Mestna občina Celje
Komisija Mladi za Celje

PEPPER CLOUD

Raziskovalna naloga

AVTORJI:
Matevž Petan
Leonardo Vrenko
Domen Razpotnik

MENTOR:
Borut Slemenšek, univ. dipl. inž.

Celje, marec 2017

ŠOLSKI CENTER CELJE
SŠ KER

PEPPER CLOUD

Raziskovalna naloga

AVTORJI:
Matevž Petan
Leonardo Vrenko
Domen Razpotnik

MENTOR:
Borut Slemenšek, univ. dipl. inž.

Mestna občina Celje, Mladi za Celje
Celje, 2017

IZJAVA*

Mentor (-ica) , Borut Slemenik, v skladu z 2. in 17. členom Pravilnika raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje, zagotavljam, da je v raziskovalni nalogi

naslovom PODATKOVNI OBLAK,

katere avtorji (-ice) so NEJEŽ PETAN , DOMEN RAČETNIK, LEONARDO VRENKO :

- besedilo v tiskani in elektronski obliki istovetno,
- pri raziskovanju uporabljeno gradivo navedeno v seznamu uporabljene literature,
- da je za objavo fotografij v nalogi pridobljeno avtorjevo(-ičino) dovoljenje in je hranjeno v šolskem arhivu,
- da sme Osrednja knjižnica Celje objaviti raziskovalno nalogo v polnem besedilu na knjižničnih portalih z navedbo, da je raziskovalna naloga nastala v okviru projekta Mladi za Celje,
- da je raziskovalno nalogo dovoljeno uporabiti za izobraževalne in raziskovalne namene s povzemanjem misli, idej, konceptov oziroma besedil iz naloge ob upoštevanju avtorstva in korektnem citiranju,
- da smo seznanjeni z razpisni pogoji projekta Mladi za Celje

Celje, 13.3.2017 žig šole Podpis mentorja(-ice)

Podpis odgovorne osebe

*

POJASNILO V skladu z 2. in 17. členom Pravilnika raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje je potrebno podpisano izjavo mentorja(-ice) in odgovorne osebe šole vključiti v izvod za knjižnico, dovoljenje za objavo avtorja(-ice) fotografskega gradiva, katerega ni avtor(-ica) raziskovalne naloge, pa hrani šola v svojem arhivu.

KAZALO

Kazalo vsebine

Povzetek

V raziskovalni nalogi smo se odločili narediti podatkovni oblak. Sestavlja ga več ključnih komponent, kot so spletna stran ter administracijski program. Cilj podatkovnega oblaka je dostop do svojih podatkov s katerekoli naprave. Pri izdelavi smo se soočili z mnogimi izzivi, ki smo jih uspešno reševali. Čeprav je podatkovni oblak že dokončan, si želimo dodati še mnogo funkcij, ki bi izkušnjo naredile še boljšo.

Abstract

In this research paper we decided to make a data cloud. It consists of several key components, such as a website and an administration program. Our goal was that you can access data from any device. We have faced a lot of challenges, but we have managed to overcome them all. Although the data cloud is finished, we want to add many features to make the experience even better.

Ključne besede / Keywords

Visual Studio – programsko okolje

C# – programski jezik

php – programski jezik

podatkovni oblak

podatki

Kratice in okrajšave

PHP - je skriptni jezik, ki se izvaja na strežniku namenjen predvsem razvoju spletnih strani.

MySql –

1. “Structured Query Language” – najbolj standardiziran jezik za dostop do podatkovnih baz.
2. Odprto kodni sistem za urejanje podatkovnih baz.

CSS – Jezik uporabljen za urejanje prikaza spletnih strani.

C# - Programski jezik uporabljen za pisanje aplikacij v okolju .NET.

Java Script – Skriptni jezik uporabljen v spletnih straneh.

MD5 – Je kodna funkcija s 128-bitnim izhodom, uporabljena za preverjanje pravilnosti podatkov.

Zahvala

Zahvaljujmo se profesorici Kristini Radoš Janežič, prof. slov., za lektoriranje naloge.

Zahvaljujem se tudi profesorju Borutu Slemenšku za pomoč in nasvete pri izdelavi raziskovalne naloge.

Zahvaljujemo se tudi Maši Škotnik za pomoč pri izdelavi logotipa.

1 UVOD

1.1 Opis/predstavitev problema

Zaradi vedno večje popularnosti računalnikov in pametnih telefonov se je naš način življenja drastično spremenil. Ljudje smo vedno bolj navezani na svoje naprave in le redko kdo bi si še lahko predstavljal življenje brez njih. Z vsakodnevno uporabo teh raznovrstnih naprav pa vsi generiramo velike količine informacij. Zaradi te spremembe veliko strokovnjakov trdi, da človeštvo prehaja v novo obdobje - obdobje informacij.

