIT 1
  ) maxno,
  (
    SELECT izletID, SUM(no) AS minno
    FROM tagizlet
    WHERE tagID IN
      (SELECT tagID FROM usertop3tagi WHERE userID = $id)
    GROUP BY izletID
    ORDER BY minno ASC
    LIMIT 1
  ) minno
  WHERE tagID IN (SELECT tagID FROM usertop3tagi WHERE userID = $id)
  GROUP BY izletID
) tagi ON ocene.izletID = tagi.izletID

# Izloči izlete, ki jih je uporabnik označil kot slabe
AND razdalja.izletID NOT IN
(
  SELECT izletID
  FROM userizlet
  WHERE userID = $id
  AND ( (ocena = 0 AND datum > NOW()- INTERVAL 30 DAY)
  OR (ocena = -2 AND datum > NOW()- INTERVAL 270 DAY) )
)

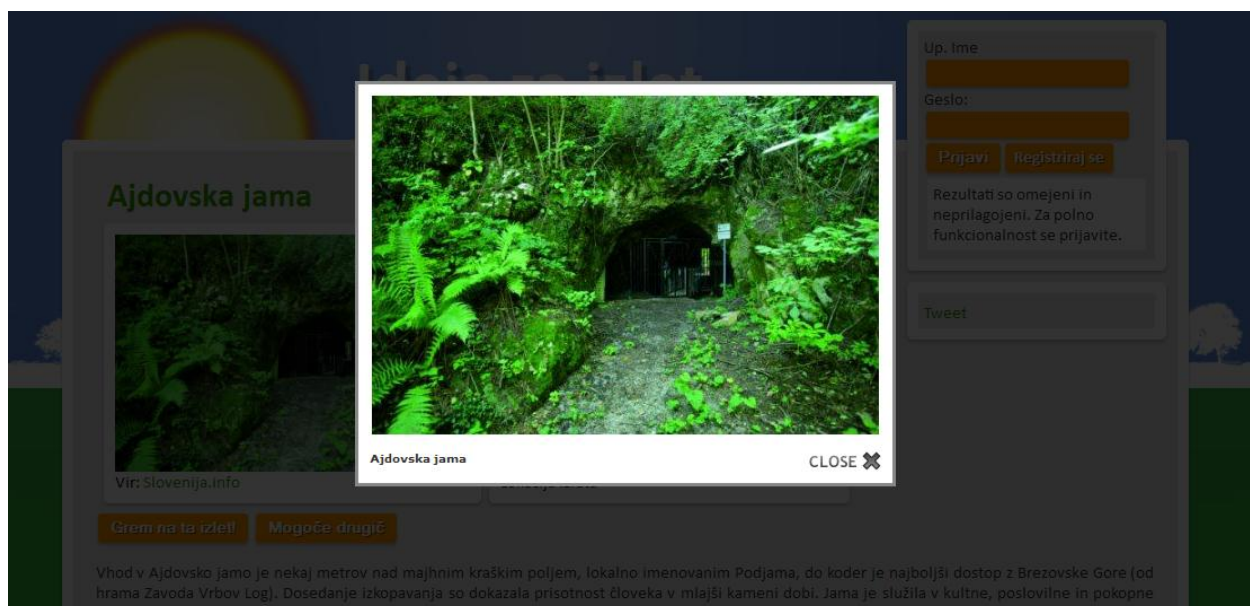
# Izloči izlete, na katerih je uporabnik trenutno
AND razdalja.izletID NOT IN
(
  SELECT izletID
  FROM aktivniizleti a
  WHERE userID = $id
)
ORDER BY skupaj DESC

```

3.4 Uporabniški vmesnik

Spletna stran je oblikovana lahko, prijetno na oko. Uporabljene barve so svetle, grafika spominja na sončen dan. Glavna vsebina je postavljena v okvir, znotraj katerega se spreminjajo predlagani izleti. Na desni strani imamo več plavajočih okvirjev, ki predstavljajo vsebino neodvisno od trenutne strani. Med njimi je obrazec za prijavo uporabnikov in gumbi socialnih omrežij. Pojavi se še seznam obiskanih izletov in izračunane razdalje. Fotografija izleta in lokacija izleta na zemljevidu Slovenije, sta podprta z LyteBox knjižnico, ki omogoča

prijetno animacijo in prikaz večje slike na sredini zaslona in zatemnitev ozadja. Prikaz večje slike prikazuje Slika 7.



