Mrežno upravljanje znanjem

Gredelj, Matea

Master's thesis / Diplomski rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:235:267367

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**

Repository / Repozitorij:

Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Matea Gredelj

Zagreb, 2019.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Mentor:

Doc. dr. sc. Vladimir Milić, mag. ing.

Student:

Matea Gredelj

Zagreb, 2019.

Izjavljujem da sam ovaj rad izradila samostalno koristeći znanja stečena tijekom studija i navedenu literaturu.

Zahvaljujem se svom mentoru doc. dr. sc. Vladimiru Miliću, mag. ing. što mi je omogućio izradu završnog rada, te odvojio svoje vrijeme kako bi korisnim savjetima i stručnom pomoći olakšao njegovu izradu.

Također se zahvaljujem bivšem mentoru prof. dr. sc. Mariu Essertu dipl. ing. na potpori, pomoći i suradnji za vrijeme mog studiranja, kako u poslovnom pogledu, tako i u privatnom. Uz njega zahvaljujem se i docentu dr. sc. Tihomiru Žiliću dipl. ing., te asistentu Juraju Beniću mag. ing. na ukazanoj pomoći tijekom izrade ovog rada, kao i za vrijeme studiranja.

Na kraju, posebno se zahvaljujem svojim roditeljima Damiru i Jasenki, sestri Miji i šogoru Marku na pruženoj potpori i ljubavi tijekom studiranja i izradi ovog rada. Hvala i mojim prijateljicama Martini, Mirni i Petri što su mi pružale potporu i bile uvijek tu za mene. I na samom kraju, veliko hvala mome malenom nećaku Maksimu na inspiraciji za ovaj diplomski rad.

Tata, ovo je za tebe.

Matea Gredelj

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE
Središnje povjerenstvo za završne i diplomske ispite
Povjerenstvo za diplomske radove studija strojarstva za smjerove:
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment, inženjerstvo materijala te mehatronika i robotika

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum:	Prilog:
Klasa:	
Ur. broj:	

DIPLOMSKI ZADATAK

Student: **MATEA GREDELJ** Mat. br.: 0035195966

Naslov rada na hrvatskom jeziku: **MREŽNO UPRAVLJANJE ZNANJEM**

Naslov rada na engleskom jeziku: **NETWORK KNOWLEDGE MANAGEMENT**

Opis zadatka:

Učenje i provjera znanja uz pomoć računala u žarištu su pedagoškog i znanstvenog interesa. Mrežna povezanost računala briše granice škole/fakulteta/ustanove i kuće/studentskog doma učenika ili studenta. Cilj ovog rada je načiniti mrežnu aplikaciju koja će omogućiti postavljanje digitaliziranog sadržaja različitih kolegija, dohvaćanje i učenje tog sadržaja, te na koncu provjeru znanja. Sadržaj kojim se pripravlja mrežno gradivo uključivat će različite formate i multimedijske tehnike, npr. simulacije, geografske karte, video zapise i slično.

Rad treba obuhvatiti sljedeće korake:
1. Objasniti proces upravljanja znanjem, projektirati računalni model i njegovu realizaciju u web2py mrežnom okviru.
2. Projektirati računalnu bazu koja će pohranjivati znanje i zadatke za provjeru učenika/studenta.
3. Projektirati i realizirati autorizirani sustav: prijave (engl. login) nastavnika i učenika/studenata.
4. Načiniti mrežno sučelje u HTML5/CSS3/jQuery/Bootstrap tehnologiji za obje velike cjeline aplikacije: učenja i provjere znanja.
5. Odabrati i realizirati dva kolegija (za učenika i studenta) sa znanjem i provjerom znanja kroz mrežno sučelje.
6. Usporediti klasičan i moderan tip učenja.
7. Zaključak.

U radu je potrebno navesti korištenu literaturu i eventualno dobivenu pomoć.

Zadatak zadan:
26. rujna 2019.

Rok predaje rada:
28. studenog 2019.

Predviđeni datum obrane:
04. prosinca 2019.
05. prosinca 2019.
06. prosinca 2019.

Zadatak zadao:
V. Milić
doc. dr. sc. Vladimir Milić

Predsjednica Povjerenstva:
prof. dr. sc. Biserka Runje

SADRŽAJ

SADRŽAJ I
POPIS SLIKA III
POPIS TABLICA V
POPIS PROGRAMSKIH KODOVA VI
POPIS OZNAKA I KRATICA VII
SAŽETAK VIII
SUMMARY IX
1. UVOD 1
2. KLASIČAN I MODERAN NAČIN UČENJA 3
3. APLIKACIJA web2py 4
3.1. Pokretanje web2py aplikacije 5
3.2. Model MVC 11
4. TEHNOLOGIJE ZA IZRADU I UREĐENJE SUČELJA 13
4.1. Prezentacijski jezik za izradu web stranica - HTML5 13
4.1.1. HTML5 u aplikaciji 14
4.2. Stilski jezik - CSS3 16
4.2.1. Prikaz boja u CSS-u 17
4.3. Skriptni programski jezik - JavaScript 18
4.4. Biblioteka JavaScript-a: jQuery 19
4.5. Alat otvorenog koda za razvoj HTML-a, CSS-a i JavaScripta-a: Bootstrap 19
5. IZRADA APLIKACIJE ZA MREŽNO UPRAVLJANJE ZNANJEM 20
6. APLIKACIJA „Sveznalica“ 22
6.1. Tema aplikacije 22
6.2. Sustav prijave 25
6.3. Početna stranica 27
6.4. Stranica „7. razred“ 32
6.5. Stranica „Ups…“ 33
6.6. Predmeti za osnovnu i srednju školu 34
6.7. Multimedijski dodaci 37
6.8. Stranica „Periodni sustav“ 37
6.9. Stranica „Kalkulator“ 38
6.10. Stranica „Računanje kvadratne jednadžbe“ 39
6.11. Stranica „Europa u brojevims“ 41
6.12. Stranica „Karta“ 44
6.13. Realizacija kolegija sa znanjem i provjerom znanja 45
6.13.1. Baze podataka 46
6.13.2. Znanje: predavanja i vježbe 49
6.13.3. Provjera znanja - kviz 52
6.13.4. Stranica „Videoteka“ 55
6.13.5. Znanje i profesori 55

7. ZAKLJUČAK 59
LITERATURA 60
PRILOZI 61

POPIS SLIKA

Slika 1. Logo aplikacije web2py 4
Slika 2. web2py sučelje za pokretanje aplikacije 5
Slika 3. Početna stranica web2py aplikacije u web pregledniku 6
Slika 4. Prijava na administrativno sučelje 6
Slika 5. Stranica sa web2py aplikacijama 7
Slika 6. Stvaranje nove aplikacije 8
Slika 7. Kategorija „models“ web2py aplikacije 8
Slika 8. Kategorija „controllers“ web2py aplikacije 9
Slika 9. Kategorija „views“ web2py aplikacije 9
Slika 10. Kategorija „modules“ web2py aplikacije 9
Slika 11. Kategorija „private files“ web2py aplikacije 9
Slika 12. Kategorija „languages“ web2py aplikacije 10
Slika 13. Kategorija „static“ web2py aplikacije 10
Slika 14. Kategorija „plugins“ web2py aplikacije 11
Slika 15. MVC obrazac 11
Slika 16. Prikaz HTML5 loga 13
Slika 17. Prikaz CSS3 loga 16
Slika 18. JavaScript logo 18
Slika 19. jQuery logo 19
Slika 20. Bootstrap logo 19
Slika 21. Izgled aplikacije u obliku datoteka na računalu 21
Slika 22. Navigacijska traka aplikacije 23
Slika 23. Padajući izbornik aplikacije 24
Slika 24. Stilizirao podnožje aplikacije 24
Slika 25. Stranica za prijavu 27
Slika 26. Prikaz padajućeg izbornika 30
Slika 27. Izgled početne stranice aplikacije 31
Slika 28. Stranica „7. razred“ 32
Slika 29. Uređena stranica „Ups…“ 34
Slika 30. Izgled prezentacije na stranici 36
Slika 31. Izgled lista na stranici uređene Boostrapovom mrežom 36
Slika 32. Stranica „Periodni sustav“ 37
Slika 33. Prikaz animacije pri dodiru određenog elementa periodnog sustava elemenata 38
Slika 34. Izgled stranice „Kalkulator“ 39
Slika 35. Izgled stranice „Kvadratna jednadžba“ 40
Slika 36. Mogućnosti grafova canvasJS-a 41
Slika 37. Tortni grafički prikaz podataka 43
Slika 38. Stupičasti grafički prikaz podataka 44
Slika 39. Stranica „Karta“ sa prikazom Google Maps-a 45
Slika 40. Sučelje za upravljanje bazama podataka 48
Slika 41. Tablični prikaz podataka iz baze 50
Slika 42. Prikaz u slučaju prazne baze podataka 51
Slika 43. Stranica „Fakultet strojarstva i brodogradnje“ 51
Slika 44. Stranica „Elektrotehnika“ sa podacima iz baze 52
Slika 45. Prikaz kviza iz elektrotehnike na stranici 53

Slika 46. Prikaz ispunjenog kviz iz biologije sa označenim netočnim odgovorima 54
Slika 47. Stranica „Videoteka“ 55
Slika 48. Izgled kartice za unos novog poglavlja 56
Slika 49. Gumb za uređivanje objave 57
Slika 50. Stranica za uređivanje objava 57

POPIS TABLICA

Tablica 1. Popis korištenih HTML *tagova* i njihova objašnjenja [5] 14
Tablica 2. Tablica različito definiranih boja [5] 18

POPIS PROGRAMSKIH KODOVA

Programski kod 1. CSS kod teme aplikacije 22
Programski kod 2. Podnožje aplikacije 24
Programski kod 3. Logo rađene aplikacije 25
Programski kod 4. Definiranje funkcije stranice 25
Programski kod 5. Obrazac za prijavu 26
Programski kod 6. Stil okvira 26
Programski kod 7. Preusmjeravanje nakon prijave/odjave 28
Programski kod 8. Definiranje početne stranice 28
Programski kod 9. Stiliziranje padajućeg izbornika i gumba 28
Programski kod 10. Padajući izbornik 30
Programski kod 11. Povezivanje funkcije pomoću klasa 31
Programski kod 12. Link za prethodnu stranicu 32
Programski kod 13. Grafika HTML-a: stvaranje polja grafike te crtanje linije, izlomljene linije i kruga u njemu 33
Programski kod 14. Pozivanje slike na stranicu 34
Programski kod 15. Bootstrap prezentacija (eng. *carousel*) 34
Programski kod 16. Funkcija za izračun kvadratne jednadžbe (jQuery) 39
Programski kod 17. Kod za pozivanje grafova pomoću id-a 42
Programski kod 18. Uređivanje grafova 42
Programski kod 19. Pozivanje i uređivanje interaktivne karte (Google Maps) 44
Programski kod 20. Definiranje baze podataka „biologija“ i „bioquiz“ 46
Programski kod 21. Definiranje baze podataka „elektrotehnika“ i „elektrokviz“ 47
Programski kod 22. Stiliziranje tablice 49
Programski kod 23. Način dohvaćanja podataka iz baze 49
Programski kod 24. Video 55
Programski kod 25. Definiranje forme 56
Programski kod 26. Definiranje forme sa podacima za izmjenu 57

POPIS OZNAKA I KRATICA

Popis oznaka

Oznaka	Jedinica	Opis
x	/	Koordinatna os
y	/	Koordinatna os
z	/	Koordinatna os

Popis kratica

Kratica	Opis
API	Aplikacijsko programsko sučelje (eng. *Application Programming Interface*)
CMYK	Suptraktivni model miješanja bolja, skraćeno od imena boja na engleskom jeziku: tirkizna (eng. *cyan*), purpurna (eng. *magenta*), žuta (eng. *yellow*) i crna (eng. *black*)
CSS	Stilski jezik (eng. *Cascading Style Sheets*)
DAL	Računalnog programa koji omogućuje pristup podacima iz baze (eng. *Data Access Layer*)
DOM	Model objekta dokumenta (eng. *Document Object Model*)
GUI	Grafičko korisničko sučelje (eng. *Graphical User Interface*)
HEX	Heksadecimalni kodovi boja
HTML	Prezentacijski jezi za izradu web stranica, (eng. *HyperText Markup Language*)
HWB	Cilindrično-koordinatni prikaz točaka u RGB modelu, skrećeno od nijanse (eng. *Hue*), bjeline (eng. *Whiteness*) i crnila (eng. *Blackness*)
ID	Identitet (eng. *Identity Document*)
IP	Internetski protokol (eng. *Internet Protocol*)
JS	JavaScript (skriptni programski jezik)
MVC	Model-Pogled-Upravitelj obrazac (eng. *Model-View-Controller*)
NCol	Sustav prirodnih boja (eng. *The Natural Color System*)
RGB	Aditivni model boja, skraćeno od imena osnovnih boja na engleskom jeziku (eng. *Red, Green, Blue*)
SQL	Strukturni upitni programski jezik visoke razine (eng. *Structured Query Language*)
URL	Usklađeni lokator sadržaja (eng. *Uniform Resource Locator*)

SAŽETAK

Kroz ovaj rad biti će objašnjen način izrade web aplikacije pomoću web2py tehnologije: spoja Python-a i HTML jezika. Tema rada je mrežno upravljanje znanjem, te će tako stvorena aplikacija omogućiti postavljanje digitaliziranog sadržaja različitih kolegija, dohvaćanje tog sadržaja, te na koncu provjeru znanja. Korisnici aplikacije (profesori i učenici/studenti) će imati pristup različitim stranicama aplikacije – profesori će imati pristup svojim bazama podataka (koje će moći uređivati) gdje će se pohranjivati znanje, dok će učenici moći pristupiti stranicama gdje će to znanje biti prikazano, kao i stranicama na kojima će to znanje biti ispitano. Stranice će biti kreirane pomoću MVC paradigme. Pomoću modela stvorit će se baze podataka, dok će upravitelj definirati funkcije stranica. Na kraju kroz pogled će se urediti izgled web stranice pomoću HTML5 i CSS3. Također će se kroz biblioteke JavaScript-a, pomoću jQuery-a i Bootstrap-a unaprijediti stranica pomoću različitih formata i multimedijskih tehnika.

Ključne riječi: web2py, MVC, HTML5, CSS3, JavaScript, jQuery, Bootstrap, učenje, znanje.

SUMMARY

This paper will explain how to build a web application using web2py technology: a combination of Python and HTML. The topic of the thesis is network knowledge management. The application created will allow professors to set up digitized content for different courses and retrieve that content, and allow students to test their knowledge. Application users (professors and students) will have access to different pages of the application - professors will have access to their (editable) databases where knowledge will be stored, while students will be able to access pages where that knowledge will be displayed, as well as pages on which this knowledge will be tested. Pages will be created using the MVC paradigm. Models will create databases, while the controller will define page functions. At the end of this thesis the view will edit the layout of the website using HTML5 and CSS3. Also through JavaScript libraries, jQuery and Bootstrap will enhance the page using various formats and multimedia techniques.

Key words: web2py, MVC, HTML5, CSS3, JavaScript, jQuery, Bootstrap, learning, knowledge.

1. UVOD

Ubrzan razvoj interneta dovodi do sve veće težnje prebacivanja stvarnog svijeta u mrežni. Kako danas postoje povezani mrežni sustavi, na primjer u medicini kojima mogu pristupiti svi liječnici sa svojim korisničkim podacima, tako dolazi do sve više ideja prebacivanja sustava na internet. U ovom radu bit će objašnjeno stvaranje jedne takve ideje: mrežno upravljanje znanjem.

Primjer mrežnog upravljanja znanjem možemo vidjeti na web aplikaciji „ScriptRunner“ kojim se koristi dio kolegija Fakulteta strojarstva i brodogradnje. U tom konkretnom slučaju znanje se ne prenosi, već se samo provjerava (profesori postavljaju zadatke te dobivaju povratnu informaciju o uspješnosti studenata, dok studenti imaju pristup samo navedenim zadacima). Sličan primjer možemo vidjeti i u aplikaciji „Učilica“ kojom se znanje prenosi i ispituje, no nažalost nije mrežno povezano. Sama ideja aplikacije u diplomskom radu zapravo je spoj dviju prethodno navedenih aplikacija.

Pomoću web2py aplikacije stvorena je web aplikacija za profesore i učenike/studente kojom se mrežnim putem može izmjenjivati i provjeravati znanje. Aplikaciji se neće moći pristupiti ako niste registrirani korisnik te će se objasniti sam sustav prijave, kao i način na koji će korisnici po raznim kriterijima moći pristupati određenim stranicama aplikacije. Učenici/studenti koji su korisnici moći će pristupiti stranicama na aplikaciji na kojima se nalaze razni predmeti/kolegiji: predavanja, vježbe, dodaci učenju te kviz za provjeru znanja. Profesori koji su korisnici moći će pristupiti stranicama aplikacije kojima mogu pristupiti i učenici/studenti, te svojim osobnim stranicama predmeta kojeg predaju. Tu će moći pomoću baze podataka dijeliti znanje – postavljati objave (predavanja, vježbe) te provjeravati to isto znanje. To sve bit će prikazano kroz iduća poglavlja ovoga rada.

U drugom poglavlju bit će uspoređen klasičan i moderan način učenja, kao i njihove prednosti i nedostaci. Treće poglavlje opisuje samu aplikaciju web2py: nastanak, kompoziciju i način rada, a uz to će biti opisan i postupak izrade nove aplikacije. Kroz isto poglavlje bit će objašnjen i MVC (eng. *Model-View-Controller*) obrazac za odvajanje pojedinih dijelova aplikacije, ovisno o njihovoj namjeni, na kojem se sam web2py temelji. Četvrto poglavlje rada sastoji se od tehnologija za izradu i uređenje sučelja web2py aplikacije. Kroz poglavlje su objašnjena načela tehnologija, kao i opći prikaz njihovog rada sa primjerima koji su korišteni dalje tijekom izrade web aplikacije. U petom poglavlju započinje sama izrada web

stranice za upravljanje mrežnim znanjem: pozadinski podaci aplikacije na računalu, kao i izrada web stranica aplikacije. Kroz šesto poglavlje rada objašnjen je i prikazan način izrade svake zasebne web stranice aplikacije, način definiranja stila te funkcija istih. Uz prikaz znanja i način ispitivanja znanja pomoću baza podataka, prikazani su i načini interaktivnog prikaza raznih vrsta potrebnih podataka.

Iako je sam web2py moćan alat, koristimo se i HTML5, CSS3, jQuery, Bootstrap i još nekim tehnikama kako bi web aplikacija bila što zanimljivija i interaktivna za studente. Uz navedene tehnologije postoje i razne druge pomoću kojih se mogu stvarati slične ili drugačije stvari od onih prikazanih u radu. Zbog brzog razvoja tehnologije sva tiskana literatura vezana za predmetno područje vrlo brzo zastarijeva. Kod web programiranja potrebno je pratiti najnovije tehnike i trendove kroz razne portale i specijalizirane stranice. Za izradu ovog diplomskog rada, uz službenu web2py stranicu korištena je i stranica w3schools – optimizirana za učenje, testiranje i obuku web programera. Stranica se sastoji od primjera i objašnjenja svih tehnologija za web programiranje. Navedeni primjeri pojednostavljeni su u svrhu jednostavnijeg čitanja i razumijevanja. Vodiči, reference i primjeri stalno se pregledavaju i ažuriraju kako bi se izbjegle pogreške te kako bi bile u korak s razvojem weba. Najbitnija literatura za web programiranje je sam web.

Za razliku od knjiga, ova web aplikacija uz tekst ima i simulacije, interaktivne grafičke prikaze, multimedijske tehnike i slično. Kako je web2py pisan Python programerskim jezikom on se također i koristi njime, pa će biti i prikazan način implementacije Python koda u web2py aplikaciju. Ovaj rad će pokazati samo dio mogućnosti koji moćan alat poput web2py-a i općenito web programiranja nudi.