Skoraj vsi smo se že srečali s težavo, da smo želeli dostopati do dokumentov, ki so bili na drugem računalniku, a enostavno nismo mogli. S to težavo so se dobro spopadli prenosljivi mediji kot so diskete ali USB ključki. A kljub vsem prenosljivim medijem je bila še vedno velika ovira človeški faktor, saj smo mogoče pozabili prenesti podatke na njih ali pa so se pokvarili ravno v trenutku, ko smo jih najbolj potrebovali. V zadnjem času postajajo vedno bolj popularni podatkovni oblaki, saj nam omogočajo dostop do podatkov preko interneta s katerekoli naprave. Zato smo se odločili, da bomo naredili svoj podatkovni oblak. Pri tem smo na prvo mesto postavili varnost podatkov ter zanesljivost našega sistema.

1.2 Hipoteze

Postavili smo si naslednji hipotezi raziskovalne naloge:

- Dostop do podatkov je mogoč iz vseh okolij (telefon/računalnik).
- Dosegljivost sistema naj bi bila 99 %.

1.3 Opis raziskovalne naloge

Pri izdelavi raziskovalne naloge smo uporabili različna spletna gradiva za pripravo idej in reševanje programskega dela naloge. Pomagali smo si tudi z znanjem in literaturo, ki smo jih pridobili skozi štiri leta šolanja.

2 Razvoj podatkovnega oblaka

2.1 Koraki dela

Razvoj našega podatkovnega oblaka je obsegal naslednje korake:

1. Načrtovanje:
   1.1. izdelava načrta;
   1.2. izbira programskih jezikov ter okolij;
   1.3. izbira imena ter izdelava logotipa.
2. Načrtovanje ter izdelava podatkovne baze.
3. Izdelava vseh potrebnih komponent.
4. Izdelava osnovne spletne strani za prijavo.
5. Izdelava uporabniških vmesnikov.
6. Testiranje in popravljanje napak.

3 Načrtovanje

3.1 Izdelava načrta

Najprej smo se odločali med izdelavo osebnega podatkovnega oblaka ter oblakom za javno rabo. Odločili smo se, da bomo izdelali javni podatkovni oblak, saj nam je predstavljal večji izziv. Ustvarili smo si osnovni načrt za postopno izdelavo oblaka, ki je zajemal izdelavo vseh potrebnih komponent.

3.2 Izbira programskih jezikov ter okolij

Za izdelavo našega projekta smo lahko izbrali med mnogimi programskimi jeziki in jeziki za sestavo spletnih aplikacij. Pri izdelavi slednje bomo uporabili jezike php, CSS, HTML in Java Script. Prednost teh jezikov je, da niso omejeni na določeno razvojno okolje. Ker v projektu potrebujemo tudi podatkovno bazo, smo se odločili, da bomo za njo uporabili bazo MySql. Po

pregledu in primerjavi možnosti, zahtevnosti, … z drugimi bazami (MSSQL, Firebird…) nam je ta nudila ugodnejše rešitve. Pri izdelavi administracijskega programa pa smo se odločili za uporabo jezika C#, saj smo znanje tega jezika pridobili v šoli. Uporabo tega jezika nam je omogočalo že iz šole dobro poznano programsko okolje Visual Studio.

3.3 Izbira imena ter izdelava logotipa

Zaradi pogostosti podatkovnih oblakov smo iskali unikatno ime, ki gre »rado v uho«. Za vsako ime (ki smo si ga izmislili) smo tudi preverili, če je že uporabljeno oziroma je domena že zasedena. V sodelovanju z dijakinjo srednje medijske šole v Celju smo naredili še logotip.

Slika 1 Logotip

4 Načrtovanje ter izdelava podatkovne baze

Podatkovna baza je »srce« našega podatkovnega oblaka, saj se v njej hranijo vsi podatki o uporabnikih. Načrtovanja podatkovne baze smo se lotili z veliko natančnostjo, saj je to najpomembnejši del naše aplikacije. V primeru, da ne bi bila pravilno zasnovana, bi bilo delo z njo nemogoče ali zelo zapleteno. Načrtovanje smo pričeli z enostavnim načrtom, ki smo ga narisali na papir. Nato smo morali ta načrt prepisati v programski jezik MySql. Kljub skrbnemu načrtovanju smo morali kasneje bazo še razširiti, saj so se naše potrebe z gradnjo aplikacije širile. Po vsaki spremembi smo morali tudi testirati vse funkcije baze, če še deluje pravilno.

```
Create Database oblak

Create table uporabniki
(
u_id int NOT NULL AUTO_INCREMENT(1,1)
uporabnisko varchar(255) CHARACTER SET utf8,
geslo varchar(255) CHARACTER SET utf8,
PRIMARY KEY(u_id)
)

CREATE TABLE datoteke
(
d_id int NOT NULL AUTO_INCREMENT(1,1),
pot varchar(255) CHARACTER SET utf8,
ime varchar(255) CHARACTER SET utf8,
tip varchar(255) CHARACTER SET utf8,
PRIMARY KEY(d_id)
)

CREATE TABLE uporabnik_datoteka
(
u_id int FOREIGN KEY REFERENCES uporabniki(u_id),
d_id int FOREIGN KEY REFERENCES datoteke(d_id),
)


CREATE TABLE vrsta
(
d_id int FOREIGN KEY REFERENCES datoteke(d_id),
tip varchar(20),
ikona varchar(255),
PRIMARY KEY(tip)
)
```
Slika 2 Primer kode podatkovne baze