Slika 7: Ogled večje slike

Izgled in pozicija plavajočih okvirjev na desni je dosežen s pomočjo stilov CSS. Širina okvirja je določena z lastnostjo `width`, notranji rob s `padding`, ki je v našem primeru širok pet pik, zunanji rob pa z `margin`. Ozadje (`background`) je nastavljeno na odtenek sive, ki je označen z vrednostjo `#E6E6E6`. Izgled obrobe določajo 3 vrednosti. Belo barvo in širino desetih pik določimo z lastnostjo `border`, zaobljenost roba pa z `border-radius`. Rahla senca, če jo lahko štejemo med obrobo, je definirana z lastnostjo `box-shadow`. Prva ničla določa število pik zamaknjenosti v desno, druga `2px` zamaknjenosti navzdol, tretja `2px` pa razpršenost. S temi vrednosti dobimo mehko senco samo na spodnji strani elementa. Na koncu je podana še barva (`0, 0, 0` je črna) in prosojnost `0.25`. Pozicija je določena z lastnostjo `float`, ki elementu pove, da plava nad ostalimi neplavajočimi elementi, vrednost `right` pa, da se drži desne strani. Z lastnostjo `clear:right` določimo, da se pojavi za ostalimi plavajočimi elementi, ki so na desni strani. Celoten razred okvirja zglada takole:

```
.plavajocOkvir {
  background: #E6E6E6;
  border: 10px solid white;
  border-radius: 5px 5px 5px 5px;
  box-shadow: 0 2px 2px rgba(0, 0, 0, 0.25);
  clear: right;
  color: #333333;
  width: 200px;
  padding: 5px;
  margin: 10px 20px 0;
  float: right;
}
```

3.5 Primeri uporabe aplikacije

Uporaba aplikacije je odvisna predvsem od tega, ali je uporabnik neprijavljen ali prijavljen. S prijavo v sistem uporabnik pridobi polno funkcionalnost aplikacije, aplikacija pa lahko zbira podatke. Gost lahko postane uporabnik tako, da se registrira. To stori na registracijski strani, do katere pride s klikom na gumb »Registriraj se«. Od gosta se zahteva uporabniško ime, geslo, e-pošto in izhodiščni naslov. Izpolnjevanje je nadzirano s pomočjo JavaScripta, ki skrbi za pravilno izpolnjenost obrazca. Ko so vsa polja pravilno izpolnjena, registracijo zaključi s klikom na »Registriraj«. Sistem takoj preveri naslov, ki ga je uporabnik vpisal in izpiše, kar je našel, kot je razvidno na Sliki 8. S tem se izognemo napakam v naslovu.

Registracija

Za uporabo spletne strani se je potrebno registrirati. Registracija je brezplačna in ne bo vzela več kot minuto.

Uporabniško ime

E-mail

Izhodiščni naslov (Domač naslov, Vikend, ...) Poljanska cesta 20, 1000 Ljubljana

Geslo

Ponovi geslo

Slika 8: Registracija novega uporabnika

3.5.1 Gosti

Ko neprijavljen obiskovalec obiše stran, ga pričaka predlog za izlet, ki je izbran naključno izmed nekaj najboljših izletov. Gostom je avtomatsko predlagan naključen izlet izmed 20 % najboljše ocenjenih izletov. Razdalje in poti se ne izračunavajo, ker je lokacija uporabnika neznana. Nabor gumbov, ki jih gost vidi, so podobni tistim, ki jih vidi registriran uporabnik, a je njihova akcija ponekod drugačna. V primeru, da gost klikne na gumb »Grem na izlet«, ga stran preusmeri na stran za registracijo novega uporabnika. Ker sistem temelji na spoznavanju posameznih uporabnikov, sistem dejanj gosta ne beleži. Beleži jih ne tudi zato, ker gre lahko za avtomatske programe (robote), ki brskajo po spletu in lahko s svojim naključnim obnašanjem pokvarijo realne podatke.