2. KLASIČAN I MODERAN NAČIN UČENJA

U današnje vrijeme kada je učenicima/studentima dostupno znanje „na dlanu“ zahvaljujući raširenosti tehnologije i interneta, klasični načini učenja postaju dosadni i naporni. Zbog navedenog se sve više pokušava modernizirati učenje: znanje i upravljanje znanjem prebaciti na internet i stvoriti e-učenje. Pod tim se pojmom smatraju svi oblici učenja pomoću računalne tehnologije. Koriste se različiti formati i multimedijske tehnike kako bi se učenike/studente što više zainteresiralo za gradivo. Svo znanje na zabavan način udaljeno je od korisnika na samo par klikova mišem. Elektroničkim učenjem se pokušava povećati motivaciju učenika/studenata, no ono nije korisno samo učenicima već i profesori profitiraju od toga. Uz povećanu motivaciju studenata, prolaznost učenika/studenata raste što dovodi do veće uspješnosti i zadovoljstva profesora, ali i učenika/studenata. Također ovaj način učenja pogodan je kako za obične učenike/studente, tako i za one koji imaju problema u učenju ili općenito nekih medicinskih problema koji ih sprječavaju u vođenju normalnog života. Uz sve navedene pozitivne strane e-učenja postoji i nekoliko negativnih, poput nedostatka komunikacije s ostalim ljudima. Nažalost, nisu svi u novčanoj mogućnosti priuštiti si potrebne opremu, no i za to postoje rješenja. Ipak, najveći problem ovog tipa učenja je nepouzdanost tehnike. Nikada ne znamo kada se našem uređaju može dogoditi kvar, ili jednostavno nestati Internet zbog nekog od mnogih mogućih razloga. Bez obzira na te nedostatke, danas je cilj modernizirati učenje kako bi se gradivo lakše i zabavnije moglo približiti učenicima/studentima i pomoći im kako sada, tako i u budućem životu.

3. APLIKACIJA web2py

Aplikacija web2py je web aplikacija otvorenog koda napisana u Python programskom jeziku. Logo aplikacije možemo vidjeti na slici 1. Stvorila ga je zajednica profesionalnih programera i profesora predvođena Massimom Di Pierrom. Prva verzija aplikacije bila je puštena 2007. godine (web2py 1.0), a posljednja verzija web2py-a (2.18.5) 2019. godine. Aplikacija omogućuje programiranje dinamičnog web sadržaja pomoću programskog jezika Python-a.

Slika 1. Logo aplikacije web2py

Aplikacija web2py je kostur (eng. *framework*) s punim stogom, što znači da sadrži sve komponente potrebne za izgradnju potpuno funkcionalne web aplikacije. Osmišljen je kako bi vodio web programere da slijede praksu softverskog inženjeringa, kao što je korištenje MVC uzorka (eng. *model-controller-view*) koji će biti objašnjen kasnije u radu. Aplikacija odvaja prikaz podatka (*model*), od prikaza podataka (*view*) te također od logike aplikacije i tijeka rada (*controller*). On pruža biblioteke koje pomažu programeru da dizajnira, implementira i testira svaki od ta tri dijela zajedno, a da ipak zajedno rade.

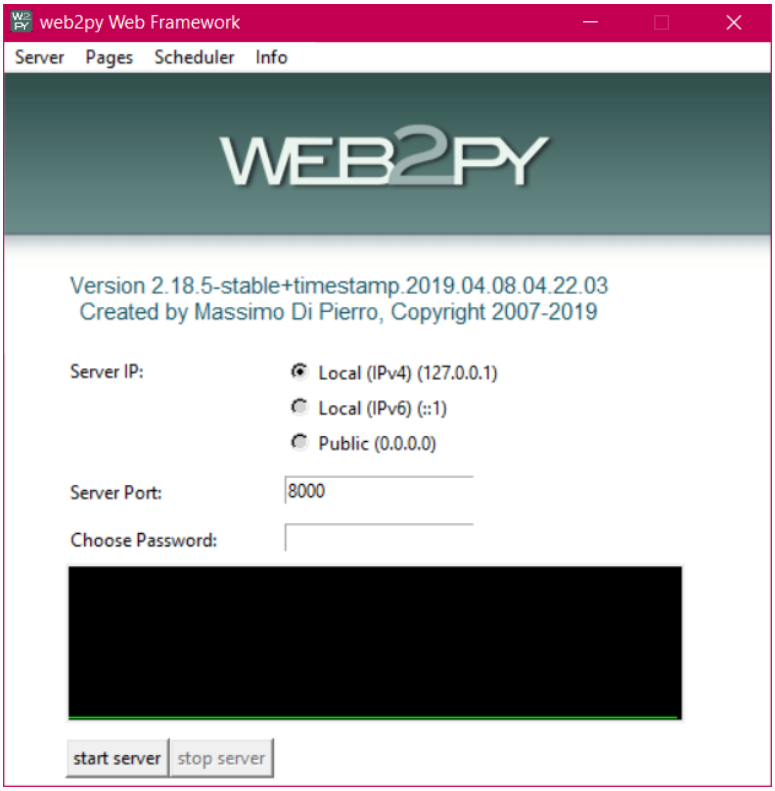
Aplikacija je izgrađena za sigurnost što znači da automatski rješava mnoge probleme koji mogu dovesti do sigurnosnih propusta. Na primjer, potvrđuje sve unose, izbjegava sve izlaze, te preimenuje prenesene datoteke. Aplikacija vodi brigu o sigurnosnim problemima, tako da programeri imaju manje izgleda za uvođenje ranjivosti.

Aplikacija web2py uključuje bazu podataka (DAL, eng. *Database Abstraction Layer*) koju dinamički piše SQL, pa isto ne mora odraditi programer. DAL zna kako generirati SQL za SQLite, MySQL, PostgreSQL, MSSQL, FireBird, Oracle, IBMDB2, Informix i Ingres. DAL također može generirati funkcijske pozive za Google Datastore prilikom pokretanja Google App Engine-a. Eksperimentalno podržava više baza podataka i stalno se dodaju nove. Jednom kada se definira jedna ili više tablica baze podataka, web2py automatski generira potpuno funkcionalno sučelje administracije baze podataka na webu za pristup bazi podataka i njenim tablicama.

Aplikacija se razlikuje od ostalih web kostura time što je to jedini kostur za potpuno prihvaćanje Web 2.0 paradigme, gdje je web računalo. Zapravo, web2py ne zahtijeva instalaciju ili konfiguraciju, te radi na bilo kojem sustavu koji može pokrenuti Python (Windows, Windows CE, Mac OS X, iOS i Unix/Linux), a faze razvoja, implementacije i održavanje aplikacija mogu se obaviti putem lokalnog ili udaljenog web sučelja. Aplikacija se pokreće sa CPython-om (implementacija C-a) i PyPy-om (Python-om pisanim u Python-u) na Python-u 2.7. [1]

3.1. Pokretanje web2py aplikacije

Da bi se započelo, potrebno je raspakirati preuzetu .zip datoteku za svoj specifični operacijski sustav (u ovom slučaju Windows), te pokrenuti web2py.exe. Prilikom pokretanja prvo se prikaže GUI *widget* vidljiv na slici 2 koji Vas zatraži da odaberete jednokratnu administratorsku lozinku, IP adresu mrežnog sučelja koje će se koristiti za web server i broj porta za posluživanje zahtjeva.



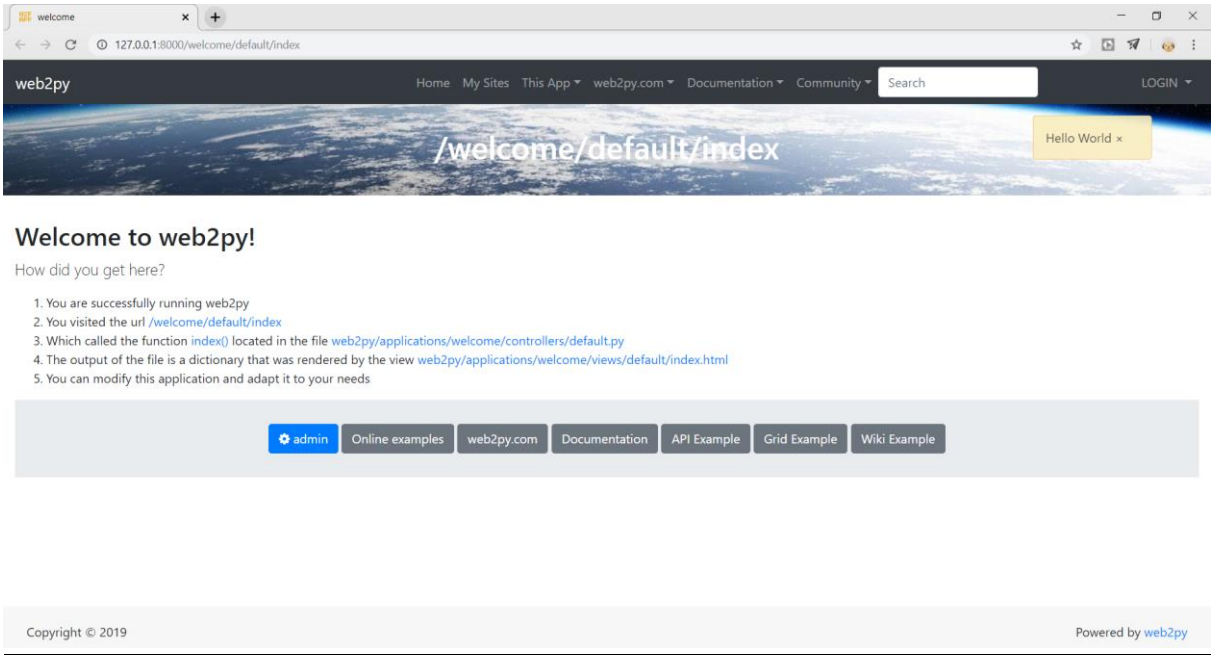
Slika 2. web2py sučelje za pokretanje aplikacije

Prema zadanim postavkama, web2py pokreće svoj web server na 127.0.0.1:8000, ali možete ga pokrenuti na bilo kojoj dostupnoj IP adresi i portu. Ako ne unesete administratorsku lozinku, sučelje za administraciju je onemogućeno. To je sigurnosna mjera za sprečavanje

javnog izlaganja administracijskog sučelja. Nakon postavljanja administratorske lozinke, web2py pokreće web preglednik stranici:

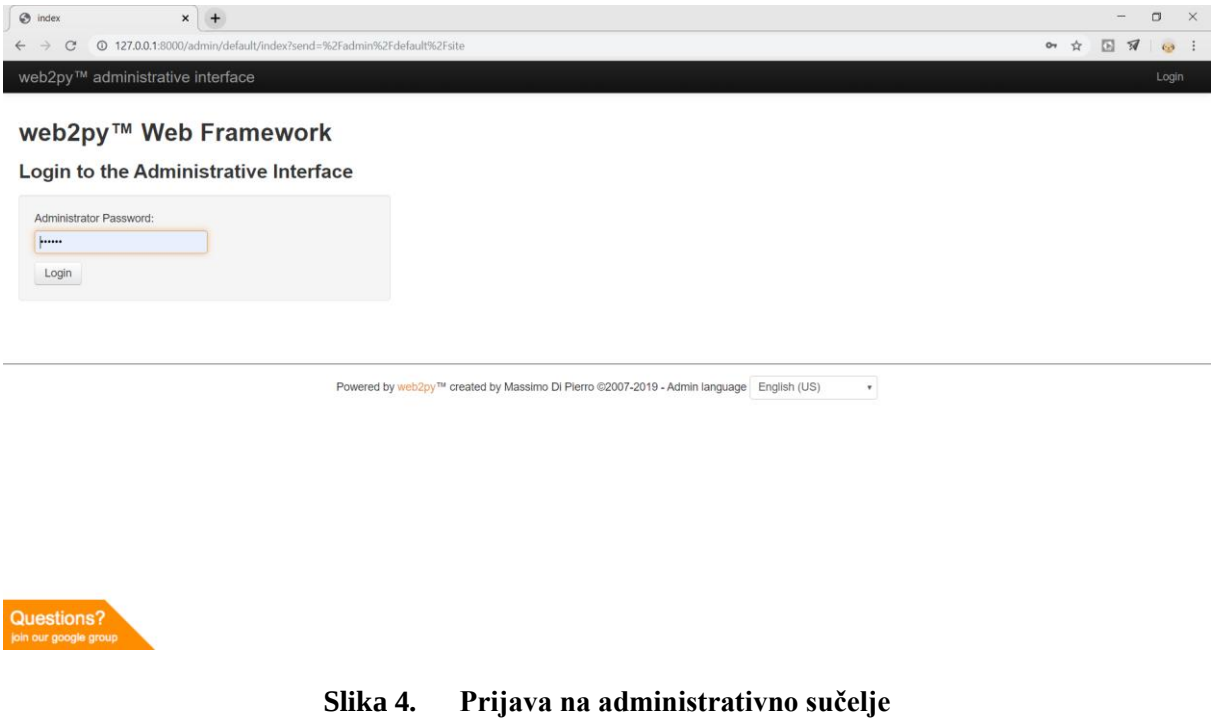
`http://127.0.0.1:8000/`,

vidljiv na slici 3. Ako računalo nema zadani preglednik, otvorite web-preglednik i unesite gore navedeni URL.



Slika 3. Početna stranica web2py aplikacije u web pregledniku

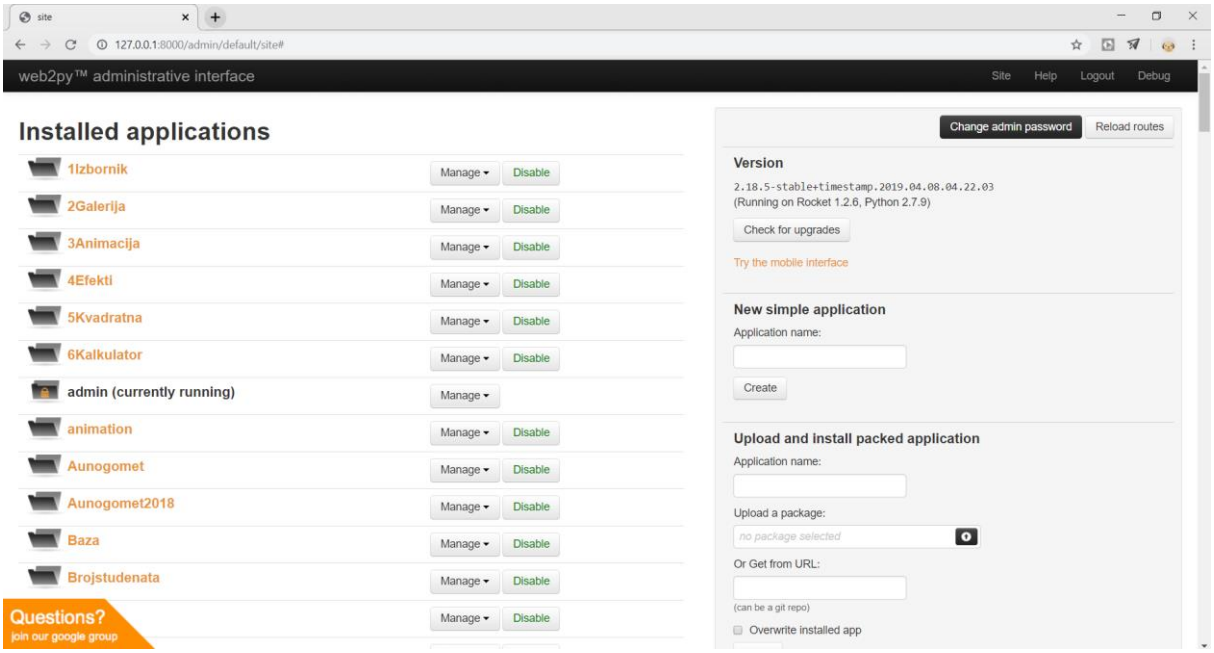
Klikom na „administrativno sučelje“ dolazi se na prijavu za administrativno sučelje vidljivo na slici 4.



Slika 4. Prijava na administrativno sučelje

Administratorska lozinka je lozinka koja je odabrana prilikom pokretanja. Kao što se može primjetiti, postoji samo jedan administrator, pa samim time postoji i samo jedna administratorska lozinka. Iz sigurnosnih se razloga od programera traži da odabere novu lozinku svaki put kada se upali web2py, no treba paziti jer ovo nije jednako mehanizmu autentifikacije u web2py aplikacijama.

Sljedeća stranica, vidljiva na slici 5, navodi sve instalirane web2py programe i omogućuje administratoru da upravlja njima. web2py dolazi s tri aplikacije: administracijska aplikacija (koju se trenutno koristi), primjer aplikacije (s online interaktivnom dokumentacijom, replika službene web2py stranice) te aplikacija dobrodošlice (osnovni predložak za bilo koju web2py aplikaciju).



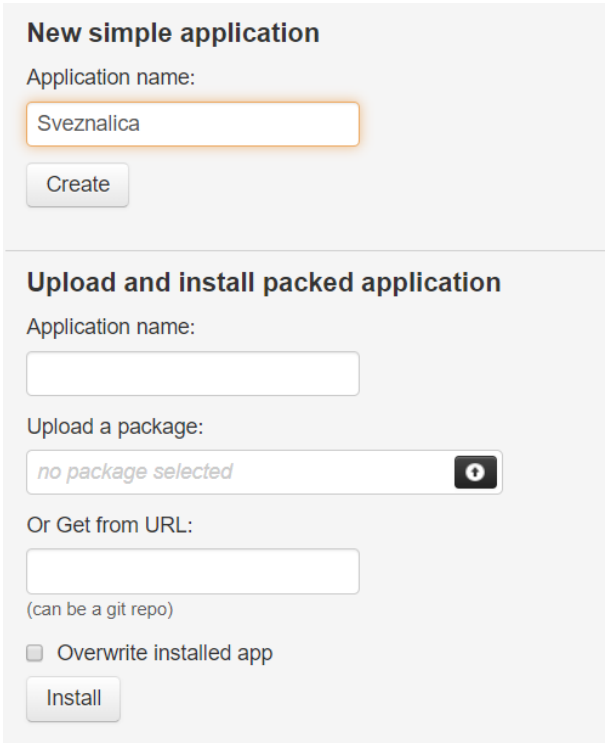
Slika 5. Stranica sa web2py aplikacijama

Na stranici administratorske aplikacije mogu se izvršiti sljedeće radnje:

- Instalirati aplikaciju ispunjavanjem obrasca u donjem desnom dijelu stranice. Imenujemo novu aplikaciju, odaberemo datoteku koja sadrži pakiranu aplikaciju ili URL gdje se program nalazi, te stisnemo '*install*'.
- Deinstalirati aplikaciju klikom na odgovarajući gumb. Također postoji stranica za potvrdu deinstalacije.
- Stvoriti novu aplikaciju odabirom novog imena i klikom na *'create'*.
- Pakirati aplikaciju za distribuciju klikom na odgovarajući gumb. Preuzeta aplikacija je tar datoteka koja sadrži sve, uključujući i bazu podataka.

- Obrisati privremene datoteke aplikacije, poput sesija, pogrešaka i datoteka predmemorije.
- Omogućiti/onemogućiti svaku aplikaciju.
- Uređivati aplikaciju [1]

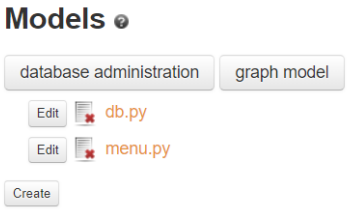
U slučaju ovog rada, stvaramo novu aplikacijom naziva „Sveznalica“ prema slici 6.