5 Izdelava vseh potrebnih komponent

5.1 Servisi

Servisi so pomembnejši del našega oblaka, saj povezujejo uporabnika z njegovimi datotekami. Prav tako povezujejo administracijsko aplikacijo z vsemi informacijami za nadziranje in urejanje uporabnikov in njihovih pravic. Vsi servisi se nahajajo na istem naslovu – strežniku. Odziv je odvisen od poizvedbe, ki jo je potrebno poslati na naslov, da dobimo želene podatke. Podatki, ki se pošiljajo na strežnik in ki jih strežnik pošilja nazaj, so zakodirani z našim kriptirnim sistemom. Preden se podatki pošljejo uporabniku, so urejeni v znano podatkovno strukturo JSON [3]. Servis vrne podatke glede na zahtevo, njihova vsebina pa je odvisna tudi od uporabnikovih pravic. Glavni parameter zahteve je parameter c (content), ki servisu sporoči kaj mora storiti.

Primer:

{“c”:”access”,”access”:”0”,”usr”:”Uporabnik1”}

- servis preko prvega parametra prepozna, da gre za urejanje dostopa uporabnika,

- drugi parameter določa dostop, ki ga bo dobil uporabnik (1 = odklenjen , 0 = zaklenjen),

- tretji parametra poda uporabnika, ki bo dobil prej omenjen dostop.

5.2 Kriptirni algoritem

Najprej smo se morali odločiti ali bomo uporabili že znan ter zelo uporabljen algoritem MD5 ali bomo napisali svojega. Izziv programiranja nas je navdihnil, da sestavimo svoj algoritem. Pričeli smo z zbiranjem idej, kako bi ta problem rešili, saj je bil izziv kar velik. Med zapisanimi idejami smo izbrali kriptirni algoritem, kjer bo vsak znak (bajt) predstavljen z osmimi znaki, ki so zapisani kot niče in enice (npr. a: 00001100). Tako smo vsakemu znaku priredili njegovo unikatno zaporedje. S tem sistemom smo zagotovili, da lahko pretvorimo kateri koli znak in da je postopek izvedljiv v obe smeri (kriptiranje in dekriptiranje). Za postopek kriptiranja smo pripravili dve polji. V prvega smo shranili vso slovensko abecedo in

dodatne znake, v drugega pa za vsak znak iz prejšnjega polja ekvivalenten kriptiran zapis z osmimi znaki.

Kasneje, v toku izdelave uporabniške aplikacije, smo se odločili, da bomo poleg našega sistema za kriptiranje gesel uporabili tudi MD5 kriptiranje.

5.2.1 Kriptiranje

Kriptiranje bomo uporabili za varen prenos ukazov in spremljajočih parametrov (besedilo), ki ne smejo biti javno izpostavljeni. Sam postopek kriptiranja je zelo enostaven, saj mora poiskati le ujemajoči se par. Ta problem smo rešili z dvojno zanko. Zanka je del programa, ki omogoča ponavljanje enega ali več stavkov. Medtem ko zunanja zanka skrbi, da se izvede točno tolikokrat, kolikor je znakov v besedilu, skrbi notranja za primerjanje (iskanje) znakov. Ko najde ujemajoči znak, se njemu pripadajoči kriptirani znaki shranijo v niz s kriptiranimi podatki. Postopek se ponavlja v skladu z zunanjo zanko. Končni rezultat je kriptirano zaporedje enic in ničel kot smo določili v našem kriptirnem polju.