The screenshot shows a web page with a blue header and a green footer. The main content area is white and contains the following elements:

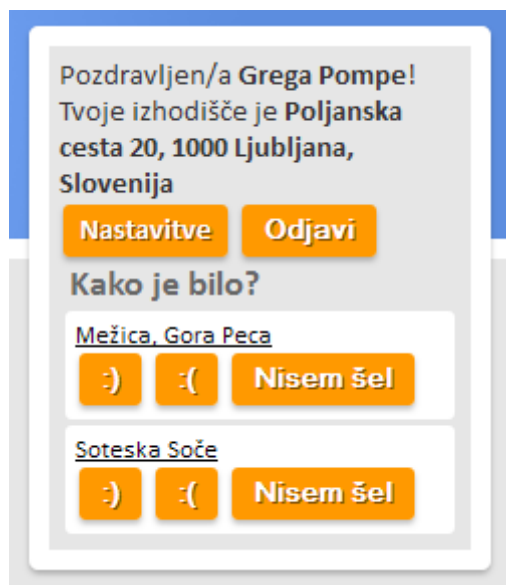
- Header:** A large yellow sun icon on the left and the title "Ideja za izlet" in white text on a blue background.
- Section Header:** "Dolina Sopote" in green text.
- Image:** A landscape photograph of a valley with mountains in the background.
- Map:** A map of Slovenia with a red pin indicating the location of the trip. The map shows major cities like Ljubljana, Zagreb, and Trieste.
- Text:** A short description of the valley, mentioning its length (18 km), elevation (760 m), and the fact that it is mostly forested (70% cover). It also mentions that the valley is easily accessible by car.
- Buttons:** Two orange buttons: "Grem na ta izlet!" (I'm going on this trip!) and "Mogoče drugač" (Maybe differently).
- Registration Form:** A white box on the right side of the page with the following fields:
 - Up. Ime (Username): An orange input field.
 - Geslo (Password): An orange input field.
 - Buttons: "Prijava" (Login) and "Registriraj se" (Register).
 - Message: "Rezultati so omejeni in neprilagojeni. Za polno funkcionalnost se prijavite." (Results are limited and not adapted. For full functionality, please log in.)
 - Social Media: A Facebook Like button (16 likes) and a Send button, and a Tweet button (0 tweets).
- Footer:** "Vir: Slovenija.info" and a "Googlaj Dolina Sopot" button.

Slika 9: Grafična podoba spletne strani, kot jo vidi gost

Informacije so namenoma omejene, ker s tem spodbudijo uporabnika v registracijo, brez katere ni mogoče zagotoviti zastavljene funkcionalnosti. Vseeno je ponujeno dovolj, da uporabnika stran pritegne in ga navduši za nadaljnjo uporabo. Gost vidi fotografijo predlaganega izleta, njeno lokacijo na zemljevidu Slovenije in kratek opis. Če uporabnik želi več informacij, ima povezavo na vir izleta in pa gumb »Googlaj«, ki najde rezultate v iskalniku Google s ključnimi besedami »ime izleta«.

3.5.2 Registrirani uporabniki

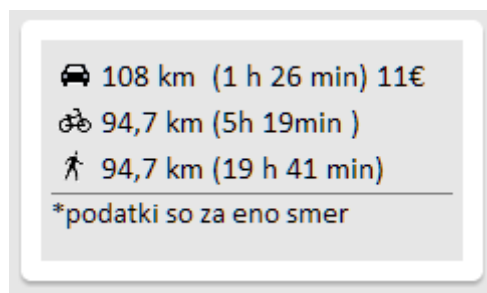
Registrirani uporabniki dobijo kvalitetnejši predlog, ker izbira predloga ne temelji samo na oceni posameznih izletov in naključju, ampak tudi na podlagi uporabnikovega izhodišča in njegovih preteklih aktivnosti na spletni strani. Če uporabniku predlog ni všeč, lahko zahteva nov predlog tako, da klikne na »Mogoče drugač«. S tem se predlog, ki mu ni bil všeč, skrije iz predlogov za en mesec. Ko uporabnik pride do predloga, ki mu je všeč, klikne »Grem na ta izlet!«. S tem se izlet pojavi na desni strani, kjer ga lahko ob povratku oceni pozitivno (smeško ☺) ali negativno (žalosten smeško ☹). Če uporabnik zaradi kakršnegakoli razloga na izlet ni šel, ima možnost »Nisem šel«. Slika 10 prikazuje obrazec. S tem se izognemo naključnim ocenam, ki jih oddajo uporabniki, ki nimajo druge izbire.