Slika 6. Stvaranje nove aplikacije

Nakon što se pritisne '*Create*', aplikacija je stvorena kao kopija ugrađene aplikacije dobrodošlice. Kako bismo je uredili, moramo stisnuti gumb '*Manage*' te u njegovom podizborniku stisnuti na gumb *'edit'*. Stranica za uređivanje nam govori što se nalazi unutar aplikacije. Svaka web2py aplikacija se sastoji od određenih datoteka, od kojih većina spada u jednu od sedam kategorija:

- „*Models*“: opisuje prikaz podataka, što je vidljivo na slici 7

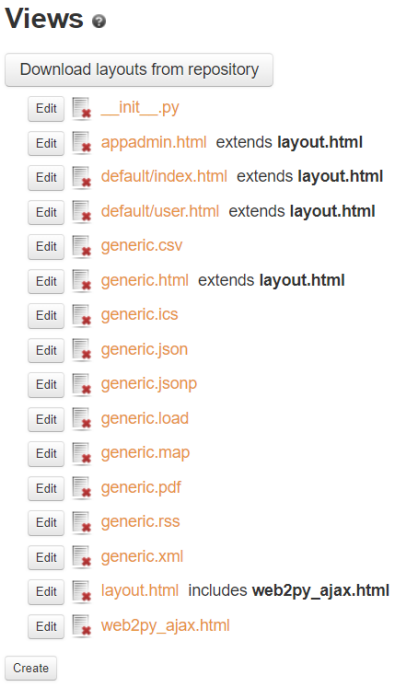


Slika 7. Kategorija „models“ web2py aplikacije

- „*Controllers*“: opisuje logiku aplikacije i tijek rada, što je vidljivo na slici 8

Slika 8. Kategorija „controllers“ web2py aplikacije

- „*View*“: opisuje prezentaciju podataka, što je vidljivo na slici 9

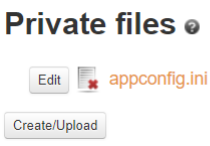


Slika 9. Kategorija „views“ web2py aplikacije

- „*Modules*“: Python moduli koji pripadaju aplikaciji vidljivi na slici 10

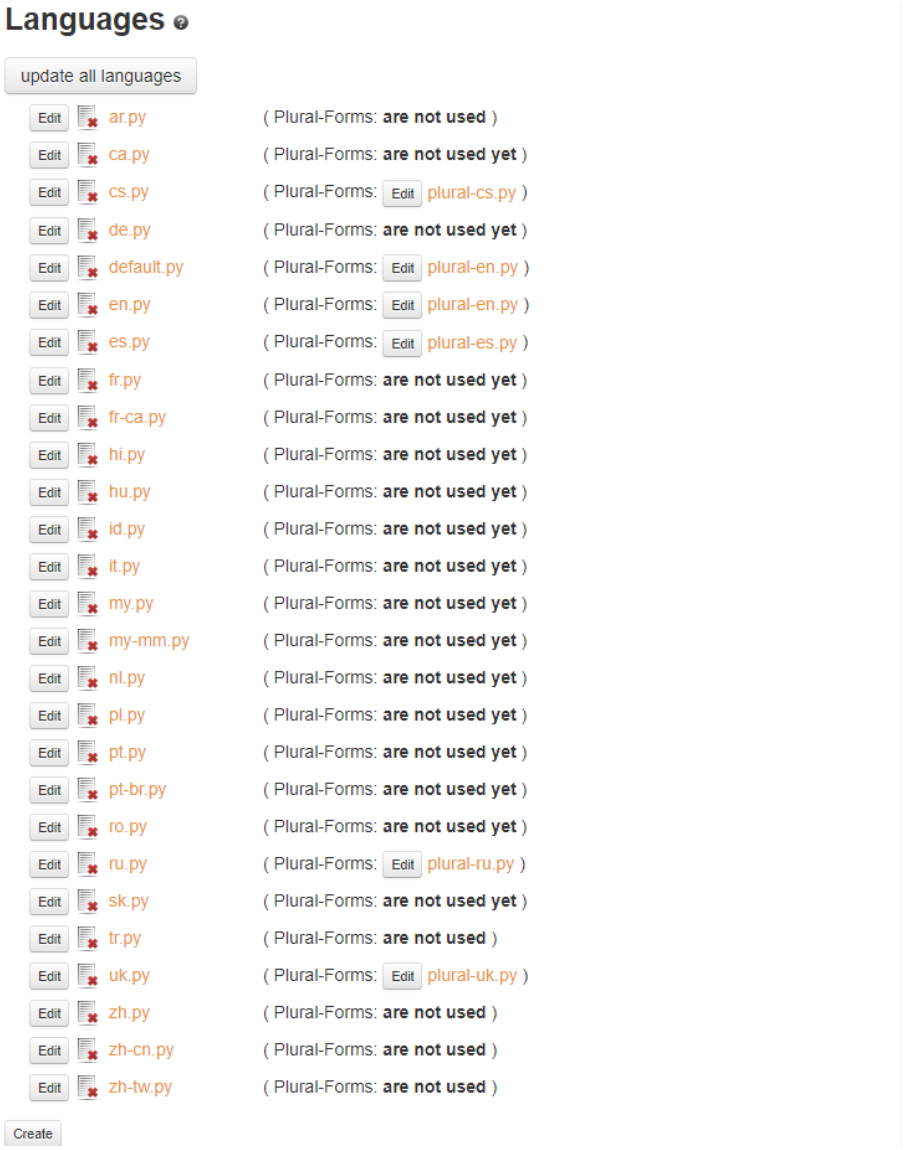
Slika 10. Kategorija „modules“ web2py aplikacije

- *„Private files“* vidljivi na slici 11



Slika 11. Kategorija „private files“ web2py aplikacije

- „*Languages*“: opisuje kako prevoditi aplikaciju na druge jezike što prikazuje slika 12

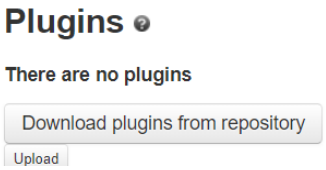


Slika 12. Kategorija „languages“ web2py aplikacije

- „*Static*“: slike, CSS datoteke, Javascript datoteke i slično, vidljive na slici 13

Slika 13. Kategorija „static“ web2py aplikacije

- „*Plugins*“: skupine datoteka dizajnirane za zajednički rad što je vidljivo na sliici 14

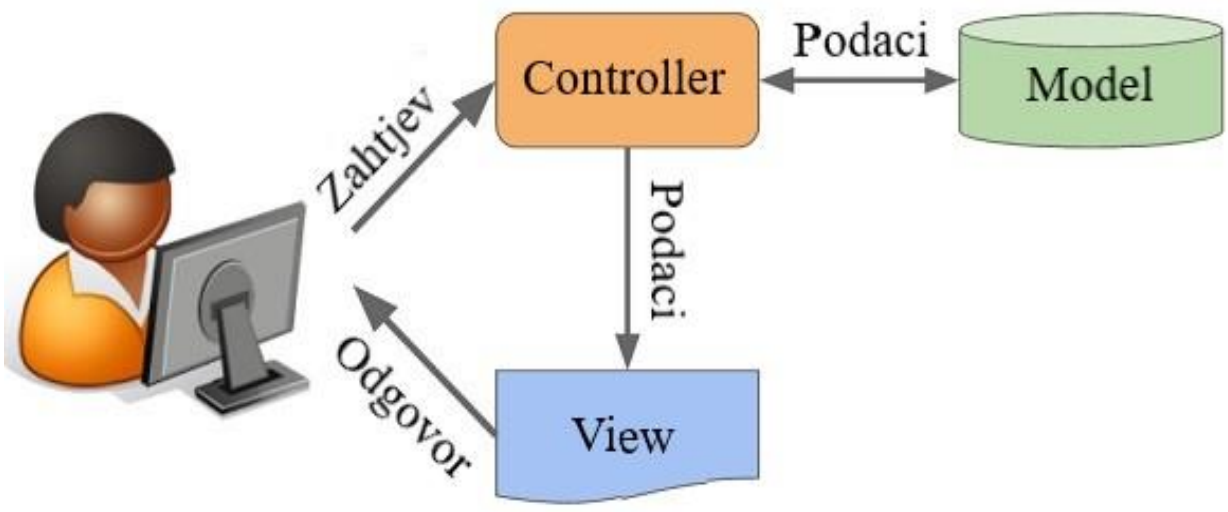


Slika 14. Kategorija „plugins“ web2py aplikacije

Sve je uredno organizirano prema modelu MVC dizajna. Svaki odjeljak na stranici za uređivanje odgovara podmapi u mapi aplikacije. Aplikacija također sadrži druge vrste datoteka (baze podataka, sesije, datoteke pogrešaka, itd.), ali nisu navedene na stranici za uređivanje jer ih administrator ne stvara ili ne mijenja, oni su stvoreni i izmijenjeni sa samom aplikacijom.

3.2. Model MVC

Model–View–Controller (MVC) je obrazac softverske arhitekture prikazan slikom 15. Koristi se u softverskom inženjeringu za odvajanje pojedinih dijelova aplikacije u komponente ovisno o njihovoj namjeni. Model se sastoji od podataka, poslovnih pravila, logike i funkcija ugrađenih u programsku logiku (eng. *business logic*). *View*, odnosno pogled je bilo kakav prikaz podataka kao što je obrazac, tablica ili dijagram. Moguć je prikaz podataka kroz više različitih pogleda. *Controller* ili upravitelj prihvaća ulazne naputke (eng. *input*) i pretvara ih u naloge modelu ili pogledu. Ovakva arhitektura olakšava nezavisan razvoj, testiranje i održavanje određene aplikacije.



Slika 15. MVC obrazac

Osim podjele aplikacije u tri vrste komponenti, *Model-View-Controller* (MVC) dizajn utvrđuje interakcije između njih:

- Upravitelj može slati naloge modelu kojima ažurira njegovo stanje (npr., uređivanje dokumenta). Također može slati naredbe u poglede kojima mijenja prikaz modela (npr., kretanje kroz dokument).
- Model dojavljuje sebi pridruženim pogledima i upraviteljima kada je došlo do promjene u njegovom stanju. Ove dojave omogućuju pogledu da prikaže obnovljeno stanje modela, a upravitelju promjenu dostupnog skup naredbi.
- Pogled zahtjeva od modela informacije potrebne za stvaranje prikaza modela korisniku. [2]

4. TEHNOLOGIJE ZA IZRADU I UREĐENJE SUČELJA

Za izradu aplikacije u web2py tehnologiji, uz već objašnjenu MVC paradigmu i Python jezik, potrebne su nam još neke tehnologije. Za uređenje sučelja potreban nam je HTML5, CSS3, JavaScript, jQuery i Bootstrap. Prije objašnjenja same upotrebe navedenih tehnologija, u idućim poglavljima će biti kratak općeniti opis tehnologija, kako bismo ih bolje upoznali prije korištenja.

4.1. Prezentacijski jezik za izradu web stranica - HTML5

Kratica HTML je za *HyperText Markup Language*, što je u prijevodu prezentacijski jezik za izradu web stranica. Hipertekst dokument stvara se pomoću HTML jezika, kojim se oblikuje sadržaj te se stvaraju hiperveze kroz tekst dokumenta. On je jednostavan za uporabu i lako se uči, što je jedan od razloga njegove opće prihvaćenosti i popularnosti. Svoju raširenost zahvaljuje jednostavnosti i tome što je od početka bio zamišljen kao besplatan i tako dostupan svima. Prikaz hipertekst dokumenta omogućuje web preglednik. Temeljna zadaća HTML jezika jest uputiti web preglednik kako prikazati hipertekst dokument. Pri tome se nastoji da taj dokument izgleda jednako bez obzira o kojemu je web pregledniku, računalu i operacijskom sustavu riječ. HTML nije programski jezik niti su ljudi koji ga koriste programeri. Njime ne možemo izvršiti nikakvu zadaću, pa čak ni najjednostavniju operaciju zbrajanja ili oduzimanja dvaju cijelih brojeva. On služi samo za opis naših hipertekstualnih dokumenata. Njegove datoteke su zapravo obične tekstualne datoteke, ekstenzija im je .html ili .htm. Osnovni građevni element svake stranice su znakovi (eng. *tags*) koji opisuju kako će se nešto prikazati u web pregledniku. Poveznice unutar HTML dokumenata povezuju dokumente u uređenu hijerarhijsku strukturu i time određuju način na koji posjetitelj doživljava sadržaj stranica. Logo HTML5 možemo vidjeti na slici 16.

Slika 16. Prikaz HTML5 loga

HTML5 je prva nova revizija standarda od HTML 4.01, koji je izdan 1999. nastao u suradnji World Wide Web Consortium (W3C) i Web Hypertext Application Technology Working Group (WHATWG). Do 2006. godine ove su dvije grupe radile odvojeno, WHATWG je radio sa web formama i aplikacijama, a W3C sa XHTML 2.0. No odlučili su udružiti snage i kreirati novu verziju HTML-a. Izdavanje konačnih specifikacija standarda HTML5 u suprotnosti je s inicijativom Web Hypertext Application Technology Working Group (WHATWG) prema kojoj bi HTML trebao biti "živi" standard koji se stalno nadograđuje, bez oznake verzije specifikacija. Zanimljivo je da već sada veliki broj preglednika ima implementiran sustav koji omogućuje interpretaciju HTML5. HTML5 donosi brojne nove mogućnosti koje HTML 4.01 i XHTML 1.x nisu imali, kao što je mogućnost reprodukcije videa na stranicama bez korištenja Adobe flasha ili Microsoftovog silverlight-a, mogućnost upravljanja pomoću tipkovnice i opcijama za bilo koju vrstu manipulacija, *'drag and drop'*, *'canvas*' kao i ostale nove elemente. [3], [4]

4.1.1. HTML5 u aplikaciji

Kao što je ranije navedeno, osnovni građevni element HTML-a su oznake, tj. *tagovi* (eng. *tag*). U tablici 1 možemo vidjeti *tagove* koji su korišteni u aplikaciji, kao i njihova objašnjenja

Tablica 1. Popis korištenih HTML *tagova* i njihova objašnjenja [5]

TAG	OPIS
<!DOCTYPE>	definira tip dokumenta
<a>	definira hiperlink
<b>	definira izraženi tekst (eng. bold)
<body>	definira tijelo dokumenta
 	definira jedan prijelaz u novu liniju
<button>	definira gumb na koji je moguće pritisnuti
<col>	određuje svojstva stupca za svaki stupac unutar <colgroup> elementa
<colgroup>	određuje grupu od jednog ili više stupaca u tablici
	definira odjeljak u dokumentu

<font>	definira font, boju i veličinu teksta
<footer>	definira podnožje za dokument ili odjeljak
<form>	definira HTML obrazac za unos korisnika
<h1> do <h6>	definira HTML naslove
<head>	definira informacije o dokumentu
<header>	definira zaglavlje za dokument ili odjeljak
<html>	definira korijen HTML dokumenta
<img>	definira sliku
<input>	definira ulaznu kontrolu
	definira stavku liste
<link>	definira poveznicu između dokumenta i vanjskog izvora
<nav>	definira veze za navigaciju
	definira uređenu listu
<p>	definira paragraf
<script>	definira skriptu (koja je od strane klijenta)
	definira odjeljak u dokumentu
<style>	definira informacije o stilu za dokument
<svg>	definira prostor za SVG grafiku
<table>	definira tablicu
<td>	definira ćeliju u tablici
<th>	definira ćeliju u tablici koja je zaglavlje
<title>	definira naslov dokumenta

<tr>	definira red u tablici
<u>	definira tekst koji bi trebao biti stilski drugačiji od ostalog teksta
	definira neuređenu listu
<video>	definira video ili film

Uz ove navedene postoji još mnoštvo *tagova* koji se koriste na razne načine, ovisno o potrebi programera i aplikacije.

4.2. Stilski jezik - CSS3

Kratica CSS stoji za „*Cascading Style Sheets*“. Radi se o stilskom jeziku koji se rabi za opis prezentacije dokumenta napisanog pomoću *markup* (HTML) jezika. Kako se web razvijao prvotno su u HTML ubacivani elementi za definiciju prezentacije (npr. tag <*font*>), ali je dovoljno brzo uočena potreba za stilskim jezikom koji će HTML osloboditi potrebe prikazivanja sadržaja (što je prvenstvena namjena HTML-a) i njegovog oblikovanja (čemu danas služi CSS). Drugim riječima, stil definira kako prikazati HTML elemente. CSS-om se uređuje sam izgled i raspored stranice. Logo CSS3-a možemo vidjeti na slici 17.

Slika 17. Prikaz CSS3 loga

Ono što je HTML5 za HTML4.1, to je CSS3 za CSS2.1 – sljedeća iteracija specifikacije koja standardizira postojeće značajke koje su donekle različito implementirane u različitim preglednicima i uvodi potpuno novi skup značajki koje CSS3 pripremaju za svijet u kojem su Web preglednici sveprisutni. Prve značajke iz specifikacije CSS3 koje su bile implementirane u preglednicima odnosile su se na prezentaciju sadržaja i temeljile su se na zaobilaznim rješenjima koja su Web dizajneri već godinama koristili: korištenje pisama iz različitih izvora, zaobljeni uglovi, sjene teksta i kutije. Zatim je uslijedio određeni broj novih selektora koji su

olakšali oblikovanje te dinamičniji efekti poput dvodimenzionalnih i trodimenzionalnih prijelaza i prijelaznih animacija. Ali, osim sjajnih vizualnih efekata CSS3 je donio i pravu malu revoluciju u obliku upita o mediju (eng. *media queries*), sintakse koja omogućava da pruže stilove pregledniku na temelju njegove veličine i mogućnosti, što je prvi korak ka istinskom oblikovanju za više uređaja.

Sljedeći važan izazov za CSS je problem rasporeda – tj. da se ponude rasporedi koji će uzimati u obzir mogućnosti korištenog agenta u kojem se prikazuju. To uključuje svojstva za dinamička korisnička sučelja i sustave moćnih mreža kojima upravlja CSS. CSS3 nije jedna specifikacija, kao što je to bio CSS2.1, gdje je sve opisano u jednom dokumentu. On je prevelik i prekompliciran za tako nešto. Umjesto toga, CSS3 je modularan – sastoji se od niza kraćih, određenijih specifikacija koje se mogu modularno implementirati u pregledniku. Kao i kod specifikacije HTML5, ideja da se čeka dok CSS3 „ne bude gotov“ prije nego što se počne koristiti je loša i smiješna, jer će neki moduli biti spremni za korištenje mnogo prije drugih. CSS modulima dodjeljuju se brojevi razina da bi se znalo kroz koliko iteracija su prošli. Neki su već na razini 4 i mogu biti temeljitije implementirani nego oni koji su još na razini 3. To naravno ne znači da ćemo jednog dana imati CSS4. To se neće dogoditi. CSS3 je kratica za „sve novije od CSS2.1“. Jednog dana razlike će nestati i sve će biti CSS. [4]

4.2.1. Prikaz boja u CSS-u

Kako je CSS odgovoran za stil, odgovoran je time i za definiranje boja pozadine, slova i slično. U idućim poglavljima prikazani su kodovi pomoću kojih je definiran stil stranice, pa je ovdje potrebno objasniti na koji način se boje mogu prikazati kako se ne bi svaki put posebno moralo objašnjavati. Postoje šest načina definiranja boje: imenom, RGB, HEX, HWB, CMYK i NCol. Ovdje su objašnjena samo prva tri načina jer su najpopularniji, a i oni su korišteni u ovome radu:

- Imenom: neke boje već imaju svoja imena, pa ih možemo tako i definirati
- RGB: boje su opisane parametrom jačine crvene, zelene i plave boje (eng. *Red, Blue, Green*) u rasponu od 0 do 225
- HEX: boje su zapisane kao heksadecimalni broj, počevši sa znakom # te sa 6 ostalih znakova, od kojih prva dva su nijansa crvene, druga dva zelene, a treća dva plave (#RRGGBB) [5]

U tablici 2 možemo vidjeti nekoliko primjera kako se za CSS može ista boja prikazati na različiti način.

Tablica 2. Tablica različito definiranih boja [5]

IME	RGB	HEX	boja
DeepPink	rgb(255, 20, 147)	#ff1493	
DeepSkyBlue	rgb(0, 191, 255)	#00bfff	
LawnGreen	rgb(124, 254, 0)	#7cfc00	
LightGrey	rgb(211, 211, 211)	#d3d3d3	
MediumAquaMarine	rgb(102, 205, 170)	#66CDAA	
OrangeRed	rgb(255, 69, 0)	#ff4500	
Gold	rgb(225, 215, 0)	#ffd700	
BlueViolet	rgb(138, 43, 226)	#812be2	
MintCream	rgb(245, 255, 250)	#f5fffa	

4.3. Skriptni programski jezik - JavaScript

JavaScript, često skraćeno kao JS, je dinamični, prototipni, multi-paradigmatski programski jezik više razine. Uz HTML i CSS, JavaScript je jedna od tri osnovne tehnologije za izradu World Wide Web sadržaja. Koriti se za izradu interaktivnih web stranica, web aplikacija, uključujući video igre. Većina web stranica to koristi, a svi moderni web preglednici to podržavaju bez potrebe ugrađivanja JavaScript motora. Logo JavaScript-a možemo vidjeti na slici 18.

Slika 18. JavaScript logo

Kao multi-paradigmatski jezik, JavaScript podržava stilove programiranja temeljenje na događajima, kao i funkcionalne i imperativne stilove programiranja (poput objektnog programiranja – Python). Također, ima API za rad s tekstom, nizovima, datumima, regularnim izrazima i osnovnom manipulacijom DOM-a, ali sam jezik ne obuhvaća I/O, kao što su umrežavanje, pohrana ili grafički objekti. [6]

4.4. Biblioteka JavaScript-a: jQuery

Biblioteka JavaScripta s više platformi dizajniranih da pojednostavljuje klijentima pisanje skripti HTML-a zove se jQuery. To je besplatan softver otvorenog koda koji koristi MIT licencu. Web analiza nam govori da je to najšire korištena JavaScript biblioteka.