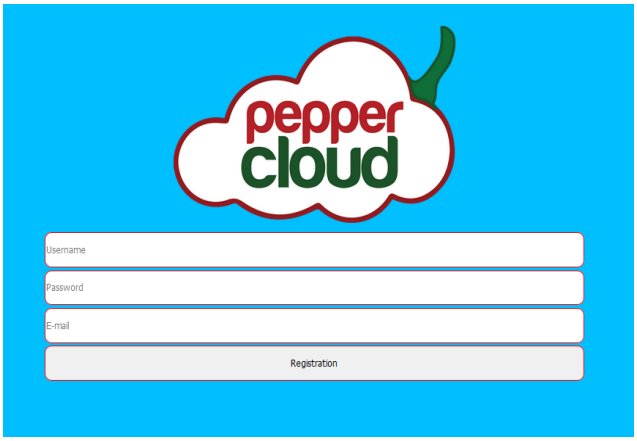
Kot primer smo zakodirali besedo »raziskovalna«:
0010111100001011010000100011101001100010010000100101001001110010000101100100011001000111100001011

5.2.2 Dekriptiranje

Postopek dekriptiranja je bil malo težji, a smo nalogo kljub temu uspešno opravili. Izvedli smo ga s tremi zankami ter dvema spremenljivkama. Prva zanka ima enako nalogo kot pri kriptiranju – nadzor izvajanja tolikokrat, kolikor je nizov po osem znakov (vsak znak je bil kriptiran z osmimi ničlami in enicami). Druga zanka je zadolžena, da združuje nize po osem znakov (npr. najprej vzame prvih osem znakov) in jih shrani v spremenljivko. Tretja zanka preveri, kateremu znaku v abecedi pripada ta niz znakov. Ko najde ujemajoči znak, se le-ta shrani v niz podatkov, ki se v servisu ustrezno obdelajo, pri uporabniku pa prikažejo.

6 Izdelava osnovne spletne strani za prijavo

Pri pripravi prve strani smo pomoč iskali na pogosto uporabljenih straneh, saj smo si tako izoblikovali idejo, kako mora izgledati prva stran, da bi privabila uporabnike. Na koncu pregleda vseh strani, smo se odločili za »tumbrl« [4], saj je stran zelo privlačna in dinamična. Seveda smo morali veliko stvari prilagoditi svojim potrebam. Sprva smo izoblikovali prijavno okno (»login«) [2]. To smo izvedli s HTML-jem, CSS-jem, JavaScript-om in php-jem. Nato smo morali vključiti ime in logotip našega podatkovnega oblaka. Odločili smo se, da bomo logotip postavili na zgornji del spletne strani. Potem smo se lotili dinamičnosti strani. Brskali smo po spletu, si ogledovali/brali različne forume ter si pomagali z lastnim znanjem. Sprva smo imeli veliko problemov, saj se koda ni pravilno izvajala. Z vztrajnostjo smo težave odpravili. Koncept dinamičnosti je bil takšen, da bi lahko uporabnik s preprostim klikom na »gumbek« skočil na stran za registracijo. Prijavno okno smo nato še obogatili s CSS-jem, da je stran še bolj privlačna za uporabnike.



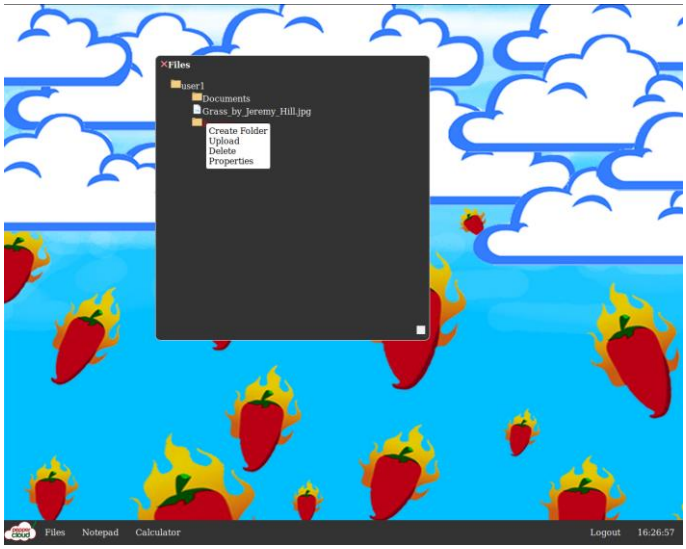
Slika 3 Primer okna za registracijo

7 Izdelava uporabniških vmesnikov

7.1 Spletni vmesnik za namizne računalnike

Večina uporabnikov sodobnih računalnikov in mobilnih naprav je navajena uporabljati »okna«, s katerimi se srečujemo vsak dan na sodobnih operacijskih sistemih. Zaradi intuitivnosti aplikacije tudi mi nismo naredili izjeme. Odločili smo se, da uporabniku naredimo okolje, kjer se bo počutil domače in posledično lažje in z večjim veseljem uporabljal naš oblak. Ko se prijavimo v oblak, nas pričaka namizje, ki nam ponuja številne funkcije, ki so dostopne preko ikon [5].