Slika 10: Seznam izletov, ki jih je uporabnik obiskal in čakajo oceno

Ko uporabnik odda oceno, ga sistem povpraša še po razlogu, zakaj mu je bil izlet všeč in se mu zahvali za glas. V tem koraku sistem dobi podatke o označbah izletov in lastnostih uporabnika. Uporabnikovih osebnih podatkov se ne zahteva in ne hrani, hrani se le podatke, ki so potrebni za delovanje aplikacije. Uporabniško ime je lahko poljubno, naslov pa je, kot rečeno, lahko poljubno natančen.

Poleg boljšega sistema predlaganja izleta za uporabnika, je deležen tudi dodatnih informacij o izletu. Na desni strani se pojavi nov plavajoč okvir (prikazuje Slika 11), ki vsebuje podatke o razdalji do izleta, merjeni po cesti, za pot z avtomobilom, kolesom ali peš. Pri vseh je ocenjen tudi čas potovanja, pri avtomobilu pa so informativno izračunani potni stroški za gorivo. Na dnu strani se prikaže še predlagana pot, po kateri lahko uporabnik pride od izhodišča do izleta.



Slika 11: Podatki o razdalji

Uporabnik lahko ureja svoje podatke preko »nastavitve«. Spreminja lahko svoj izhodiščni naslov in e-poštni naslov. Shranjen naslov se za zagotovitev točnosti prikaže še na zemljevidu. Uporabnikove nastavitve so prikazane na Sliki 12.

Uporabnik

Uporabniško ime **Grega Pompe**

Izhodišni naslov **Poljanska cesta 20, 1000 Ljubljana, Sloveni**

Email **grega.pompe@gmail.com**

Shrani

Naslov

Ljubljana

Map data ©2011 Tele Atlas

Slika 12: Urejevalnik uporabniških podatkov

3.6 Spletni urejevalnik vsebine

Urejevalnik omogoča pregled nad izleti, ki so vpisani v bazo (prikazuje Slika 13). Že vpisane izlete lahko urejamo in popravljamo njihovo vsebino (prikazuje slika 14). Polje Opis je podprto s knjižnico TinyMCE, ki prikaže kodo HTML na način, ki se bo prikazala na spletni strani (poznano tudi kot WYSIWYG). Na ta način je uredniku delo močno olajšano, vsebina je bolj pregledna in možnost napak manjša. Urednik lahko izlete tudi briše in dodaja nove. Poleg izletov je mogoč tudi pregled nad registriranimi uporabniki in brisanjem le teh.

Urejevalnik Ideja za izlet!

Uporabniki | Izleti | Izpis

Izleti

| Naziv | Zemljepisna širina | Zemljepisna dolžina | Edit | Delete |
|---------------------------------------|--------------------|---------------------|------|--------|
| Ajdovska jama | 45.9451 | 15.4357 | Edit | Delete |
| Akvarij Piran | 45.5281 | 13.5676 | Edit | Delete |
| Akvarij-terarij Maribor | 46.5641 | 15.6497 | Edit | Delete |
| Alpski botanični vrt Juliana v Trenti | 46.3982 | 13.746 | Edit | Delete |
| Ankova slapova | 46.3676 | 14.518 | Edit | Delete |
| Antonijev rov - Rudnik živega srebra | 46.0059 | 14.0303 | Edit | Delete |
| Akvarij-terarij Maribor | 46.5641 | 15.6497 | Edit | Delete |

Slika 13: Seznam izletov v urejevalniku

Izlet

izlet Alpski botanični vrt Juliana v Trenti

Leta 1926 ga je ustanovil Albert Bois de Chesne.

Opis Botanični vrt je zavarovan od leta 1951.
Zanj skrbi Prirodoslovni muzej Slovenije.

V vrtu uspeva okoli **600** različnih vrst rastlin.