Sintaksa jQuery-a je dizajnirana kako bi olakšala kretanje dokumentom, odabrala DOM elemente, kreirala animacije, obrađivala događaje i razvila Ajax-ove aplikacije. Također pruža mogućnost razvojnim programerima za stvaranje dodataka na vrhu biblioteke JavaScript-a. To omogućuje razvojnim programerima stvaranje apstrakcija za nisku raznu interakcije i animacije, napredne efekte i tematizirane *widgete* visoke razine. Modularni pristup JQuery biblioteci omogućuje stvaranje snažnih dinamičkih web stranica i web aplikacija. Logo jQuery-a možemo vidjeti na slici 19. [7]

Slika 19. jQuery logo

4.5. Alat otvorenog koda za razvoj HTML-a, CSS-a i JavaScripta-a: Bootstrap

Besplatna biblioteka otvorenog koda za izradu web stranica i web aplikacija korištena u radu zove se Bootstrap. Sadrži HTML i CSS temeljne predloške za tipografiju, obrasce, gumbe, navigaciju i ostale komponente sučelja, kao i dodatna JavaScript proširenja. Za razliku od mnogih web kostura, ona se bavi samo razvojem '*front-end*'-a (dijelovi web stranice namijenjeni za prikaz korisniku u smislu dizajna, interaktivnosti, animacije, efekata, tranzicija, galerija, menija i slično). Logo Bootstrap-a možemo vidjeti na slici 20. [8]

Slika 20. Bootstrap logo

5. IZRADA APLIKACIJE ZA MREŽNO UPRAVLJANJE ZNANJEM

Kao što je već rečeno, aplikacija za mrežno upravljanje znanjem bit će izvedena u web2py-u, pomoću ranije navedenih tehnologija za uređivanje. Izraditi će se aplikacija za mrežno povezivanje učenika/studenata i profesora. Profesori će moći postavljati predavanja (cijela ili sažeto da se ponovi gradivo) koje će učenici/studenti moći po različitim kategorijama gledati i preuzimati. Također profesorima je omogućeno postavljanje ispita, kako bi mogli provjeravati znanje svojih učenika/studenata. Uz tekstove, mrežno gradivo sadrži i razne multimedijske tehnike. Različiti formati prenošenja znanja pomažu većoj zainteresiranosti učenika/studenata te potiču njihovu kreativnost i volju za učenjem. U današnje vrijeme kada su tehnika i internet svakodnevica, potrebno je i školu prilagoditi tome kako bi se ona približila učenicima/studentima.

Diplomskim zadatkom potrebno je odabrati i realizirati dva kolegija - jedan za učenike i jedan za studente. Proizvoljno je odabrana biologija za sedmi razred te elektrotehnika za Fakultet strojarstva i brodogradnje. Ta dva kolegija razrađena su u potpunosti, dok postoji još nekoliko kolegija koji su djelomično razrađeni kako bi se pokazala snaga i mogućnost web2py-a, kao i same mrežne povezanosti. U tim kolegijima prikazani su različiti formati i multimedijske tehnike koje učenje čine lakšim i zanimljivijim.

Pomoću MVC paradigme izrada aplikacije se dijeli na 3 velike cjeline:

- „*Models*“ – sadrži *db.py* u kojem se definiraju baze podataka, te *menu.py* u kojem definiramo izbornik koji želimo ili ne želimo imati, kao i njegov izgled. Izbornik je već napravljen u sklopu web2py-a , pa ga možemo prilagoditi sebi ili ga u potpunosti promijeniti ili čak i izbrisati.
- „*Controller*“ – sadrži *appadmin.py* koji ne mijenjamo, niti ne dopunjavamo zato što nam web2py omogućava već programiran sustav za administratore, te sadrži *default.py* u koji upisujemo naše stranice poput funkcija i logiku koja je potrebna za stranicu, kao i povratne informacije u obliku rječnika.
- „*View*“ – sastoji se od raznih HTML datoteka te u njemu definiramo svoje datoteke koje stvaraju naše stranice u aplikaciji. HTML datoteke su nazvane

  `default/X.html`

  gdje je X ime stranice koju trenutno stvaramo.

Uz MVC u web2py aplikaciji također još koristimo i „*static*“. U folder „*images*“ dodajemo slike koje se prikazuju na stranici (kao logo, naslov i slično), dok u folderu „*css*“ možemo napraviti naš CSS dokument:

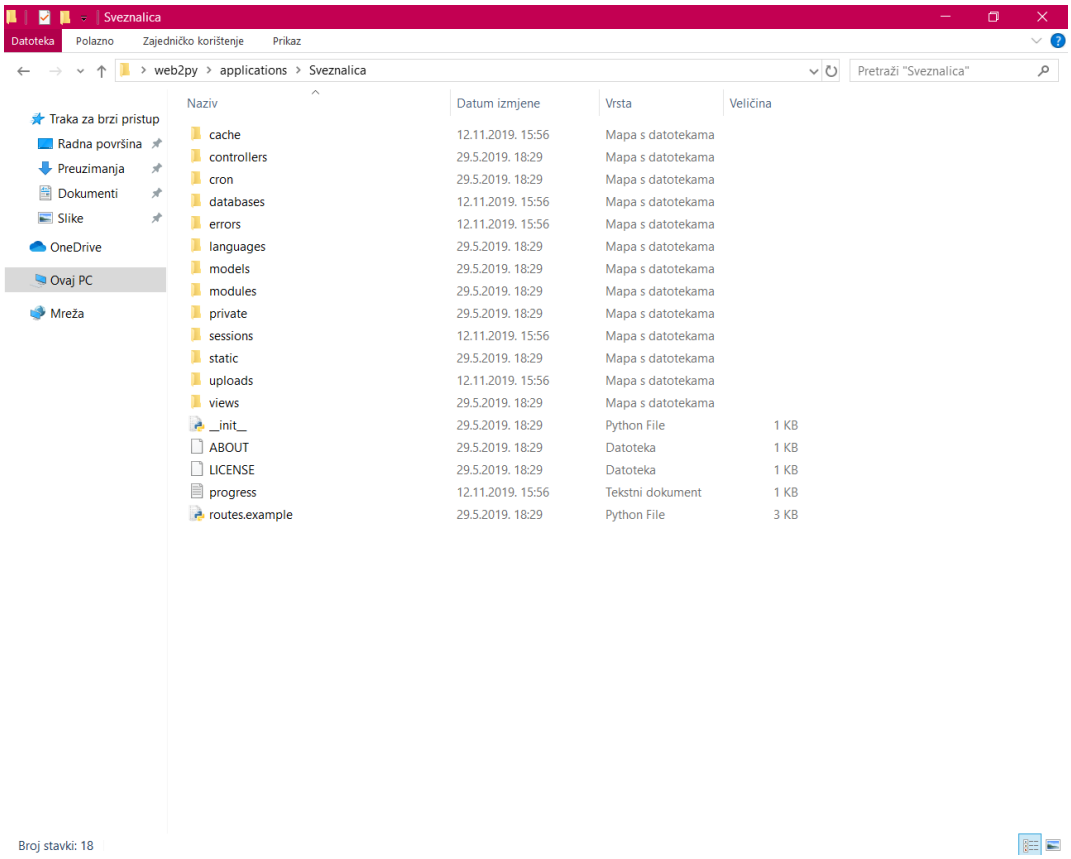
```
Y.css,
```

gdje je Y ime CSS dokumenta. On sadržava stil koji primjenjujemo na stranici, a pozivamo ga u HTML dokument pomoću oznake <*link*>. No, to nije jedini način na koji stil (CSS) možemo primijeniti na stranicu. Također, možemo CSS primijeniti odmah u HTML datoteci pomoću oznake <*style*>, pa na taj način nemamo dva dokumenta (CSS i HTML), nego samo jedan (HTML).

Nadalje, koristimo i datoteku „*uploads*“. U nju se spremaju svi dokumenti (u ovom slučaju slike i video uradci) koji se objave pomoću naše aplikacije (što profesori objave).

Ostatak web2py dokumenata neće se dirati.

Izgled web2py aplikacije spremljene na računalo u obliku datoteka vidljiv je na slici 21.



Slika 21. Izgled aplikacije u obliku datoteka na računalu

6. APLIKACIJA „Sveznalica“

Kao što je ranije već navedeno, aplikacija za mrežno upravljanje znanjem nazvana je „Sveznalica“. Ideja Sveznalice je da nakon prijave učenik/student može pristupiti svom predviđenom gradivu, ponoviti ili naučiti ga na zabavan i lakši način, te na kraju provjeriti svoje znanje kvizom. Za to vrijeme ideja za profesore je da mogu postavljati razne tekstove, slike i video uratke kao dijelove predavanja, te u konačnici postaviti pitanja za gore navedeni kviz, kako bi provjerili naučeno znanje svojih učenika/studenata. Dalje će u radu biti objašnjena izrada stranica te njihova funkcija, kao što su stranica za prijavu, stranica s gradivom, stranica s videima, stranica sa kvizom i slično.

6.1. Tema aplikacije

Iako web2py već ima svoj gotov raspored, tj. izgled (eng. *layout*), pomoću CSS-a su promijenjene neke njegove značajke, dok su neke stvari promijenjene u potpunosti. Kroz cijelu aplikaciju se provlači ista tema: izgled stranice i izbornika uz minimalne promjene (različita boja za različite kolegije). CSS kod te teme je prikazan programskim kodom 1.

Programski kod 1. CSS kod teme aplikacije

```
.navbar-inverse {
  background-image: linear-gradient(to bottom, white, white);
  background-repeat: repeat-x;
  border: transparent;
  border-radius: 7px 7px 7px 7px;
  min-height: 40px;
  padding-left: 20px;
  padding-right: 20px;
  }
.dropdown-menu {
  background-clip: padding-box;
  background-color: white;
  border: 1px solid #004d80;
  border-radius: 6px 6px 6px 6px;
  box-shadow: 0 5px 10px rgba(0, 0, 0, 0.2);
  display: none;
  float: left;
  left: 0;
  list-style: none outside none;
  margin: 2px 0 0;
  min-width: 160px;
  padding: 5px 0;
  position: absolute;
  top: 100%;
  z-index: 1000;
  }
.navbar .nav > li > a {
  color: #33adff;
  float: none;
```

```
30.   padding: 15px 15px;
31.   text-decoration: none;
32.   font-family: Trebuchet MS;
33.   font-size:20px;
34.   }
35. .dropdown-menu li > a {
36.   display: block;
37.   padding: 3px 20px;
38.   clear: both;
39.   font-weight: normal;
40.   line-height: 20px;
41.   color: #33adff;
42.   white-space: nowrap;
43.   font-family: Trebuchet MS;
44.   font-size: 20px
45.   }
46. footer {
47.   background-color: #33adff;
48.   text-align: center;
49.   color: white;
50.   position: fixed;
51.   left: 0;
52.   bottom: 0;
53.   width: 100%;
54.   }
```

Linije 1-9 opisuju navigacijsku traku vidljivu na slici 22, tj. izgled navigacijske trake koja se nalazi na vrhu stranice. Na navigacijskoj traci se inače nalazi izbornik, no ova aplikacija je nema, pa se na njoj samo s lijeve strane nalazi logo, a s desne gumb za prijavu/odjavu. Odabrano je da bude bijele boje, no može biti bilo koje, a može biti i linearni prijelaz boje iz bijele u bilo koju drugu. Uzorak se može ponavljati po x i y osi, u ovom slučaju se ponavlja po x. Rub je proziran, te zakrivljen sa radijusom od 7 piksela. Također je određena i minimalna visina navigacijske trake koja iznosi 40 piksela. U zadnje dvije linije je određeno minimalno ispunjenje sa lijeve i desne strane teksta, koji bi se nalazio na navigacijskog traci kao jedan izbor izbornika.

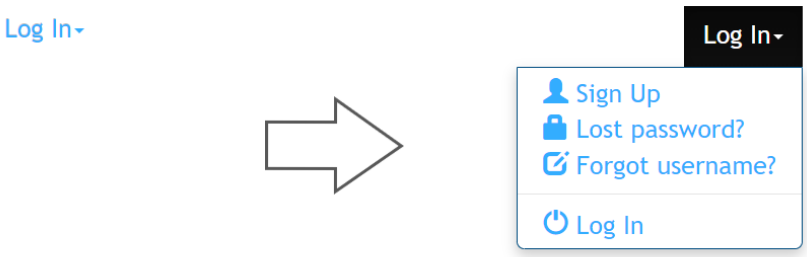
Slika 22. Navigacijska traka aplikacije

Linije 10-26 opisuju padajući izbornik koji je bijele boje, poput kvadrata ima vidljivi rub tamnoplave boje, dok mu je je radijus zakrivljenosti ruba 6 piksela. Taj kvadrat u kojem se nalazi padajući izbornik, tj. kutija, ima sjenu definiranu debljinom i bojom. Tekst je definiran tako da se nalazi s lijeve strane, margine su debljine 2 piksela, minimalna širina iznosi 160 piksela, dok ispunjenje oko teksta iznosi 5 piksela. Pozicija je apsolutna, što znači da je smješteno u odnosu na prvi postavljeni roditeljski element. Također, padajući izbornik se nalazi na samom vrhu.

Linije 27-34 opisuju navigacijsku traku u slučaju liste sa hiperlinkovima, tj. u slučaju gumba za neku akciju. Ovdje je opisana veličina slova i font: 20 piksela i Trebuchet MS, definirano je da tekst nema ukrasa, prostor oko teksta iznosi 15 piksela te je boja slova nijansa svijetlo plave.

Linije 35-45 opisuju padajući izbornik u slučaju liste sa hiperlinkovima. Kao u prethodnom, definirani su font, debljina, veličina slova, boja i ispuna oko teksta. Također je opisano ponašanje prikaza okvira za prikazivanje elemenata kao blok element.

Slika 23. prikazuje izgled stilizirane padajuće navigacijske trake i izbornika prije i poslije otvaranja „Log in“ gumba.



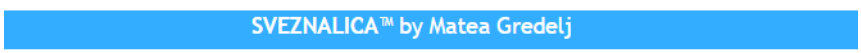
Slika 23. Padajući izbornik aplikacije

Linije 46-54 opisuju podnožje kao blok nijanse svijetlo plave boje, sa bijelom slovima, koji se nalazi fiksiran na samom dnu stranice, te se proteže čitavom duljinom. Također je definirano da se tekst nalazi u sredini. Podnožje se nalazi na svakoj stranici no mijenja mu se boja kako je već navedeno. U pogledu (eng. *view*) podnožje pozivamo u HTML datoteci način prikazan programskim kodom 2.

Programski kod 2. Podnožje aplikacije

```
1.  {{block footer}}
2.  <footer style="font-
    family: Trebuchet MS;"><b>SVEZNALICA</b>™ by Matea Gredelj</p></footer>
```

Prvo je potrebno blokirati podnožje koje dolazi uz web2py raspored aplikacije što radimo u prvoj liniji koda, dok u drugoj pozivamo svoje podnožje i u njega upisujemo gore navedeni tekst. Izgled uređenog podnožja vidljiv je na slici 24.



Slika 24. Stilizirao podnožje aplikacije

Na slici 22 možemo vidjeti da se u gornjem lijevom kutu nalazi logo. Njega definiramo pomoću modela u *menu.py* na način prikazan programskim kodom 3.

Programski kod 3. Logo rađene aplikacije

```
1.  response.logo = IMG(_src=URL('static','images/sveznalica.png'),
2.                      _style="height:50px; width:auto; position:right;",
3.                      _alt=T('Sveznalica-logo'))
```

U prvoj liniji koda logo je pozvan kao slika uz pomoć atributa *„src“* koji određuje URL slike, koji je ovdje u mapi aplikacije spremljenoj na računalo. Unutar aplikacije „Sveznalica“ imamo mapu *„static“* unutar koje je još jedna mapa *„images“* u kojoj se nalazi naša slika „sveznalica.png“ koju koristimo kao logo. U idućoj liniji koda stiliziramo tu sliku, dok u trećoj liniji koda nudimo alternativu za sliku. U slučaju da aplikacija ne može dohvatiti sliku iz nekog razloga (izbrisana iz mape, preimenovana i slično) na mjestu loga će pisati „Sveznalica-logo“.

Također, potrebno je izbrisati dio već gotovog web2py koda kako bi sa navigacijske trake izbacili izbornik web2py-a. Na taj način nam s lijeve strane ostaje samo logo, a s desne padajući izbornik za prijavu/odjavu.

6.2. Sustav prijave

Kako aplikaciji ne bi mogao pristupiti bilo tko, već samo učenici/studenti određenih škola/fakulteta, potrebno je napraviti sustav prijave. Znači, potrebno je napraviti bazu podataka koja će sadržavati osobne podatke učenika, studenata i profesora: ime, prezime, korisničko ime, e-mail adresu i lozinku. Zatim je potrebno napraviti obrazac kojim bi se prijavljivalo. Sve to može se raditi od nule, no nepotrebno. web2py već ima svoju bazu podataka za prijavu te svoj gotovi obrazac koji možemo prilagoditi svojim potrebama.

Kao što je ranije navedeno, potrebna baza podataka je već dijelom web2py aplikacije i zove se:

```
db.auth_user.
```

U nju pomoću „*database administrationa*“ upisujemo korisnike pazeći na upisane podatke kako bi kasnije mogli dati profesorima veće ovlasti.

Nakon toga potrebno je u default.py definirati funkciju stranice što je u ovom slučaju vraćanje obrasca (eng. *form*) za prijavu. Način kojim to definiramo prikazan je programskim kodom 4.

Programski kod 4. Definiranje funkcije stranice

```
1.  def index():
2.      response.title = "Login"
3.      return dict(form=auth())
```

Ovdje također definiramo i naslov naše stranice koji je u ovom slučaju „Login“.

Slijedeći MVC paradigmu, treći korak je „*view*“. Njega definiramo u datoteci:

`default/index.html`.

U navedenoj HTML datoteci potrebno je pozvati obrazac za prijavu kako je prikazano u programskom kodu 5.

Programski kod 5. Obrazac za prijavu

```
1.  <div id="frame" style="font-family:Trebuchet MS; color:#4db8ff; font-size:20px">
2.      <h2>{{=T( request.args(0).replace('_',' ').capitalize() )}}</h2>
3.      <div id="web2py_user_form">
4.          {{=form}}
5.          {{if request.args(0)=='login':}}
6.          {{if not 'register' in auth.settings.actions_disabled:}}
7.          {{pass}}
8.          {{if not 'request_reset_password' in auth.settings.actions_disabled:}}
9.          <br/>
10.         <a href="{{=URL(args='request_reset_password')}}" style="color:#008ae6; font
    -size:15px">Zaboravili ste lozinku?</a>
11.         {{pass}}
12.         {{pass}}
13.     </div>
14. </div>
```

U prvoj liniji koda otvaramo tag <*div*> kako bi smo započeli novi odjeljak. Također, prikazano je kako možemo stil, tj. CSS, definirati unutar *taga*. Definirani su font, boja i veličina slova. Ovdje također imamo id odjeljka. Id je atribut koji specificira jedinstveni id za HTML element (vrijednost mora biti jedinstvena u HTML dokumentu). Vrijednost id-a mogu koristiti i CSS i JavaScript za izvođenje određenih zadataka za element s određenom id vrijednosti. U ovom slučaju koristimo ga kao CSS i definiramo kako je prikazano programskim kodom 6.

Programski kod 6. Stil okvira

```
1.  <style>
2.      #frame {
3.              background: white;
4.              border: 3px solid white;
5.              padding: 100px;
6.              margin: auto;
7.              width: 40%;
8.              height: 30%;
9.              position: relative;
10.             top: 80px;
11.             text-align: center;
12.             }
13. </style>
```

Id nazvan „*frame*“ je zapravo okvir u kojem se nalazi naš obrazac za prijavu. On je bijele boje sa 3 piksela debelom bijelom granicom. Prostor oko teksta iznosi 100 piksela. Širina okvira iznosi 40% ukupne širine stranice, dok je visina 30%. Okvir se nalazi 80 piksela od vrha u odnosu na svoj normalni položaj. Također, tekst koji se nalazi u okviru je centriran. Uz ovaj

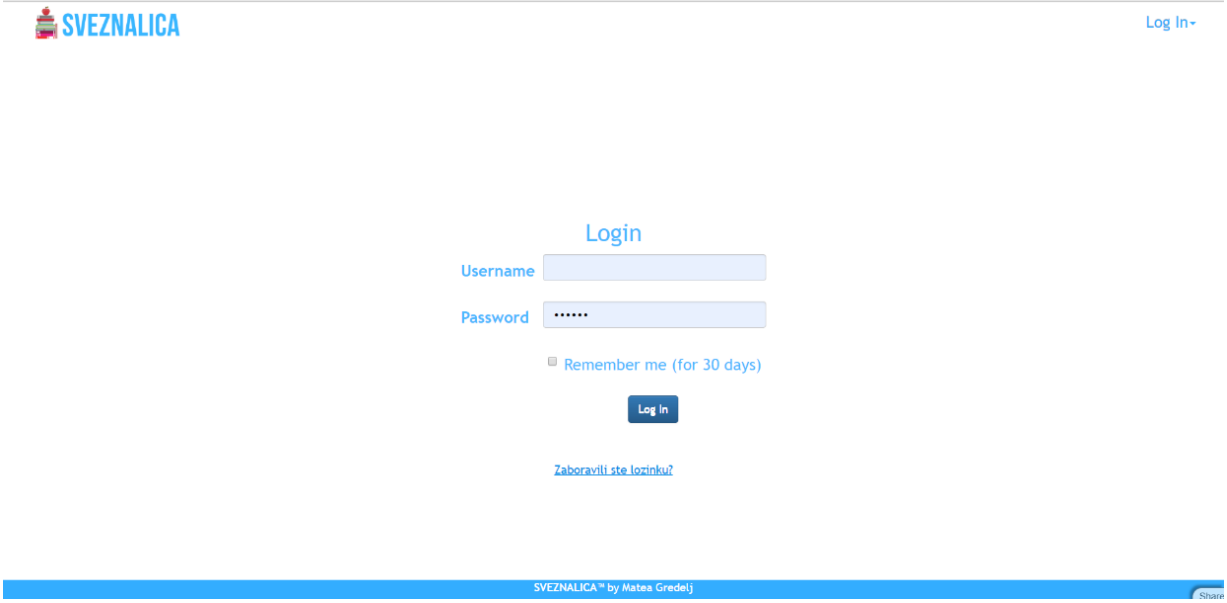
CSS, u stranicu je uključen i CSS iz prethodnog poglavlja. Kako je on uključen na skoro svakoj stranici, to se više neće spominjati. U slučaju da tog CSS-a nema, isto će biti naglašeno.