Najpomembnejše pa je hranjenje podatkov. Podatki se na strežnik prenašajo preko POST metode [1] in sicer z AJAX tehnologijo. To pomeni, da lahko vzporedno z nalaganjem počnemo tudi druge stvari kot so odpiranje datotek, brisanje, urejanje ... Ker je naš oblak namenjen širši javnosti in posledično temu več uporabnikom, je seveda eden ključnih pomenov pošiljanje datotek med njimi. Izmenjava informacij je v današnjem času zelo pomembna.



Slika 4 Uporabniški vmesnik

Naš oblak ni namenjen samo hranjenju in urejanju datotek. Zato smo se odločili, da ga nadgradimo z različnimi programi kot so:

- Beležka - v njo si lahko shranimo določeno besedilo oz. zapiske,
- Računalo,
- MP4/MP3 predvajalnik,
- prikazovalnik slik,
- ...

Vsak od teh programov omogoča uporabniku, da lahko skoraj vse kar je potrebno za delo, opravlja neposredno na oblaku. Tako ima vedno vse na dosegu, če tudi je njegov računalnik doma.

7.2 Spletna aplikacija za mobilne telefone

Zavedamo se, da živimo v času, v katerem želimo vse podatke takoj. Zato smo se odločili narediti tudi mobilno verzijo oblaka, ki je prilagojena tem napravam. Pomembno je, da lahko uporabniki v katerem koli trenutku dostopajo do svojih podatkov z različnimi napravami in pri tem delajo z enako uporabniško izkušnjo. To je pomembno, da se uporabniki enako znajdejo in delajo z različnimi napravami, pri tem pa za prilagajanje uporabijo kar čim manj časa. Ker na mobilnih napravah ni toliko prostora za »okna« smo se odločili, da bomo izpostavili samo glavno storitev našega oblaka, to je hranjenje, prenašanje in odpiranje datotek. To smo izvedli z preprostim seznamom datotek, brez zapletenega izgleda.

7.3 Administracijski program

Med izdelavo podatkovnega oblaka smo presodili, da bo potreba po administracijskem programu zelo velika, saj nam bo zelo olajšal delo z uporabniki. Namesto spletne strani smo se odločili, da bomo naredili aplikacijo za operacijski sistem Windows, kot programski jezik pa smo izbrali C#. Program je sestavljen iz treh oken. Pri izdelavi te aplikacije je bilo zelo pomembno, da je vse ostalo delovalo pravilno, saj je bilo delo zelo vezano na pravilno delovanje spletne aplikacije. Pri tem smo tudi opazili in takoj odpravili nekaj napak, ki smo jih naredili pri izdelavi servisov. Pri oblikovanju programa smo se odločili, da bodo okna zelo enostavna, saj s tem zagotavljamo večjo uporabnost.

7.3.1 Prijavno okno

Prijavno okno služi le prijavi v svoj račun. Kot dodatno varnost smo zagotovili, da se lahko prijavijo le uporabniki s pravicami skrbnika. Prav tako program tudi preverja, da vpisana polja

niso prazna ter da sta uporabniško ime in geslo pravilna. V primeru, da ni internetne povezave, program to pravočasno zazna in o tem uporabnika tudi opozori.

7.3.2 Glavno okno

Glavno okno naj bi bil center, iz katerega lahko upravljaš vse račune uporabnikov. Tukaj lahko na zelo enostaven in hiter način izveš ključne stvari o uporabniškem računu, kot so zasedenost prostora, email, uporabniško ime, logo (slika) računa. Omogoča pa tudi slednje:

- aktivacija/deaktivacija računa,
- izbris računa,
- ponastavitev gesla računa.

Vgrajeno ima tudi bližnjico, ki odpre okno za kreiranje uporabnikov. Program pa uporablja sprotno osveževanje, saj lahko le tako zagotovi, da ne gledamo starih podatkov. Po potrebi je mogoče vedno osvežiti podatke z le enim klikom na gumb.

7.3.3 Registracijsko okno

Registracijsko okno je namenjeno kreiranju novih uporabnikov. Ob kreiranju uporabniškega računa preveri, ali so vse potrebne vsebine v vnosnih poljih popolnjene in ali uporabnik s tem uporabniškim imenom/emailom že obstaja. V primeru, da je zadoščeno vsem pogojem, se uporabniški račun kreira, sicer program opozori administratorja o napaki.

8 Testiranje in odpravljanje napak

Po opravljenem programiranju je seveda sledilo še testiranje. Za testiranje smo naš sistem uporabljali več dni, tako smo poskusili odkriti in odpraviti čim več napak. S testiranjem smo odkrili veliko manj napak kot smo jih pričakovali. Vse najdene napake smo tudi čim hitreje odpravili. Menimo, da smo naš sistem izpopolnili, a še vedno se lahko najde kakšna napaka, ki smo jo spregledali.