Med njimi je največ alpskih in kraških vrst, med katerimi so tudi endemiti, vrste, ki rastejo samo pri nas.

Posebnost ponudbe:
Juliana je prvi in tudi edini alpski botanični vrt v naravnem okolju na Slovenskem.

Path: p Words: 74

Naslov

X

Y

Save

Slika 14: Urejanje izleta v urejevalniku

4 Sklepne ugotovitve

Pri razvoju spletne aplikacije smo se srečali z mnogimi dilemami in vprašanji. Veliko časa smo posvetili zagotavljanju preproste uporabe aplikacije. Z vsako novo idejo, kako pridobiti več podatkov od uporabnika, smo si zamislili, kako se bo uporabnik ob tem počutil. Bo morda preobremenjen, mu bo stran zaradi tega postala nezanimiva in prekomplicirana. Če je bilo mogoče podatek pridobiti brez tega negativnega učinka, smo ga sprejeli, drugače smo iskali drugo rešitev. Trudili smo se zagotoviti dovolj podatkov o izletu, na podlagi katerih se uporabnik že lahko odloči, če mu je všeč. Razvili smo več verzij, ki smo jih nato zavrgli, ker niso ustrezale zastavljenim ciljem.

Končni rezultat je spletna aplikacija, ki je izpolnila naša pričakovanja. Ker gre za splošno koristno aplikacijo, smo jo objavili v svetovnem spletu pod domeno www.idejazazilet.si, gostuje pa na domačem strežniku.

Kazalo slik

| | |
|--|----|
| Slika 1: Uporabniški vmesnik MySQL Workbench | 6 |
| Slika 2: Prikaz hkratnega urejanja kode s tekstovnim urejevalnikom in z grafičnim prikazom v Dreamweaverju | 7 |
| Slika 3: Urejanje slogov CSS v Dreamweaverju | 8 |
| Slika 4: Uporabniški vmesnik razvojnega okolja Eclipse | 9 |
| Slika 5: Vnosno polje za seznam URL naslovov izletov | 13 |
| Slika 6: Shema ureditve podatkovne baze | 15 |
| Slika 7: Ogled večje slike | 21 |
| Slika 8: Registracija novega uporabnika | 22 |
| Slika 9: Grafična podoba spletne strani, kot jo vidi gost | 23 |
| Slika 10: Seznam izletov, ki jih je uporabnik obiskal in čakajo oceno | 24 |
| Slika 11: Podatki o razdalji | 24 |
| Slika 12: Urejevalnik uporabniških podatkov | 25 |
| Slika 13: Seznam izletov v urejevalniku | 25 |
| Slika 14: Urejanje izleta v urejevalniku | 26 |

Kazalo tabel

| | |
|------------------------|----|
| Tabela 1: Razpon ocen | 17 |
| Tabela 2: Izračun ocen | 17 |

Literatura in viri

- [1] PHP. Dostopno na:
<http://en.wikipedia.org/wiki/PHP> (zadnji obisk 15.7.2011)
- [2] Simple XML. Dostopno na:
http://www.w3schools.com/PHP/php_xml_simplexml.asp (zadnji obisk 15.7.2011)
- [3] MySQL. Dostopno na:
<http://en.wikipedia.org/wiki/Mysql> (zadnji obisk 15.7.2011)
- [4] Adobe Dreamweaver. Dostopno na:
<http://en.wikipedia.org/wiki/Dreamweaver> (zadnji obisk 15.7.2011)
- [5] Elipse. Dostopno na:
[http://en.wikipedia.org/wiki/Eclipse_\(software\)](http://en.wikipedia.org/wiki/Eclipse_(software)) (zadnji obisk 15.7.2011)
- [6] PHP Development Tools. Dostopno na:
http://en.wikipedia.org/wiki/PHP_Development_Tools (zadnji obisk 15.7.2011)
- [7] Google Maps API Web Services. Dostopno na:
<http://code.google.com/intl/sl-SI/apis/maps/documentation/webservices/index.html>
(zadnji obisk 15.7.2011)
- [8] Uradni slovenski turistični informacijski portal. Dostopno na:
<http://en.wikipedia.org/wiki/PHP> (zadnji obisk 15.7.2011)