Kada imamo okvir, potrebno je i u njega postaviti obrazac. Nužno je stoga pogledati u prethodni programski kod. U njegovoj drugoj liniji pozivamo iz baze naslov obrasca: „Login“, a on je u *<h2>* stilu. U idućoj liniji otvaramo odjeljak sa id-om:

```
web2py_user_form
```

koji je već ugrađen u web2py. U četvrtoj liniji koda pozivamo obrazac koji smo također bili definirali kroz *default.py*. U linijama 5 do 11 imamo dvije if logike. Prva logika nam govori da ako osoba nije registrirana ne može pristupiti dalje stranicama, dok nam druga govori ako smo registrirani, no zaboravili lozinku, da klikom na tekst „Zaboravili ste lozinku?“ možete biti preusmjereni na drugu stranicu na kojoj možete promijeniti lozinku. Preusmjeravanje je moguće zbog definiranja hiperlinka pomoću taga *<a>* unutar koje imamo atribut *href* koji određuje URL stranice na koju poveznica ide. Taj link je također stiliziran. Stranica na koju se preusmjerava je web2py-ova stranica sa gotovim obrascem za promjenu lozinke.

Na kraju, kada povežemo sve navedeno dobivamo gotovu stranicu sa obrascem za prijavu iza kojeg se nalazi potpuni sustav autorizacije što je prikazano slikom 25.



Slika 25. Stranica za prijavu

6.3. Početna stranica

Nakon što se osoba prijavi na aplikaciju potrebno je otvoriti novu stranicu – početnu stranicu aplikacije, što je potrebno definirati u modelu. Također je potrebno definirati kuda se korisnik

vraća kada se odjavi. Programski kod 7 prikazuje način definiranja preusmjeravanja stranice nakon prijave/odjave.

Programski kod 7. Preusmjeravanje nakon prijave/odjave

```
1.  auth.settings.login_next = URL('pocetna')
2.  auth.settings.logout_next = URL('index')
```

Prva linija koda govori nam da se nakon prijave korisnika preusmjerava na početnu stranicu, dok se nakon odjave korisnika preusmjerava na stranicu za prijavu (stranica objašnjena u prethodnom poglavlju).

Kako bi definirali početnu stranicu, ponovno je potrebno definirati funkciju stranice u upravitelju (eng. *controller*). To se, kao što je već prethodno objašnjeno, radi u datoteci *default.py*, a prikazano programskim kodom 8.

Programski kod 8. Definiranje početne stranice

```
1.  @auth.requires_login()
2.  def pocetna():
3.      response.title= "Sveznalica"
4.      return dict()
```

Prva linija koda ukazuje na to da je potrebno biti prijavljen kako bi se moglo pristupiti stranici. U ostatku koda imamo definiranu funkciju koja daje naslov našoj stranici u aplikaciji te vraća prazan rječnik. Većina idućih stranica u aplikaciji je definirana na ovaj način pa se to neće više posebno objašnjavati. U slučaju da je funkcija drugačija, bit će naglašeno.

Kako imamo definiranu funkciju, potrebno je stvoriti HTML datoteku:

`default/pocetna.html.`

Početna stranica je zamišljena kao izbornik s tri gumba: za osnovnu školu, za srednju školu i za fakultet. Svaki gumb u sebi sadrži padajući izbornik koji nudi razrede za školu, te različite fakultete.

Kao što je do sada već i pokazano, potrebno je napraviti CSS kako bi stranica bila stilizirana. U ovom slučaju potrebno je stilizirati gumbe i padajuće izbornike što je prikazano programskim kodom 9.

Programski kod 9. Stiliziranje padajućeg izbornika i gumba

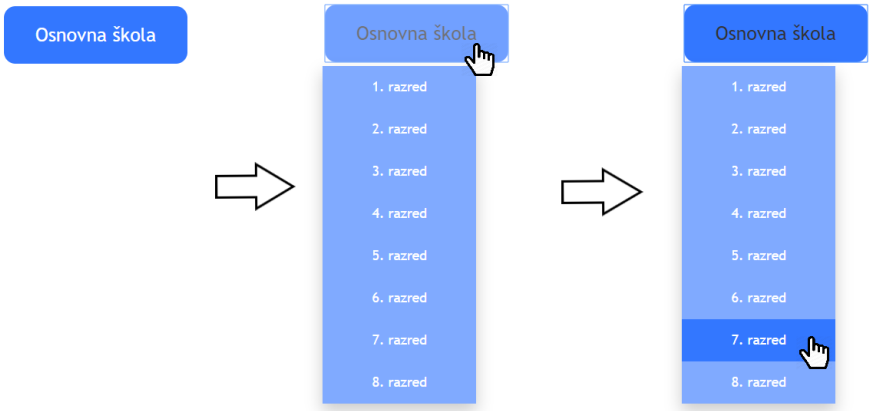
```
1.  <style>
2.     .dropbtn1 {
3.         background-color: #3377ff;
4.         color: white;
5.         padding: 15px 32px;
6.         text-align: center;
7.         display: inline-block;
```

```
8.          font-size: 20px;
9.          font-family: Trebuchet MS;
10.         margin: 4px 2px;
11.         border-radius: 12px;
12.         cursor: pointer;
13.         opacity: 1;
14.         transition: 0.3s;
15.     }
16.
17.    .dropbtn1:hover {opacity: 0.7}, .dropbtn1:focus {
18.         background-color: #004de6;
19.      }
20.
21.      .dropdown1 {
22.          position: relative;
23.          display: inline-block;
24.      }
25.
26.      .dropdown1-content {
27.          display: none;
28.          position: absolute;
29.          background-color:#80aaff;
30.              min-width: 160px;
31.          overflow: auto;
32.          box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
33.      }
34.
35.      .dropdown1-content a {
36.          color: white;
37.          font-family: Trebuchet MS;
38.          padding: 12px 16px;
39.          text-decoration: none;
40.          display: block;
41.      }
42.
43.      .dropdown1-content a:hover {background-color: #3377ff}
44.
45. </style>
```

Prikazani kod samo je trećina CSS-a potrebnog za stil. Ovdje je opisan stil za prvi gumb i njegov padajući izbornik. Ostatak koda nije prikazan zato što je jednak ovome, samo su promijenjene boje te brojevi u nazivima.

„.dropbtn1“ definira izgled prvog gumba: da je plave boje s bijelim slovima veličine 20 piksela koja se nalaze u sredini gumba. Većina koda je već ranije objašnjena i relativno jasna sama po sebi. U linijama od 13 do 17 definiramo da u početku gumb nije proziran, no kada na njega dođemo mišem i pritisnemo ga, postaje 30% prozirniji u roku od 0.3 sekunde.

„.dropdown1“ definira izgled padajućeg izbornika: boju, oblik, font, veličinu i slično. 38. linija koda opisuje izgled (boju) trenutnog izbora izbornika. Uz HTML kod, koji će kasnije biti objašnjen, izgled gumba i padajućeg izbornika sa ovdje opisanim stilom možemo vidjeti na slici 26.



Slika 26. Prikaz padajućeg izbornika

Potreban HTML kod prikazan je programskim kodom 10.

Programski kod 10. Padajući izbornik

```
<div class="row">
        <div class="col-sm-4" style="text-align: center;">
            <div class="dropdown1">
                <button id="myBtn1" class="dropbtn1">Osnovna škola</button>
                <div id="myDropdown1" class="dropdown1-content">
                    <a href="ups">1. razred</a>
                    <a href="ups">2. razred</a>
                    <a href="ups">3. razred</a>
                    <a href="ups">4. razred</a>
                    <a href="ups">5. razred</a>
                    <a href="ups">6. razred</a>
                    <a href="sedmi">7. razred</a>
                    <a href="ups">8. razred</a>
                </div>
            </div>
        </div>
        …
</div>
```

U prvoj liniji koda, u stvaranju novog odjeljka, možemo vidjeti klasu. Klasa je atribut koji HTML koristi za definiranje jednakih stilova za elemente s istim nazivom klase. Ova klasa je automatski povezana sa Bootstrapom, to jest sa Bootstrapovim izgledom mreže stranice (eng. *grid*). U drugoj liniji koda se pojavljuje također Bootsrapova klasa. Ona nam govori kako je radna površina stranice podijeljena na tri stupca jednake širine, te se ovaj gumb i padajući izbornik nalaze u prvom stupcu mreže. Također, pomoću klasa pozivamo stil koji je objašnjen u prethodnom programskom kodu i primjenjujemo ga na gumb „Osnovna škola“, te na njegov padajući izbornik. Padajući izbornik se sastoji od osam linkova, od kojih je sedam isto i vodi na stranicu „Ups…“, dok jedan razred (sedmi) vodi na stranicu „7. razred“. Zadatkom je potrebno projektirati samo jedan premet, tako da ovdje nije sve projektirano, no kada bi se projektiralo, bilo bi isto kao i za sedmi razred, samo s malo drugačijim sadržajem.

U sedamnaestoj liniji koda možemo vidjeti tri točkice što označava da se tu nalazi još linija koda. Linije koda su skoro pa identične ovima, pa ih nije potrebno navoditi. Prikazano je kako je to napravljeno za osnovnu školu, dok za srednju školu i fakultet primijenimo istu logiku izrade (promijenimo nazive).

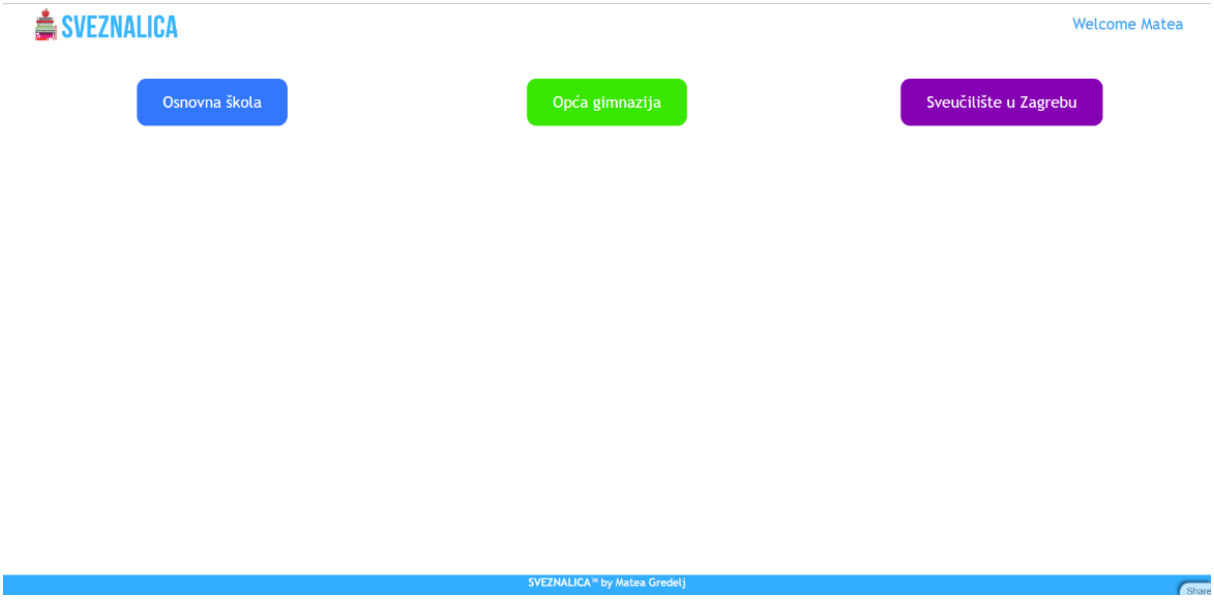
Uz klase možemo primijetiti da ovdje imamo i id. Taj id povezujemo sa skriptom prikazanom programskim kodom 11.

Programski kod 11. Povezivanje funkcije pomoću klasa

```
1.  <script>
2.          document.getElementById("myBtn1").onclick = function() {myFunction1()};
3.          function myFunction1() {
4.              document.getElementById("myDropdown1").classList.toggle("show");
5.          }
6.  </script>
```

Kao i ranije, ovdje je samo prvi dio, jer su ostala dva skoro pa identična. Kako bi se na dodir gumba otvorio padajući izbornik, potrebno je napraviti funkciju koja će to raditi. Ovdje vidimo funkciju koja preko id-a elementa u kodu poziva funkciju koja na pritisak gumba otvara, tj. prikazuje padajući izbornik.

Kada se sve to poveže dobivamo gotovu početnu stranicu za izbor razine školovanja prikazanu slikom 27.



Slika 27. Izgled početne stranice aplikacije

6.4. Stranica „7. razred“

Kao što je u prethodnom poglavlju spomenuto, odabrani predmet koji je razrađen u potpunosti je biologija za sedmi razred. No, kako bi došli do biologije, potrebno je napraviti izbornik za sedmi razred.

Na stranici se nalazi 8 gumbi raspoređenih u dva reda koji predstavljaju linkove za stranice s predmetima, te jedan gumb u trećem redu koji vodi na repozitorij video uradaka koje profesori postavljaju („Videoteka“). Kao i na prethodnoj stranici, to je napravljeno pomoću Bootstrapove mreže. Stil gumbova je definiran samo jednom u CSS-vom stilu, dok je kod pozivanja gumbova u HTML-u samo on nadopunjen bojom kako bi se gumbi razlikovali – svaki predmet ima svoju boju. Programski kod za ovo nije potrebno dodatno objašnjavati jer je sve već viđeno i objašnjeno. Jedino što je potrebno nadodati da je u donjem desnom kutu dodan link za povratak na prethodnu stranicu, tako na pritiskom strelicu i tekst link nas odvodi na prethodnu stranicu, tj. na „Početnu“ stranicu što je prikazano programskim kodom 12.

Programski kod 12. Link za prethodnu stranicu

```
1.  <div>
2.      <a href="pocetna" style="color:#33adff;font-size:15px;font-
    family: Trebuchet MS;"><b>⇦ Prethodna</b></a>
3.  </div>
```

Izgled uređene stranice za 7. razred vidljiv je na slici 28.

Slika 28. Stranica „7. razred“

6.5. Stranica „Ups…“

Kako stranice predmeta koje nije potrebno trenutno razraditi nisu definirane, umjesto njih je napravljena univerzalna stranica zvana „Ups…“ na kojoj je obavijest da stranica još ne postoji, odnosno da je u izradi. Također, nudi se mogućnost povratka na prethodnu stranicu.

Ova stranica je veoma zanimljiva zbog svojeg CSS-a. U HTML datoteci imamo *<svg> tag* pomoću kojeg otvaramo polje za izradu grafike na način prikazan programskim kodom 13.

Programski kod 13. Grafika HTML-a: stvaranje polja grafike te crtanje linije, izlomljene linije i kruga u njemu

```
1. <svg>
2.     <g>
3.         <line x2="227.62" y1="31.28" y2="31.28"></line>
4.         <polyline points="222.62 25.78 228.12 31.28 222.62 36.78"></polyline>
5.         <circle cx="224.67" cy="30.94" r="30.5" transform="rotate(180 224.67 30.94)
   scale(1, -1) translate(0, -61)"></circle>
6.     </g>
7. </svg>
```

Tag <g> označava zajedničku grupaciju idućih elemenata: linije, izlomljene linije i kruga. Linija je definirana x i y početnim i završnim koordinatama, izlomljena linija (završna strelica) je također definirana koordinatama koje se povezuju, dok je krug definiran koordinatama središta kruga te radijusom. Također u petoj liniji koda možemo vidjeti da se događa neka transformacija sa krugom kada dodirnemo taj grafički element – krug nestane.

CSS dokument za ovu grafiku se nalazi u prilogu zato što je dosta opširan.

Većina koda je u radu već bila objašnjena: boja, veličina fonta, font, poravnanje i slično. Novitet je svojstvo „*stroke*“ koje definira boju, debljinu, različite krajeve i vrste isprekidane i obične linije, teksta ili obrisa elemenata, debljinu. Također, ovdje su definirane i transformacije gumba po z i x osi te njihov vremenski period transformacije.

Još jedna zanimljivost u ovom CSS kodu je to da ne koristimo postotke ili piksele kao jedinice nego rem i em, gdje je rem jedinica u odnosu na veličinu korijenskog elementa, dok je em jedinica u odnosu na trenutnu veličinu fonta (2em znači dva puta veću veličinu fonta).

Na kraju dobivamo stranicu čiji izgled možemo vidjeti na slici 29.

Slika 29. Uređena stranica „Ups…“

Kao što možemo vidjeti ovdje se nalazi i slika, tj. .gif format. On se inače nalazi u mapi „*images*“ aplikacije, a poziva se na način prikazan programskim kodom 14.

Programski kod 14. Pozivanje slike na stranicu

```
1.  <img src="{{=URL('static','images/ups.gif')}}" alt="Ups..." style="width:30%;">
```

6.6. Predmeti za osnovnu i srednju školu

Svi predmeti za osnovnu i srednju školu imaju isti kostur. Na početku na sredini stranice nalazi se ime predmeta sa slikom, može se reći na neki način logo predmeta, koji je u principu slika.

Nakon toga nalazi se jedna vrsta prezentacije (eng. *carousel*). Ona je Bootstrapov priključak. U njoj možemo vidjeti slike koje su specifične za određeno gradivo, a uz njih i naslov i opis tog gradiva, tj. velikog poglavlja gradiva. Prezentacija također sadrži strelice za lijevo i desno kako bi mogli pregledavati sadržaj, no može se i sama pokretati. Skraćenu verziju možemo vidjeti u programskom kodu 15.