9 ZAKLJUČEK

9.1 Hipoteze

Naši hipotezi sta se glasili:

- Dostop do podatkov je mogoč iz vseh okolij (telefon/računalnik).
- Dosegljivost sistema naj bi bila 99 %.

V naši raziskovalni nalogi smo omogočili dostop do podatkovnega oblaka z vseh okolij. Oblak deluje z računalniki, ki imajo naložene različne operacijske sisteme, enako velja tudi za tablične računalnike kot tudi pametne telefone. S tem je prva hipoteza potrjena.

Med izdelavo in testiranjem aplikacije smo dodelili dostope še nekaj prijateljem in sošolcem, ki so aplikacijo testirali ob različnih časih. V toku izdelave raziskovalne naloge nismo dobili nobenega sporočila o nedosegljivosti oblaka, zato predvidevamo, da je vila dosegljivost spletne strani je bila ves čas 100 %. S tem potrjujemo tudi drugo hipotezo. Za natančnejšo analizo bi morali dodeliti še več uporabniških računov in opraviti bolj natančne »meritve« o dosegljivosti sistema.

9.2 Težave pri delu

V procesu izdelovanja smo se soočili z mnogimi težavami ter izzivi. S pomočjo spleta, posvetovanja s profesorji ter našim znanjem iz računalništva smo rešili enega za drugim. Največ težav smo imeli s komunikacijo med strežnikom in administracijskim programom, saj smo se s tem prvič soočili. Z vztrajnostjo ter trdim delom nam je kljub vsem težavam uspelo narediti delujoč in varen podatkovni oblak.

9.3 Kaj smo se naučili novega

Med izdelavo smo pridobili veliko novega znanja z različnih področij. Naučili smo se, kako vzpostaviti varno povezavo med uporabniškim vmesnikom ter strežnikom, pridobili smo tudi

veliko znanja v programskem jeziku C#. Vso novo znanje nam bo pomagalo v našem nadaljnjem izobraževanju, ter tudi kasneje v življenju in službi.

9.4 Prihodnost

V prihodnosti želimo razširiti našo idejo, da bo lažje dostopna. Razmišljali smo tudi o vključevanju (»intergriranju«) našega podatkovnega oblaga v druge storitve. Prav tako pa razmišljamo, da bi podatkovni oblak prilagodili potrebam šole in bi lahko bil uporabljen kot spletna učilnica.
Želimo si, da bi ta projekt postal večji, da bi skozi leta pridobival vedno več uporabnikov. Naredili smo tudi že načrte za večjezično podporo, saj menimo, da je to ključni del, če želimo podatkovni oblak nuditi tudi v drugih državah. Ampak dejstvo je, da smo avtorji v zaključnem letniku naše srednje šole in da se bo le težko našel čas za nadgradnjo oblaka.

10 VIRI IN LITERATURA

1. Post metoda. [Online]. Stack OverFlow. [Citirano: 15. 1. 2017]. Dostopno na spletnem naslovu: http://stackoverflow.com/questions/4088625/net-simplest-way-to-send-post-with-data-and-read-response

2. Prijava v oblak. [Online]. The code player. [Citirano: 10. 1. 2017]. Dostopno na spletnem naslovu: http://thecodeplayer.com/walkthrough/make-a-simple-cloud-in-css3

3. Deserializacija Jsona. [Online]. Stack OverFlow. [Citirano: 23. 1. 2017]. Dostopno na spletnem naslovu: http://stackoverflow.com/questions/34043384/easiest-way-to-parse-json-response

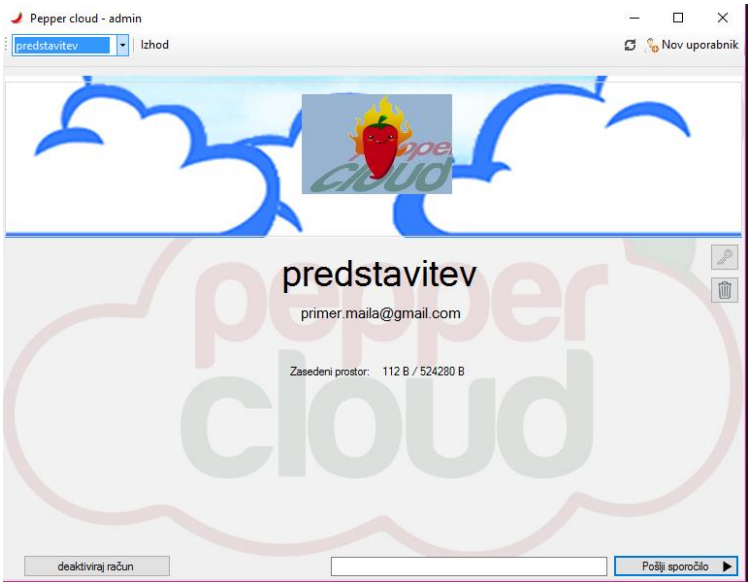
4. Osnovni model spletne strani. [Online]. Tumblr. [Citirano: 1. 2. 2017]. Dostopno na spletnem naslovu: https://www.tumblr.com/