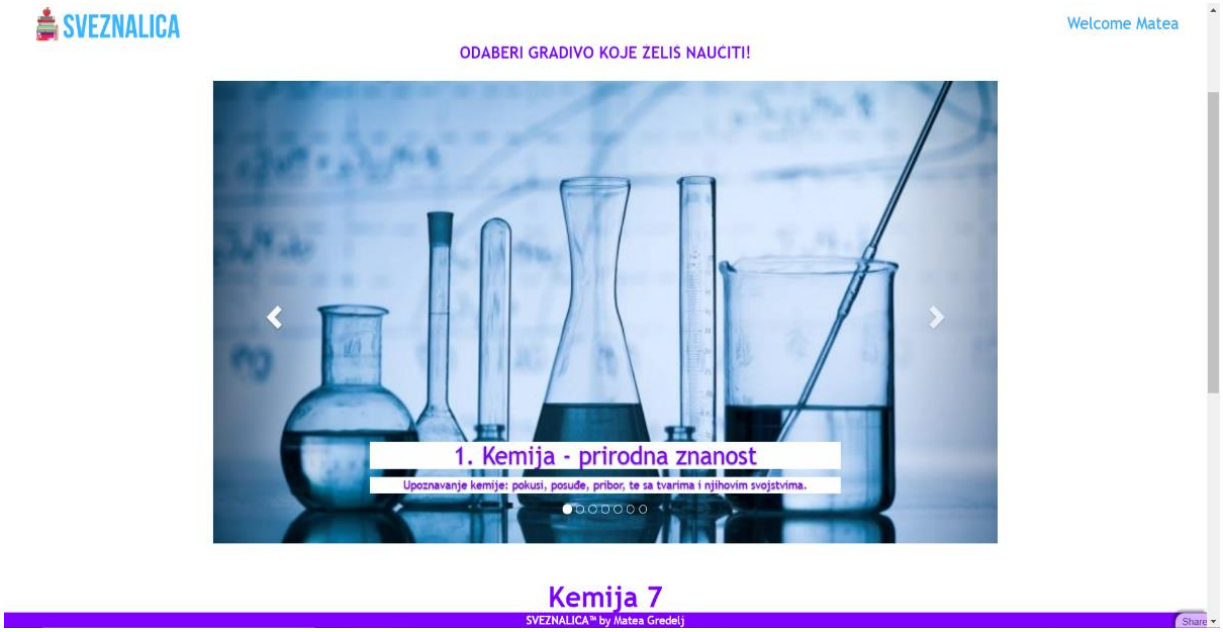
Programski kod 15. Bootstrap prezentacija (eng. *carousel*)

```
1.  <div class="container" style="width:50%;">
2.      <div id="myCarousel" class="carousel slide" data-ride="carousel">
3.          <ol class="carousel-indicators">
4.              <li data-target="#myCarousel" data-slide-to="0" class="active"></li>
5.              <li data-target="#myCarousel" data-slide-to="1"></li>
6.              <li data-target="#myCarousel" data-slide-to="2"></li>
7.              <li data-target="#myCarousel" data-slide-to="i"></li>
8.          </ol>
9.      <div class="carousel-inner">
10.         <div class="item active">
11.
```

```
12.             <a href="ups"><img src="{{=URL('static','images/ImePredmeta/ImeSlike')}}" al
    t="Ime Slike" style="width:100%;"></a>
13.             <div class="carousel-caption">
14.                 <h2>1. Naslov</h2>
15.                 <p>Opis</p>
16.             </div>
17.         </div>
18.         <div class="item">
19.             <a href="ups"><img src="{{=URL('static','images/ImePredmeta/ImeSlike')}}" al
    t="Ime Slike" style="width:100%;"></a>
20.             <div class="carousel-caption">
21.                 <h2>i. Naslov</h2>
22.                 <p>Opis</p>
23.             </div> 2
24.         </div>
25.         <div class="item">
26.             ...
27.         </div>
28.     </div>
29.
30.     <a class="left carousel-control" href="#myCarousel" data-slide="prev">
31.        <span class="glyphicon glyphicon-chevron-left"></span>
32.        <span class="sr-only">Prethodni</span>
33.     </a>
34.     <a class="right carousel-control" href="#myCarousel" data-slide="next">
35.         <span class="glyphicon glyphicon-chevron-right"></span>
36.         <span class="sr-only">Slijedeći</span>
37.     </a>
38. </div>
```

Ovo je poopćeni kod za Bootstrapov „vrtuljak“ prezentaciju koji je prilagođen za ovu aplikaciju. U sedmoj liniji koda broj je zamijenjen s varijablom i. Takvu istu liniju koda možemo ponavljati s obzirom na to koliko poglavlja (slika) imamo za pokazivanje. Tako da i može biti svaki broj veći od 2. Također, svaka slika predstavlja i link za gradivo. U linijama 11 i 18 su poopćeni naslovi mapa i imena slika. Ovisno o predmetu i imenu slike koji nam je potreban, to upisujemo. U dvadeset i petoj liniji možemo vidjeti tri točkice koje označavaju da ovdje ima još koda. Ovdje se ponavlja isti kod kao i u linijama 18-22. Koliko gore definiranih i-ova imamo, toliko ponovljenih linija 18-22 imamo. U linijama 29 do 36 definirane su strelice za lijevo i desno. CSS za ovo prilagođen je svakoj stranici u vezi boja, no fontovi, veličine slova i ostalo je sve u potpunosti isto i već do sada objašnjeno.

Izgled jedne takve prezentacije možemo vidjeti na slici 30 koja prikazuje početnu stranicu kemije.



Slika 30. Izgled prezentacije na stranici

Nakon toga nalazi se popis s glavnim poglavljima i njegovim potpoglavljima . Popis je složen prema Bootstrapovoj mreži u tri stupca. Također popis je zapisan u obliku uređene liste sa uređenom podlistom. Liste su također i linkovi za stranice s gradivom (predavanjima). Kod za listu poglavlja iz kemije za sedmi razred možemo vidjeti u kodu koji se nalazi u prilogu, a prikaz toga na rađenoj stranici na slici 31.

Slika 31. Izgled lista na stranici uređene Boostrapovom mrežom

Takve iste liste (samo s drugim sadržajem i drugom bojom) se nalaze na svim uređenim stranicama predmeta.

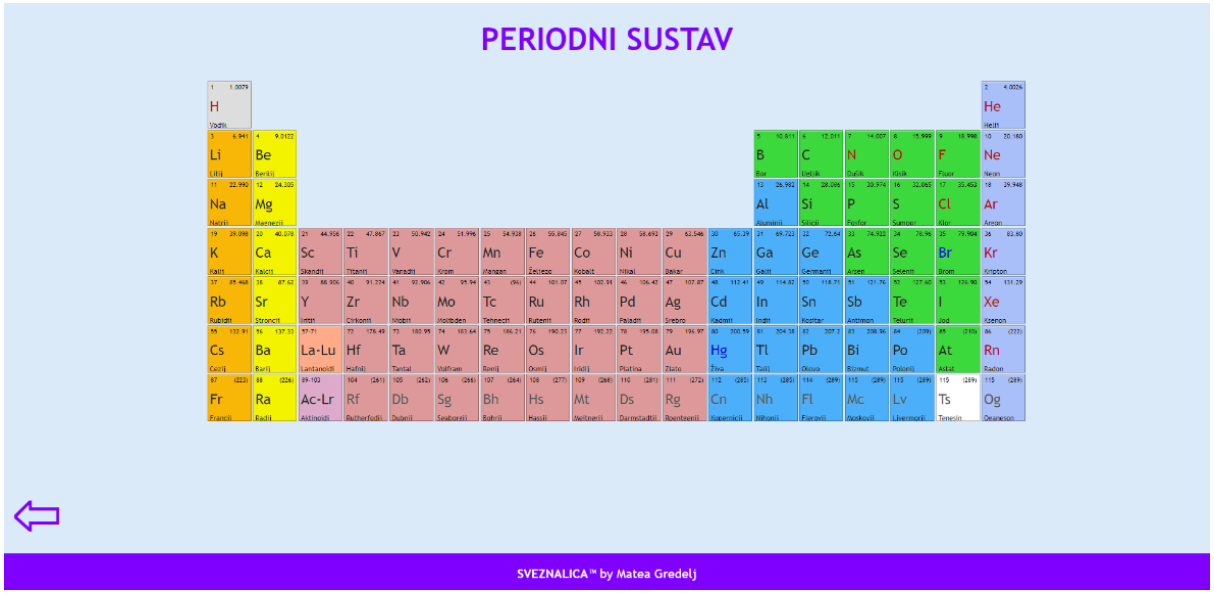
6.7. Multimedijski dodaci

Kako bi se učenje i gradivo približilo učenicima/studentima potrebno je gradivo prikazati kroz razne formate i multimedijske tehnike. Kroz ovaj rad biti će prikazano samo nekoliko dodataka za stranice kako bi se obogatio njihov sadržaj. Dalje prikazan sadržaj je samo dio mogućnosti web2py-a, kao i samog web programiranja.

Za predmet kemiju napravljen je interaktivni periodni sustav, dok je za matematiku napravljena simulacija kalkulatora, te program za računanje rješenja kvadratne jednadžbe. Uz to, napravljene su još dvije stranice za geografiju: interaktivna karta, te stranica sa grafičkim prikazima statističkih podataka vezanih uz gradivo. Sve to pobliže je objašnjeno u idućim poglavljima.

6.8. Stranica „Periodni sustav“

Cijeli programski kod stranice „Periodni sustav“ nalazi se u prilogu, a tu će biti objašnjen. Slika 32 prikazuje gotovu stranicu „Periodni sustav“.



Slika 32. Stranica „Periodni sustav“

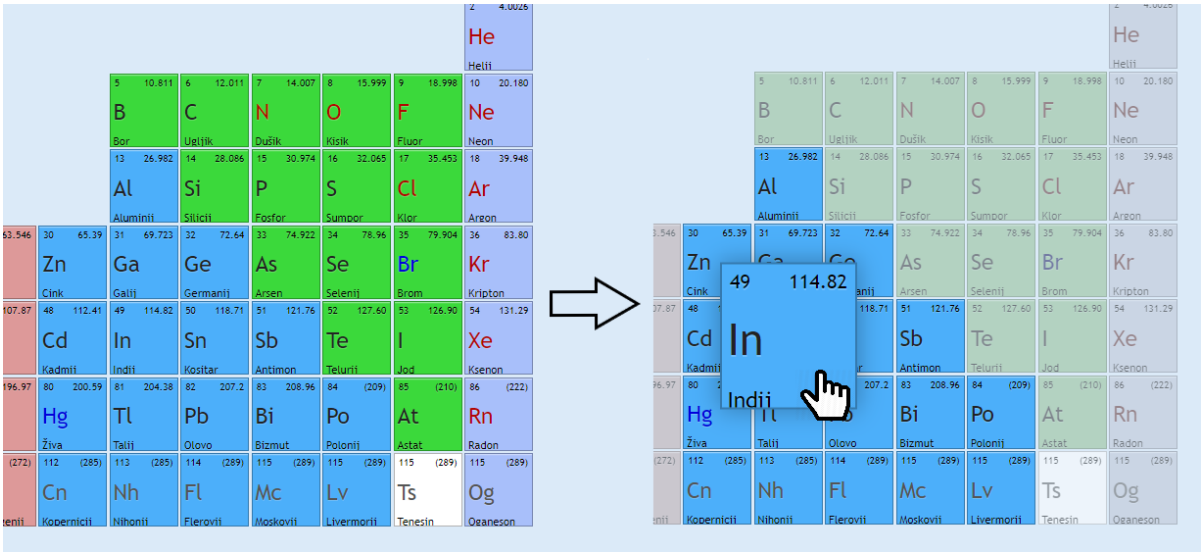
Periodni sustav elemenata je sustavni tablični poredak kemijskih elemenata koji odražava njihovu atomsku građu i sličnost njihovih fizikalnih i kemijskih svojstava. Svi elementi u periodnom sustavu naznačeni su svojim kemijskim simbolima – u sredini, punim imenom – dolje lijevo, te oznakama: atomski (redni) broj – gore lijevo; maseni broj – gore desno.

Stil (CSS) za periodni sustav napravljen je u sklopu HTML dokumenta: kroz *tag* <*style*>. Kroz klase definiramo oblik kućica, njihove boje, pozicije teksta, transformacije i slično.

Većina koda već je objašnjena u prethodnim poglavljima pa ovdje nije potrebno ulaziti u detalje.

Periodni sustav je u HTML dokumentu prikazan kao lista koja pomoću klasa poprima oblik nama poznatog periodnog sustava elemenata. Prazni prostori u periodnom sustavu elemenata su također dijelovi liste s različitom klasom kako bi bili prikazani kao nepostojeći.

Kako bi periodni sustav elemenata bio interaktivan uključujemo funkciju koja na prelazak miša iznad elementa povećava kućicu s elementom i njegovim podacima, a deaktivira (mijenja boju) svih ostalih elemenata druge skupine. Skupine elemenata se razlikuju po bojama. Funkcije su napravljene pomoću jQuery-a. Funkcija aktivira i deaktivira klase koje su definirane u CSS-u ovisno o potezu koji se radi pokazivačem iznad kućica elemenata. Promjena klasa kućica vidljiva je na slici 33.



Slika 33. Prikaz animacije pri dodiru određenog elementa periodnog sustava elemenata

6.9. Stranica „Kalkulator“

Cijeli programski kod stranice „Kalkulator“ nalazi se u prilogu, a tu će biti objašnjen.

Kao i na prethodnoj stranici „Periodni sustav“ korišteni su CSS za stil i jQuery za funkcije. U HTML dokumentu su u *<style> tagu* definirani oblici, boje, veličine i slično, dok je u *tagu <script>* definirana jQuery funkcija.

Za razliku od periodnog sustava, koji je za prikaz koristio liste, ovdje se koriste odjeljci, tj. * tag* za sve tipke kalkulatora, ekran te oblik, a pomoću klasa se poziva njihov stil (oblik, boja, font, veličina, itd.). Također je definirano da pokazivač promijeni oblik kada dođe nad tipku.

Pomoću funkcije definiramo događaje koji se odvijaju na pritisak određene tipke. Koristimo se varijablama kako bi postavili početno stanje te s njima ulazimo u if-else logiku. Ako je pritisnut neki od operatora (+, -, *, …) potrebno je zapamtiti njegovu funkciju kako bi se kasnije mogla izvršiti. No, ako je pritisnuta tipka „C“ potrebno je sve izbrisati i vratiti se na početno stanje: 0. Zatim, ako je pritisnuta tipka „=“ potrebno je izvršiti matematičku funkciju ranije unesenu u kalkulator te prikazati rješenje. Kako bi funkcija znala što je uneseno, koristimo se klasama – da znamo da li je pritisnuta brojka, znak za operaciju, znak za jednako ili znak za čišćenje.

Kada spojimo CSS, HTML i jQuery dobivamo funkcionalni kalkulator prikazan na slici 34.

Slika 34. Izgled stranice „Kalkulator“

6.10. Stranica „Računanje kvadratne jednadžbe“

Kako bi se učenicima/studentima omogućila provjera rješenja ili skratilo vrijeme računanja kvadratnih jednadžbi, kreirana je ova stranica čiji je zadatak za unesene podatke, tj. koeficijente jednadžbe pronaći rješenja.

Ponovno pomoću poznatog CSS-a uređujemo stil (izgled), dok u HTML dokumentu definiramo ulaze (eng. *input*), pomoću kojih kasnije u funkciji računamo rješenja jednadžbe.

Funkcija je izrađena pomoću jQuery tehnologije, a to je prikazano programskim kodom 16.

Programski kod 16. Funkcija za izračun kvadratne jednadžbe (jQuery)

```
1.  <script type="text/javascript">
2.      $(document).ready(function(){
3.          $("#izracunaj").click(function(){
4.              var a = $("#A").val(),
```

```
5.                  b = $("#B").val(),
6.                  c = $("#C").val(),
7.                  d=(Math.pow(b,2)-(4*a*c));
8.
9.              var x1, x2;
10.
11.             if (d<0){
12.             x1 = -b/(2*a)+"-"+Math.sqrt(Math.abs(d))/(2*a)+"i";
13.             x2 = -b/(2*a)+"+"+Math.sqrt(Math.abs(d))/(2*a)+"i";
14.         }
15.         else{
16.             x1=(-b + Math.sqrt(d)) / (2*a);
17.             x2=(-b - Math.sqrt(d)) / (2*a);
18.         }
19.             $("#r1").val(x1);
20.             $("#r2").val(x2);
21.         });
22.     });
23. </script>
```

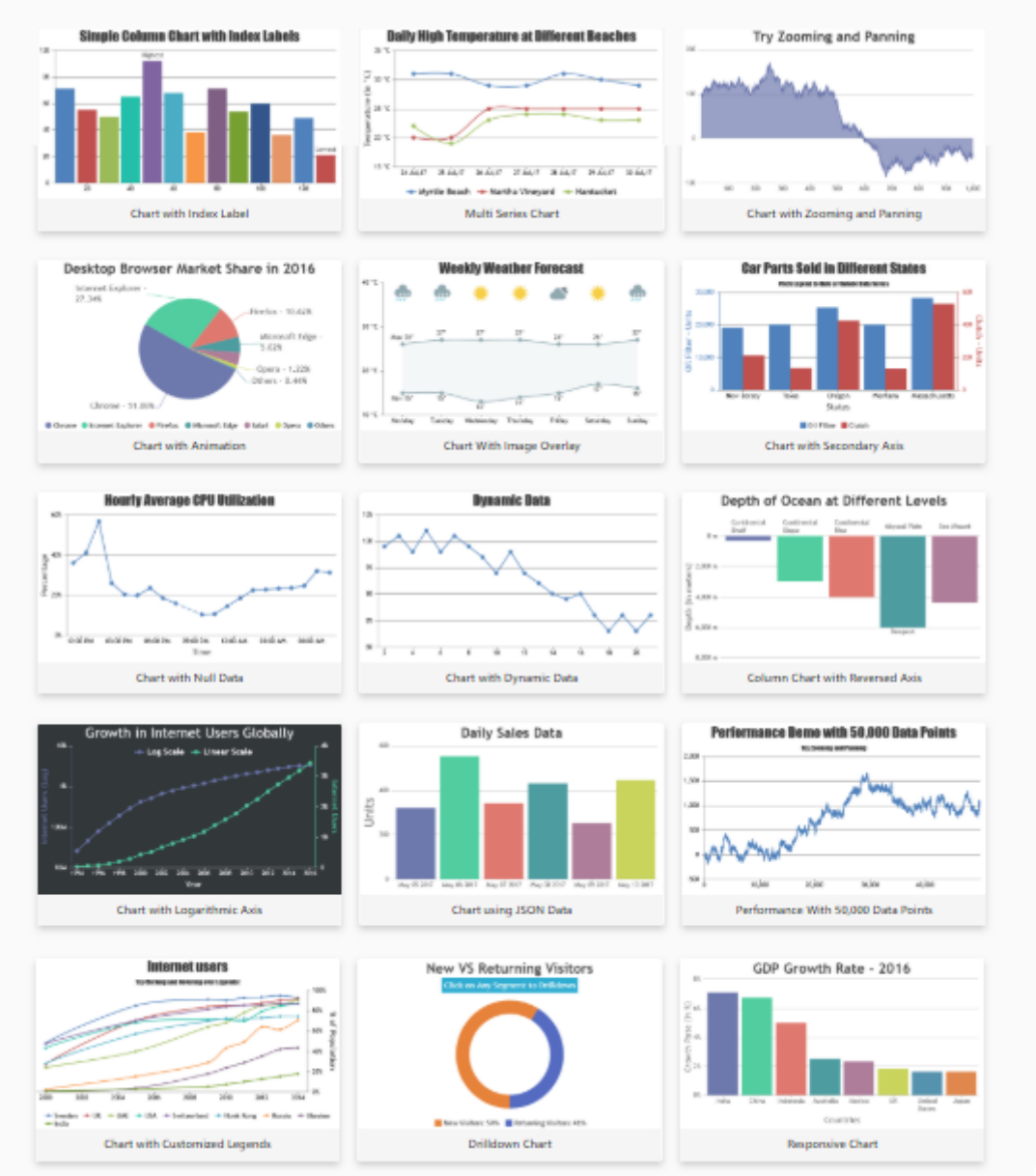
Kao što možemo vidjeti i ova funkcija se koristi *if-else* logikom. Funkcija se aktivira dodirom na gumb „Izračunaj“. U varijable a, b i c spremaju se unesene vrijednosti koeficijenata jednadžbe te s u d sprema vrijednost determinante koja se tada računa za gore navedene koeficijente. Zatim se stvaraju varijable rješenja: x_1 i x_2. Ako je determinanta manja od nule znači da imamo kompleksne brojeve te se rješenja računaju i zapisuju u tom obliku, a ako je veća od nula rješenja se računaju po poznatoj formuli. Kada su izračunata rješenja, ona se prikazuju u ranije određenom prostoru.

Izgled stranice i njen ispravan rad (rješavanje kvadratne jednadžbe) vidljiv je na slici 35.

Slika 35. Izgled stranice „Kvadratna jednadžba“

6.11. Stranica „Europa u brojevims“

Kako bi se statistički podaci prikazali na jednostavniji i zabavniji način (da nisu samo brojevi), na web stranice je danas veoma lako postaviti grafičke prikaze. To je ovdje i učinjeno pomoću tehnologije canvasJS – HTML5 grafičke biblioteke. Različite grafičke prikaze canvasJS-a možemo vidjeti na slici 36.



Slika 36. Mogućnosti grafova canvasJS-a

Mogućnosti za grafičke prikaze su neograničene:

- mogući su stupičasti, trakasti, linijski, tortni i još mnogo različitih vrsta grafičkog prikaza
- podaci mogu biti dinamični

- grafovi se mogu zumirati i pomicati
- broj podataka može biti i veći, do 50 000
- više varijabla može biti prikazano na jednom grafu
- postoje animacije za grafičke prikaze
- i još mnogo različitih

Za stranicu od geografije odabrana su dva grafa: udio regija u površini Europe – tortasti grafički prikaz te udio broja stanovnika po regijama Europe – stupičasti grafički prikaz.

Kako bi se grafovi prikazali u HTML datoteku upisujemo programski kod 17.

Programski kod 17. Kod za pozivanje grafova pomoću id-a

```
<div id="chartContainer" style="height: 370px; width: 100%;"></div><br><br>
<div id="chartContainer2" style="height: 370px; width: 70%;"></div><br><br>
```

Kao što možemo vidjeti, grafovi se pozivaju pomoću id-a. a pozivamo ih pomoću funkcija koje možemo vidjeti u programskom kodu 18.