5. Ikone. [Online]. Flat icon. [Citirano: 4. 2. 2017]. Dostopno na spletnem naslovu: http://www.flaticon.com/

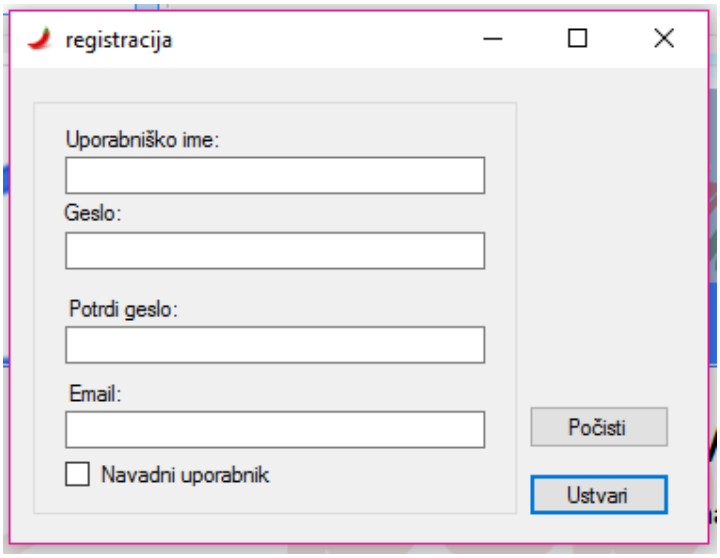
11 Priloge

Priloga 1: Slike administracijskega programa.
Priloga 2: Sistem za kriptiranje – navadna verzija.
Priloga 3: Slike spletne strani.
Priloga 4: Koda podatkovne baze.

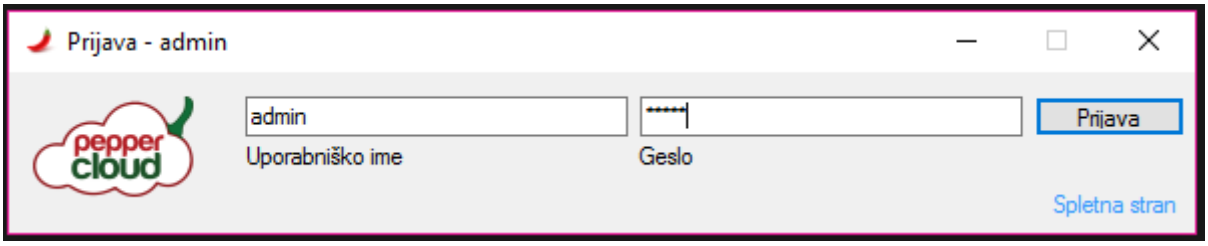
Priloga 1:



Slika 5 glavno okno



Slika 6 okno za registracijo



Slika 7 okno za prijavo

Priloga 2: Sistem za kriptiranje

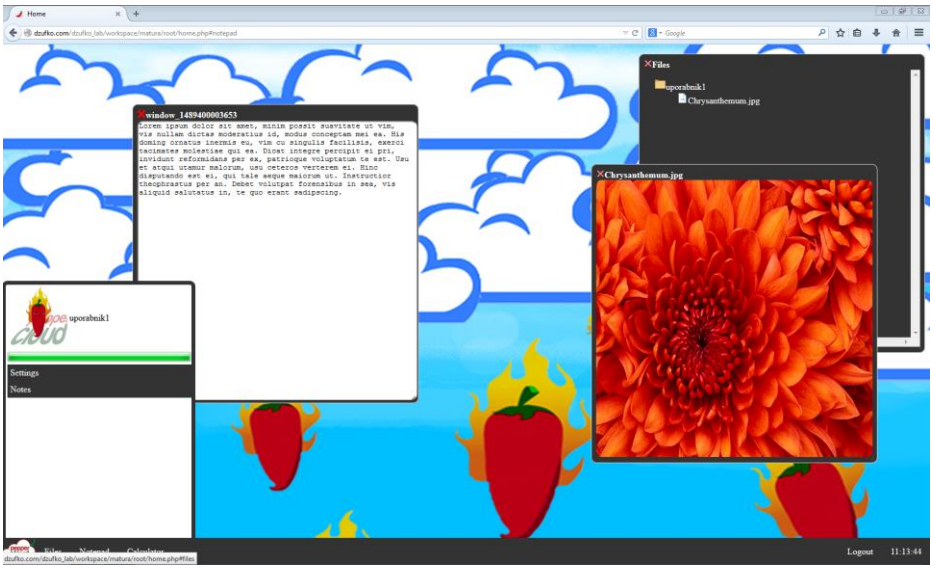
```
namespace kriptiranje
{
   public static class cript
   {

      #region abecedi
      private static char[] abeceda = new char[99] { ' ','0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'a', 'B', 'b', 'C', 'c', 'Č', 'č', 'D', 'd',
'E', 'e', 'F', 'f', 'G', 'g', 'H', 'h', 'I', 'i', 'J', 'j', 'K', 'k', 'L', 'l', 'M', 'm', 'N', 'n', 'O', 'o', 'P', 'Q', 'q', 'p', 'R', 'r', 'S', 's', 'Š', 'š', 'T', 't', 'U', 'u',
'V', 'v', 'W', 'w', 'X', 'x', 'Y', 'y', 'Z', 'z', 'Ž', 'ž', ',', '.', '-', '-', '@', '{', '}', '<', '>', '/', '(', ')', '&', '%', '#', '$', '€', '|', '\\', '~', '^', '[', ']', '?',
'=', '!', "'", '"', '\', '+' };
      private static string[] znaki = new string[99] { "11100010", "10111010", "01100100", "11110111", "11000100",
"00000100", "00000101", "00000110", "00000111", "00001000", "00001001", "00001010", "00001011", "00001100",
"00001101", "00001110", "00001111", "00010000", "00010001", "00010010", "00010011", "00010100", "00010101",
"00010110", "00010111", "00011000", "00011001", "00011010", "00011011", "00011100", "00011101", "00011110",
"00011111", "00100000", "00100001", "00100010", "00100011", "00100100", "00100101", "00100110", "00100111",
"00101000", "00101001", "00101010", "00101011", "00101100", "00101101", "00101110", "00101111", "00110000",
"00110001", "00110010", "00110011", "00110100", "00110101", "00110110", "00110111", "00111000", "00111001",
"00111010", "00111011", "00111100", "00111101", "00111110", "00111111", "01000000", "01000001", "01000010",
"01000011", "01000100", "01000101", "01000110", "01000111", "01001000", "01001001", "01001010", "01001011",
"01001100", "01001101", "01001110", "01001111", "01010000", "01010001", "01010010", "01010011", "01010100",
"01010101", "01010110", "01010111", "01011000", "01011001", "01011010", "01011011", "01011100", "01011101",
"01011110", "01011111", "01100000", "01100001"};
      #endregion
      public static string encrypt(string vnos1)
      {
         char[] vnos;
         vnos = vnos1.ToCharArray();
         string izpis = "";
         for(int a = 0; a < vnos.Length; a++)
         {
            for(int b = 0; b < abeceda.Length; b++)
            {
               if(vnos[a] == abeceda[b])
               {
                  izpis = izpis + znaki[b];
                  break;
               }
            }
         }
         return izpis;
      }
```

```
    public static string decrypt(string vnos1)
    {
        char[] vnos;
        vnos = vnos1.ToCharArray();
        string izpis = "", acs = "";

        for(int a = 0; a < vnos.Length; a = a + 8)
        {
            acs = "";
            for(int c = a; c < a + 8; c++)
            {
                acs = acs + vnos[c];
            }
            for(int b = 0; b < znaki.Length; b++)
            {
                if(acs == znaki[b])
                {
                    izpis = izpis + abeceda[b];
                    break;
                }
            }
        }
        return izpis;
    }
  }
}
```

Priloga 3: slika spletne strani



Priloga 4: koda podatkovne baze

```
Create Database oblak
Create table uporabniki
(
u_id int NOT NULL AUTO_INCREMENT(1,1),
uporabnisko varchar(255) CHARACTER SET utf8,
geslo varchar(255) CHARACTER SET utf8,
PRIMARY KEY(u_id)
)
CREATE TABLE datoteke
(
d_id int NOT NULL AUTO_INCREMENT(1,1),
pot varchar(255) CHARACTER SET utf8,
ime varchar(255) CHARACTER SET utf8,
tip varchar(255) CHARACTER SET utf8,
PRIMARY KEY(d_id)
)
CREATE TABLE uporabnik_datoteka
(
u_id int FOREIGN KEY REFERENCES uporabniki(u_id),
d_id int FOREIGN KEY REFERENCES datoteke(d_id),
)
CREATE TABLE vrsta
(
d_id int FOREIGN KEY REFERENCES datoteke(d_id),
tip varchar(20),
ikona varchar(255),
PRIMARY KEY(tip)
)
```