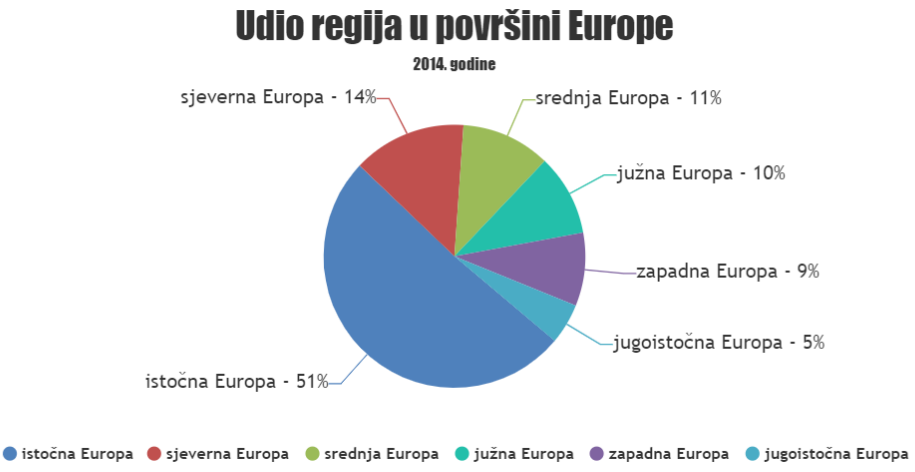
Programski kod 18. Uređivanje grafova

```
<script>
window.onload = function () {
var options = {
    title: {
        text: "Udio regija u površini Europe"
    },
    subtitles: [{
        text: "2014. godine"
   }],
    animationEnabled: true,
    data: [{
        type: "pie",
        startAngle: 40,
        toolTipContent: "<b>{label}</b>: {y}%",
        showInLegend: "true",
        legendText: "{label}",
        indexLabelFontSize: 16,
        indexLabel: "{label} - {y}%",
        dataPoints: [
            { y: 51, label: "istočna Europa" },
            { y: 14, label: "sjeverna Europa" },
            { y: 11, label: "srednja Europa" },
            { y: 10, label: "južna Europa" },
            { y: 9, label: "zapadna Europa" },
            { y: 5, label: "jugoistočna Europa" },
        ]
    }]
};
    var options2 = {
    animationEnabled: true,
    title: {
        text: "Udio broja stanovnika Europe po regijama"
    },
    axisY: {
        title: "Udio stanovnika",
```

```
36.         suffix: "%",
37.         includeZero: false
38.     },
39.     axisX: {
40.         title: "Regija"
41.     },
42.     data: [{
43.         type: "column",
44.         yValueFormatString: "#,##0.0#"%"",
45.         dataPoints: [
46.             { label: "istočna Europa", y: 23.3 },
47.             { label: "sjeverna Europa", y: 4.6 },
48.             { label: "srednja Europa", y: 24.1 },
49.             { label: "južna Europa", y: 18.9 },
50.             { label: "zapadna Europa", y: 22.8 },
51.             { label: "jugoistočna Europa", y: 6.3 }
52.         ]
53.     }]
54. };
55. $("#chartContainer").CanvasJSChart(options);
56. $("#chartContainer2").CanvasJSChart(options2);
57. }
58. </script>
```

U ovom programskom kodu prikazana je funkcija za crtanje dva grafa.

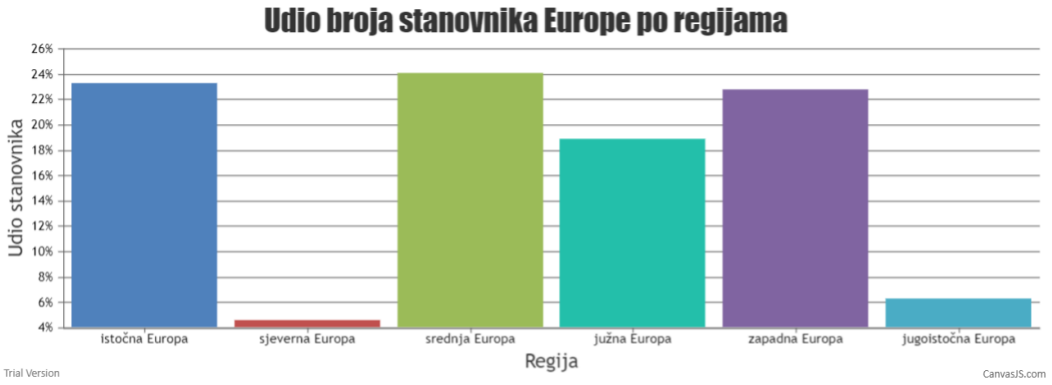
Od 4 do 29 linije imamo prvi graf. Njemu u prvim linijama tog koda definiramo naslov i podnaslov te uključujemo animaciju tijekom prikaza. Zatim se u linijama od 12 do 27 bavimo podacima samog grafa, njihovom prikazu, te podacima koje iscrtavamo. Nadalje, od 13. linije definiramo vrstu grafičkog prikaza, početni kut iscrtavanja grafa, način ispisa podataka kada pokazivačem dođemo iznad njih, tekst legende, veličinu fonta, oznaku indeksa (u ovom slučaju %) te podatke (imena regija i njihove udijele u površini Europe). Izgled tortnog grafa na web stranici vidljiv je na slici 37.



Slika 37. Tortni grafički prikaz podataka

Od linije 30 do linije 55 imamo drugi graf. U prvim linijama, kao i kod prethodnog grafa definiramo naslov i podnaslov te nazive osi i njihova svojstva. Od linije 43 definiramo vrstu

grafičkog prikaza, način na koji se prikazuju podaci te vrijednosti koje se iscrtavaju na grafu (udio broja stanovnika Europe po regijama). Izgled stupičastog grafa na web stranici vidljiv je na slici 37.



Slika 38. Stupičasti grafički prikaz podataka

6.12. Stranica „Karta“

Kako većina geografije koju učimo u školi sastoji se od karata i učenja stvari u atlasima, logično je da se i to stavi u multimedijalni oblik za web stranicu. Zahvaljujući Google-u danas na svoje stranice možemo postaviti njegovu interaktivnu kartu: „Google maps“. Također je možemo prilagoditi svojim željama. Način za pozivanje karte i funkcije za kartu možemo vidjeti u programskom kodu 19.

Programski kod 19. Pozivanje i uređivanje interaktivne karte (Google Maps)

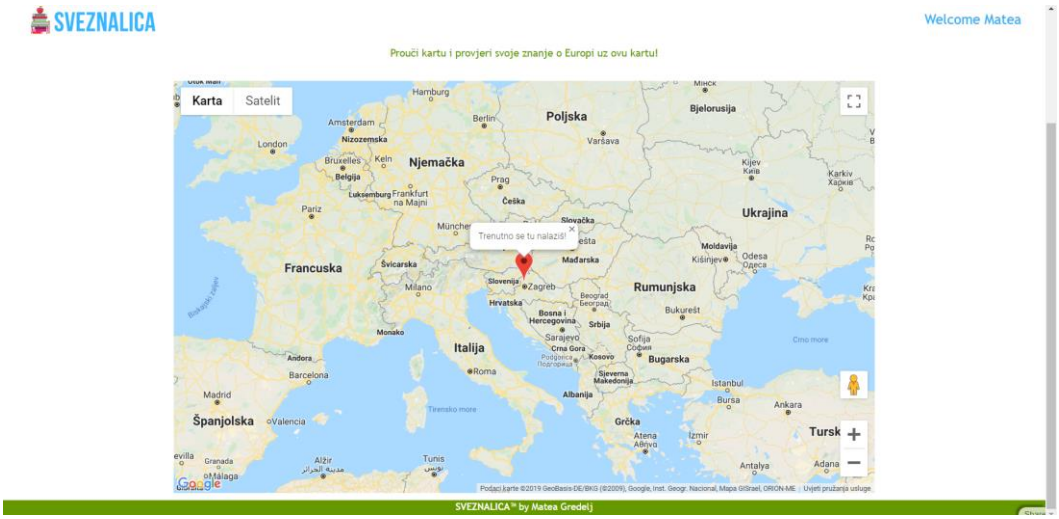
```
<div id="map" ></div>

<script>
    function myMap() {
        var myCenter = new google.maps.LatLng(45.7963138,15.9714571);
        var mapCanvas = document.getElementById("map");
        var mapOptions = {center: myCenter, zoom: 5};
        var map = new google.maps.Map(mapCanvas, mapOptions);
        var marker = new google.maps.Marker({
            position: myCenter,
            animation: google.maps.Animation.BOUNCE
        });
        marker.setMap(map);
        google.maps.event.addListener(marker,'click',function() {
            map.setZoom(9);
            map.setCenter(marker.getPosition());
        });
        var infowindow = new google.maps.InfoWindow({
            content: "Trenutno se tu nalaziš!"
        });
        infowindow.open(map,marker);
    }
</script>
<script async defer src="https://maps.googleapis.com/maps/api/js?key=AIzaSyBHzr5Nxe9mmyv89pPz7wdoSLUQSn9mJcE&callback=myMap"></script>
```

Prva linija koda poziva na stranicu kartu pomoću id-a. Također postoji i CSS datoteka koja definira širinu i poziciju karte na stranici, no kako je to već viđen kod nije ga potrebno ponovno objašnjavati.

U prvoj skripti nalazi se funkcija koja stvara kartu i prilagođava pomoću varijabla našim potrebama. Redom nam varijable za kartu definiraju središte karte (Ivana Lučića 5), id pomoću kojeg se karta prikazuje na stranici, da je gore određeno središte u centru karte i povećano 5 puta. Iduće što je kroz varijablu definirano je marker koji se nalazi u prethodno definirano središtu te da je isti animiran – skakuće gore dolje po točki središta. Pritiskom na taj marker mapa se povećava 9 puta, te pozicionira i dalje u naše središte. Iznad markera se nalazi i prozor na kojemu stoji natpis: „Trenutno se tu nalaziš!“. Time smo prilagodili stranicu svojim potrebama.

U drugoj skripti imamo link pomoću kojeg pozivamo sve ostale varijable mape, no kako bismo to mogli, moramo imati Google račun i od njega zatražiti ključ pomoću kojeg dobivamo pristup Google Maps-u.

Izgled naše karte na stranici prikazan je slikom 39.



Slika 39. Stranica „Karta“ sa prikazom Google Maps-a

6.13. Realizacija kolegija sa znanjem i provjerom znanja

Dva kolegija koja su u potpunosti realizirana su „Biologija 7“ za 7. razred osnovne škole i „Elektrotehnika“ za Fakultet strojarstva i brodogradnje. Kako se svaki kolegij sastoji od dva dijela – znanje i provjeru znanja, za svaki dio su potrebne i baze podataka koje sadrže predavanja/vježbe i zadatke za provjeru stečenog znanja.

6.13.1. Baze podataka

Za svaki predmet/kolegij postoje po dvije baze podataka. Njih stvaramo u modelu u *db.py* kao što je to već ranije objašnjeno.

Način stvaranja baze podataka za biologiju prikazan je programskim kodom 20.

Programski kod 20. Definiranje baze podataka „biologija“ i „bioquiz“

```
1.  db.define_table('biologija',
2.                  Field('poglavlje', required=True, requires=IS_IN_SET(['1.1 Biološka
    raznolikost', '1.2 Pojava života na Zemlji', '1.3 Darwinova teorija evolucije', '1.4
     Dokazi evolucije i prirodna selekcija', '1.5 Raznolikost živog svijeta', '2.1 Najje
    dnostavniji oblici života', '2.2 Stanica', '2.3 Stanične diobe', '3.1 Praživotinje i
     alge', '3.2 Gljive i lišajevi', '4.1 Podjela carstva životinja', '4.2 Spužve i žarn
    jaci', '4.3 Plošnjaci i oblići', '4.4 Mekušci', '4.5 Kolutićavci', '4.6 Člankonošci'
    , '4.7 Bodljikaši', '5.1 Svitkovci', '5.2 Ribe i vodozemci', '5.3 Gmazovi', '5.4 Pti
    ce', '5.5 Sisavci', '6.1 Razvoj biljaka', '6.2 Mahovine i papratnjače', '6.3 Golosje
    menjače', '6.4 Kritosjemenjače'])),
3.                  Field('naslov', 'string', required=True, requires=IS_NOT_EMPTY()),

4.                  Field('tekst', 'text', required=True, requires=IS_NOT_EMPTY()),
5.                  Field('slika', 'upload'),
6.                  Field('naslov_slike', 'string'),
7.                  Field('video', 'upload'),
8.                  Field('naslov_videa', 'string'),
9.                  )
10. db.define_table('bioquiz',
11.                 Field('poglavlje', 'string', required=True, requires=IS_IN_SET(['1.
    Pojava i razvoj života na Zemlji', '2. Najjednostavniji oblici života i stanica', '3
    . Protoktisti i gljive', '4. Beskralježnjaci', '5. Svitkovci', '6. Biljke'])),
12.                 Field('pitanje', 'text', required=True, requires=IS_NOT_EMPTY()),

13.                 Field('odgovor_a', 'string', required=True, unique=True, requires=IS
    _NOT_EMPTY()),
14.                 Field('odgovor_b', 'string', required=True, unique=True, requires=IS
    _NOT_EMPTY()),
15.                 Field('odgovor_c', 'string', required=True, unique=True, requires=IS
    _NOT_EMPTY()),
16.                 Field('odgovor_d', 'string', required=True, unique=True, requires=IS
    _NOT_EMPTY()),
17.                 Field('tocno', 'string', required=True, requires=IS_IN_SET(['a', 'b'
    , 'c', 'd'])),
18.                 )
```

Tablicu baze podataka stvaramo pomoću funkcije:

```
db.define_table().
```

Prva baza podataka (linije 1-9) zove se „biologija“ i sastoji se od 7 polja (eng. *field*): poglavlje, naslov, tekst, slika, naslov slike, video i naslov videa. Prvo moramo imenovati polje, definirati tip te upisati ako polje ima neke zahtjeve. Prvo polje ima naslov „poglavlje“, te mu je tip „*string*“. Također ima zahtjev da nešto mora biti upisano/odabrano:

```
required=True,
```

ali i da mora biti u nalazi u setu već unaprijed definiranih poglavlja:

```
requires=IS_IN_SET([…]).
```

Iduće polje, „naslov“, je također tipa „*string*“ i ima zahtjev da bude upisano te zahtjev da ne smije ostati prazan *string*:

```
requires=IS_NOT_EMPTY().
```

Treće polje, „tekst“, tipa je „*text*“ i ima iste zahtjeve kao i prethodno polje.

Iduća četiri polja su dva polja za unos slike/videa tipa „*upload*“ (jer ih prenosimo sa računala u bazu) te njihovi naslovi.

Druga baza podataka (linije 11-19) sastoji se od pitanja za kviz. Prvo je potrebno odabrati poglavlje za koje je pitanje, zatim upisati pitanje, upisati četiri moguća odgovora, a na kraju iz seta odabrati koji od ponuđenih odgovora je točan. Kako ne bi došlo do greške da se ponude dva ista odgovora, ovdje imamo zahtjev za jedinstvenost:

```
unique=True.
```

Kako imamo definirane baze podataka za biologiju, potrebno je napraviti i baze podataka za elektrotehniku. Rade se na isti način kao i baze za biologiju što možemo vidjeti na programskom kodu 21.

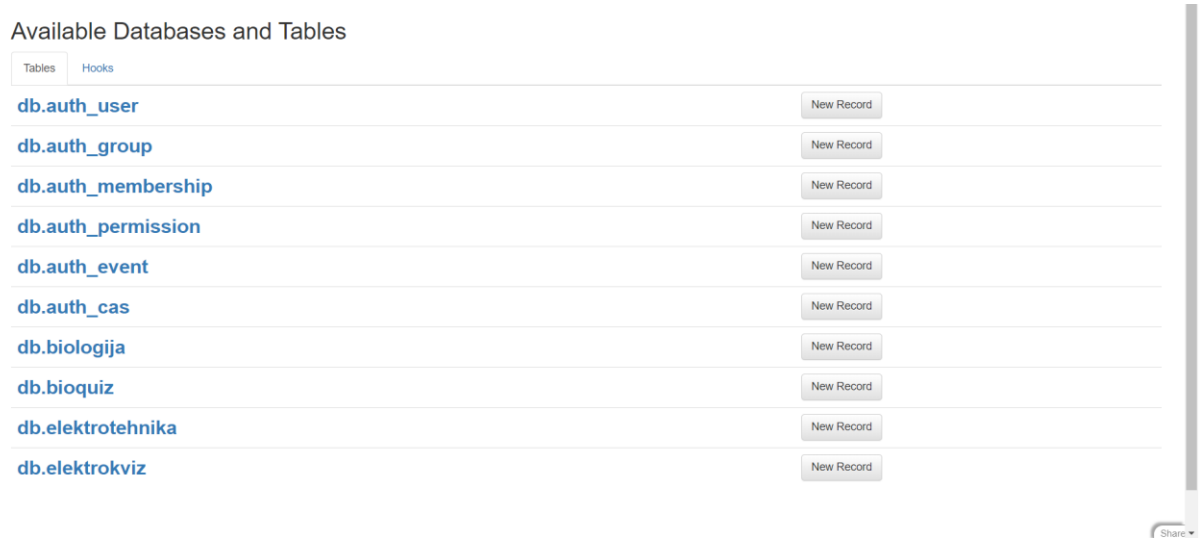
Programski kod 21. Definiranje baze podataka „elektrotehnika“ i „elektrokviz“

```
1.  db.define_table('elektrotehnika',
2.                  Field('vrsta', required=True, requires=IS_IN_SET(['predavanja', 'vje
    žbe'])),
3.                  Field('poglavlje', required=True, requires=IS_IN_SET(['1. Elektricit
    et i magnetizam', '2. Električne mreže i zakoni', '3. Istosmjerna i izmjenična anali
    za', '4. Prijelazne pojave', '5. Frekvencijska analiza, mrežne funkcije', '6. Elektr
    ični sustavi', '7. Trofazni sustav', '8. Elektronika', '9. Dioda, trijak', '10. Tran
    zistor', '11. Računsko pojačalo', '12. Elektromagnetizam', '13. Električni strojevi'
    , '14. Generatori', '15. Motori'])),
4.                  Field('naslov', 'string', required=True, requires=IS_NOT_EMPTY()),

5.                  Field('tekst', 'text', required=True, requires=IS_NOT_EMPTY()),
6.                  Field('slika', 'upload'),
7.                  Field('naslov_slike', 'string'),
8.                  Field('video', 'upload'),
9.                  Field('naslov_videa', 'string'),
10.                )
11.
12. db.define_table('elektrokviz',
13.                 Field('pitanje', 'text', required=True, requires=IS_NOT_EMPTY()),

14.                 Field('slika', 'upload', required=True, requires=IS_NOT_EMPTY()),

15.                 Field('odgovor_a', 'string', required=True, unique=True, requires=IS
    _NOT_EMPTY()),
16.                 Field('odgovor_b', 'string', required=True, unique=True, requires=IS
    _NOT_EMPTY()),
17.                 Field('odgovor_c', 'string', required=True, unique=True, requires=IS
    _NOT_EMPTY()),
18.                 Field('odgovor_d', 'string', required=True, unique=True, requires=IS
    _NOT_EMPTY()),
19.                 Field('tocno', 'string', required=True, requires=IS_IN_SET(['a', 'b'
    , 'c', 'd'])),
20.                )
```

Jedina razlika od baze podataka za biologiju je ta da imamo jedno polje više kod baze „elektrotehnika“ – polje u kojem određujemo jesu li ovo vježbe ili se radi o predavanju.

Ukratko, kreirali smo baze podataka u koje unosimo znanje (poglavlje, naslov, tekst, slika/video) i podatke za provjeru tog znanja (pitanje, mogući odgovori, točan odgovor).

Kao administrator stranice, web2py nam omogućuje da pristupimo tim bazama podataka i uređujemo ih na jednostavan način, što možemo vidjeti na slici 40.



Slika 40. Sučelje za upravljanje bazama podataka

Uz navedene zahtjeve postoji još mnogštvo drugih. Sve ovisi o potrebama za bazu podataka. Neki od ostalih mogućih zahtjeva su:

- =IS_IN_DB()
- =IS_NOT_IN_DB()
- =IS_INT_IN_RANGE()
- =IS_ALPHANUMERIC()
- =IS_MATCH()
- =IS UPPER()
- itd.

Kako postoje različite mogućnosti za zahtjeve, tako postoje i različiti tipovi polja baze podataka. Neki od mogućih tipova su:

- boolean
- integer
- blob
- double

- time
- password
- json

6.13.2. Znanje: predavanja i vježbe

Kao što je u ranijem poglavlju opisano, početna stranica za školske predmete sastoji se od prezentacije i popisa poglavlja pa tako je i u slučaju i nama odabranog predmeta (biologija). Također je rečeno da su to poveznice za nove stranice na kojima se nalaze predavanja, tj. dijelovi predavanja. Kako imamo bazu podataka sa predavanjima iz biologije, na tim stranicama se nalaze upravo ti podaci iz baze podataka. Kako bi prikazali način izrade te stranice nasumično je odabrano drugo poglavlje („2. Najjednostavniji oblici života i stanica“) za primjer.

Stranica je zamišljena kao tablice u kojima se nalaze podaci iz baze podataka ovisno o poglavlju i potpoglavlju. Tablica se sastoji od dva stupca: u prvom se nalazi naslov lekcije i tekst, dok je u desnom slika vezana uz predavanje.

CSS stil stranice jednak je onome za stranicu „Biologija 7“ te je većina koda objašnjena. Novitet u kodu je prikazan programskim kodom 22.

Programski kod 22. Stiliziranje tablice

```
1. table{
2.    text-align: left;
3.    background-color: #00cc00;
4.    width: 70%;
5. }
6. td, th  {
7.    padding: 15px;
8. }
9. tr:nth-child(even) {background-color: #006600;}
```

Definiramo da je tablica 70% širine stranice, da je tekst poravnan s lijevom stranom te njenu pozadinsku boju (zelenu). U zadnjoj liniji koda (9) definiramo da svaki parni red ima drugačiju boju (tamnije zelenu).

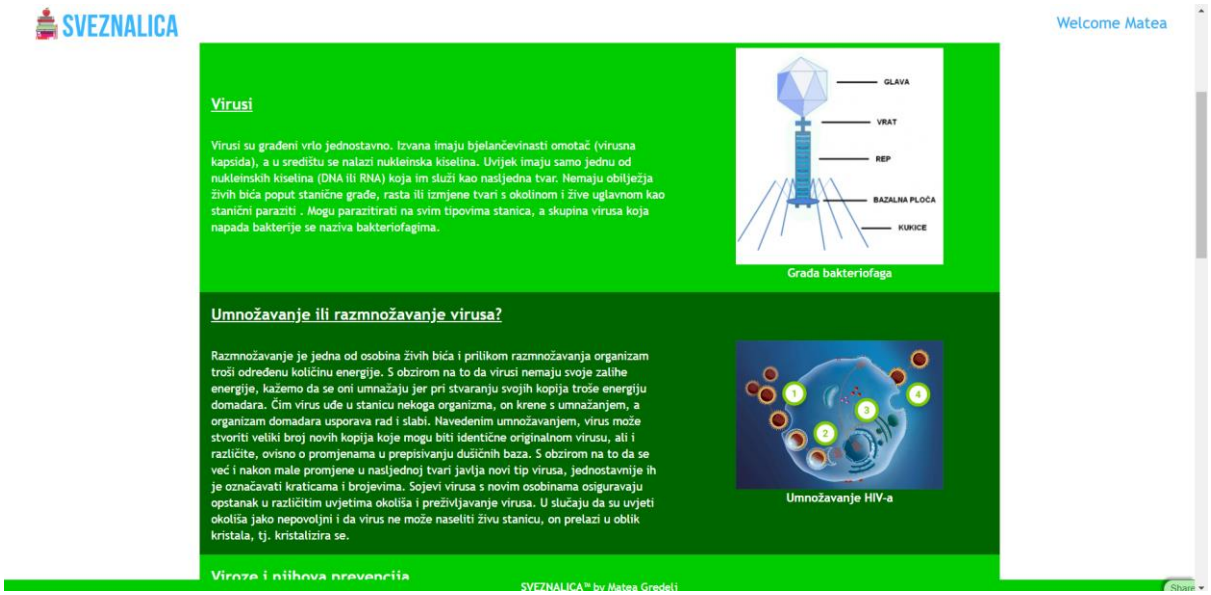
Nakon što definiramo naslove i njihov stil, potrebno je pomoću HTML koda pozvati iz baze podatke. To radimo na način prikazan programskim kodom 23.

Programski kod 23. Način dohvaćanja podataka iz baze

```
1. <table align="center">
2.     {{if db(db.biologija.poglavlje=='2.1 Najjednostavniji oblici života').isempty()==False:}}
3.         {{for redak in db((db.biologija.poglavlje=='2.1 Najjednostavniji oblici života')).select():}}
4.             <tr>
```

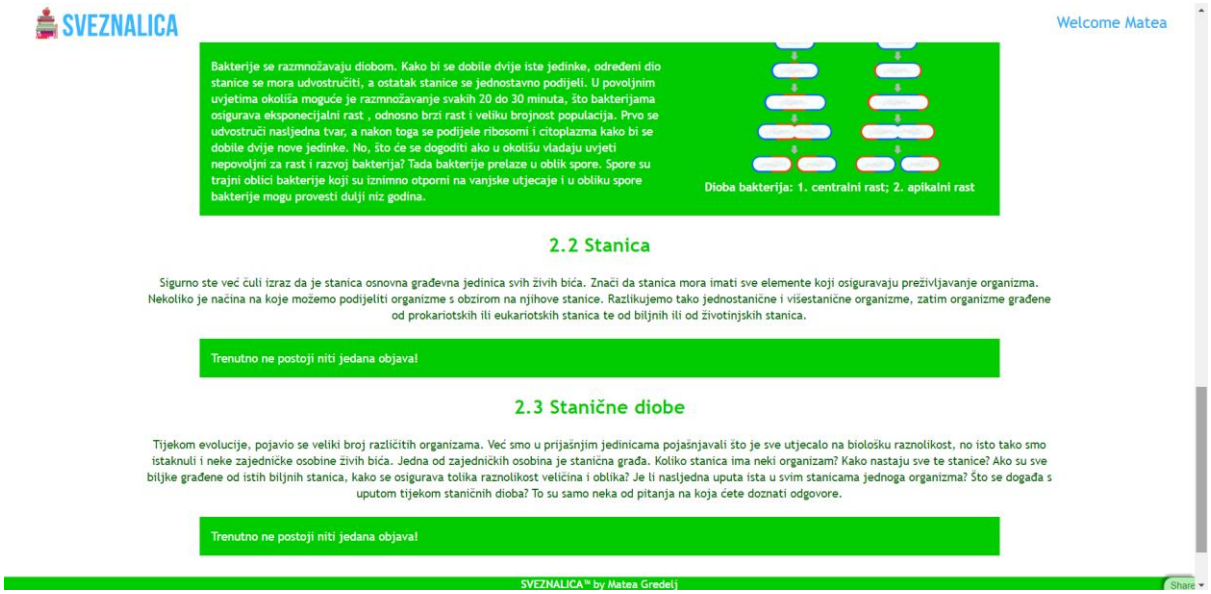
```
5.                 <td>
6.                     <div class="naslov">
7.                         <b><u>{{=redak.naslov}}</u></b><br><br>
8.                     </div>
9.                     {{=redak.tekst}}
10.                </td>
11.                <td style="text-align: center;" width=40%>
12.                    <a href="{{=URL(f='download', args=redak.slika)}}">
13.                    <img style="width:70%;  "src="{{=URL('download', args=redak.slik
    a)}}"/>
14.                    </a><br>
15.                    <b>{{=redak.naslov_slike}}</b>
16.                </td>
17.            </tr>
18.        {{pass}}
19.    {{else:}}
20.        <table align="center" style="background-
    color:#00cc00; color: white;"><tr><td>Trenutno ne postoji niti jedana objava!</td></
    tr></table>
21.    {{pass}}
22. </table>
```

U prvoj tablici se nalazi poglavlje 2.1. te se koristimo *if* logikom kako bi iz baze došli do ostalih podataka. To možemo vidjeti u drugoj liniji koda: ako podatak u bazi podataka „biologija“ i pod poljem poglavlje ima „2.1. Najjednostaviji oblici života“ odabiru se podaci iz tog retka, što je definirano u trećoj liniji. U sedmoj liniji iz baze podataka sa prethodno ispunjenim uvjetima, uzimamo naslov te u devetoj liniji tekst. U dvanaestoj i trinaestoj liniji pozivamo iz baze sliku i postavljamo mogućnost preuzimanja slike na svoje računalo pomoću klika na samu sliku. Također, ispod slike se nalazi i njen naslov, koji je kao i prethodno, iz baze podataka (15. linija). Izgled stranice vidljiv je na slici 41.



Slika 41. Tablični prikaz podataka iz baze

No, ako u bazi ne postoji traženi podatak u linijama od 19 do 21, definirano je da stranica ispiše: „Trenutno ne postoji niti jedna objava!“. To možemo vidjeti slici 42.



Slika 42. Prikaz u slučaju prazne baze podataka

Prethodno navedeni kod primjenjujemo ovisno o količini poglavlja te samo mijenjamo uvjete kod *if* logike.

Istom logikom koristimo se kako bi napravili i stranicu za elektrotehniku. Nakon što odaberemo na početnoj stranici „Fakultet strojarstva i brodogradnje“ ono nas dovodi na stranicu s mogućim kolegijima. Stranicu možemo vidjeti slici 43.

Slika 43. Stranica „Fakultet strojarstva i brodogradnje“

Kao što se može primjetiti, stranica je napravljena istom logikom kao i stranica „7. razred“ pa je nije potrebno dodatno objašnjavati.

Pritiskom na gumb „Elektrotehnika“ dolazimo na stranicu na kojoj se nalaze predavanja i vježbe – znanje. Za razliku od škola, ovdje nema početne stranice sa prezentacijama, nego se

odmah dolazi na objave profesora. Programski kod za ovu stranicu jednaka jednak je kao i na prethodno objašnjenoj stranici iz biologije. *If* logika u kodu je ista, samo su promijenjeni uvjeti: da li su vježbe ili predavanja te koje poglavlje je u pitanju. Također kod stila je promijenjena boja. To sve možemo vidjeti na slici 44.

Slika 44. Stranica „Elektrotehnika“ sa podacima iz baze

6.13.3. Provjera znanja - kviz

Nakon što učenik/student nauči određeno gradivo, potrebno je provjeriti njegovo znanje, pa je napravljena stranica s kvizom. Na kraju prethodno objašnjenih stranica nalaze se poveznice na novu stranicu: stranicu s kvizom.

Kviz je zamišljen kao ispit sačinjen od određenog broja pitanja (biologija 10 i elektrotehnika 5) sa ponuđena 4 moguća odgovora (kao što je definirano u bazama podataka). Na kraju se nalazi gumb „Predaj“ nakon čijeg pritiska funkcija provjerava točnost odgovora te vraća povratnu informaciju o točnosti riješenih zadataka. Konačno učeniku/studentu navodi na koja pitanja je krivo odgovorio.

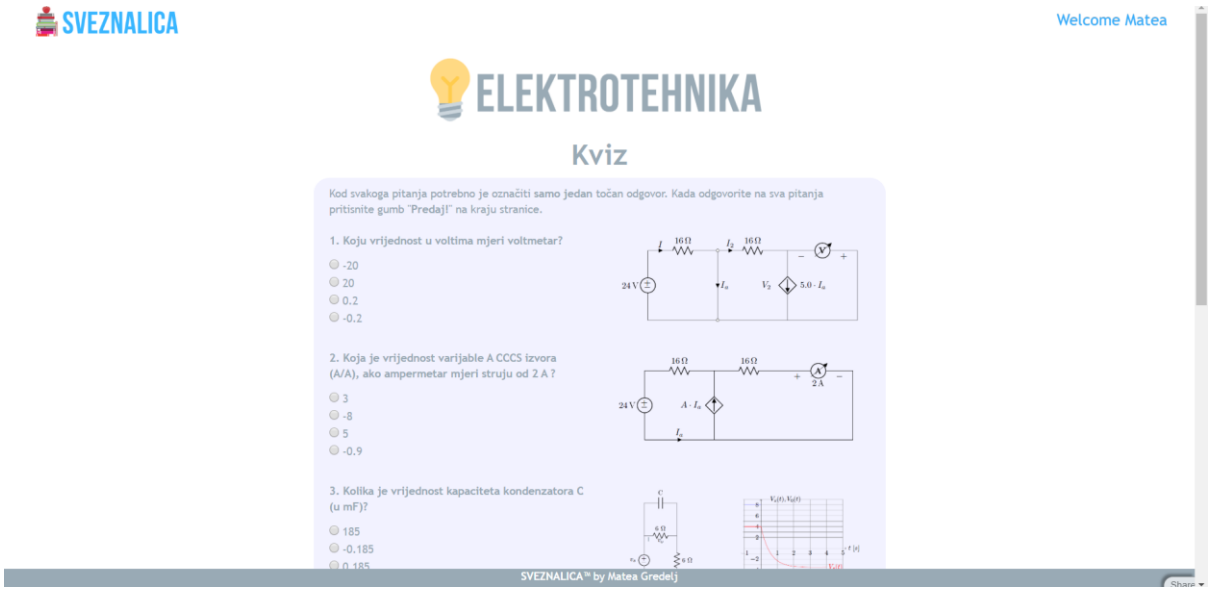
Programski kod (kraća verzija) za stranicu s kvizom iz biologije nalazi se u prilogu (zbog svoje dužine), no ovdje će biti objašnjen.

Kako bi iz baze („bioquiz“) podatke prikazali na stranici koristimo se gore objašnjenom logikom: za podatke iz te baze čije je poglavlje „2. Najjednostavniji oblici života i stanica“ uzimamo traženo. Tako prvo ispisujemo pitanje te ispod njega imamo četiri ponuđena odgovora sa mogućnošću odabira. Kroz petlju se podaci formiraju u kviz. Na kraju tog kviza nalazi se gumb „Predaj“. Nakon svakog pitanja, te na samom kraju kviza imamo prazne

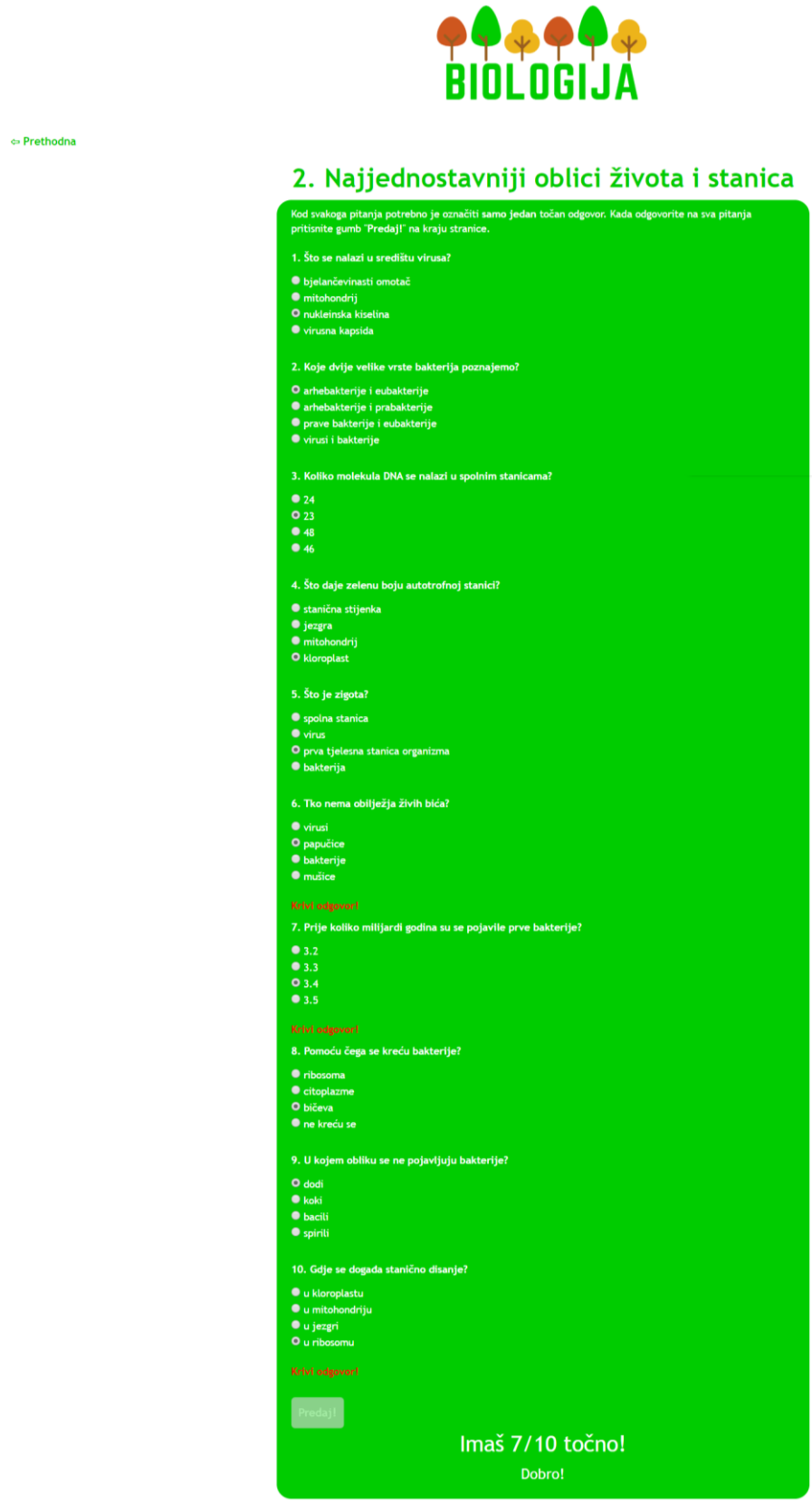
skrivene paragrafe u koje se ispisuju neke poruke nakon predaje odgovora. Nakon pritiska na gumb „Predaj“ pokreće se funkcija „*check();*“. U funkciji su definirane varijable: za svako pitanje po jedno, te varijabla točnih odgovora koja na početku iznosi 0. Nakon toga funkcija prolazi kroz pitanja i uspoređuje dani odgovor sa točnim odgovorom iz baze podataka. Za svaki točan odgovor varijabla točnosti se povećava za 1, dok u slučaju krivog odgovora u prazan paragraf ispod pitanja se ispisuje „Krivi odgovor!“. Zatim definiramo varijablu u kojoj se nalazi lista s porukama, te definiramo varijablu pomoću koje pozivamo poruke iz liste. Pomoću *if* logike slažemo ljestvicu po kojoj se dodjeljuju ocjene, tj. poruke iz liste. Na kraju funkcije definiramo radnje ispisivanja poruka u paragrafe na kraju kviza: broj točnih odgovora i njima pripadnu poruku. Također definiramo da su ti paragrafi sada vidljivi. Uz prikazivanje poruka, pritiskom na gumb također i blokiramo gumb: mijenjamo mu boju i više ga nije moguće pritisnuti.

Kviz elektrotehnike je skoro pa identičan. Razlika je u tome što su zadaci računski pa ih ima samo 5. Također, uz svako pitanje i ponuđene odgovore postoji i slika zadatka. (npr. strujnog kruga). Razmještaj podataka napravljen je pomoću Bootstrapove mreže.

Djelomični kviz iz elektrotehnike vidljiv je na slici 45, dok je cijeli ispunjeni kviz iz biologije prikazan na slici 46. Također na slici 46 je vidljiv rezultat kviza kao i označeni netočno odgovorena pitanja.



Slika 45. Prikaz kviza iz elektrotehnike na stranici



Slika 46. Prikaz ispunjenog kviz iz biologije sa označenim netočnim odgovorima

6.13.4. Stranica „Videoteka“

U bazama podataka također se nalaze polja za video i njegov naslov. Kako bi učenici/studenti mogli pristupiti videima stvaramo stranicu „Videoteka“ za svaki razred/fakultet. U njoj se nalaze tablice za svaki predmet: video zapisi i njihovi nazivi u lijevom stupcu te naslovi poglavlja u desnom.

Način prizivanja video zapisa je veoma sličan načinu prizivanja slike iz baze podataka, a on je prikazan programskim kodom 24.

Programski kod 24. Video

```
1.  <video style="width:100%" src="{{=URL('download', args=redak.video)}}" controls></vi
    deo>
```

Pomoću naredbe „*controls*“ uz video nam se pojavljuje i traka za upravljanje videom.

Ostatak programskog koda za stranicu nije potrebno dodatno objašnjavati pošto je sve već objašnjeno u ranijim poglavljima. Videoteku za 7. razred možemo vidjeti na slici47.

Slika 47. Stranica „Videoteka“

6.13.5. Znanje i profesori

Kako bi prethodna poglavlja mogla biti realizirana potrebno je imati podatke u bazi. Za to su zaduženi profesori koji bi u bazu postavljali i mijenjali podatke. Stranice za profesore se sastoje od tri dijela: stranica za unos i pregled unesenih podataka, stranica za uređivanja baze s predavanjima/vježbama i stranica za uređivanje baze s pitanjima za kviz.

Stranica za unos i pregled unesenih podataka zamišljena je kao stranica s dvije kartice (eng. *tab*): jedna za predavanja/vježbe i druga za kviz. Pritiskom na karticu predavanja/vježbe otvara se formular pomoću kojega punimo bazu podataka. On se u HTML datoteci poziva pomoću naredbe

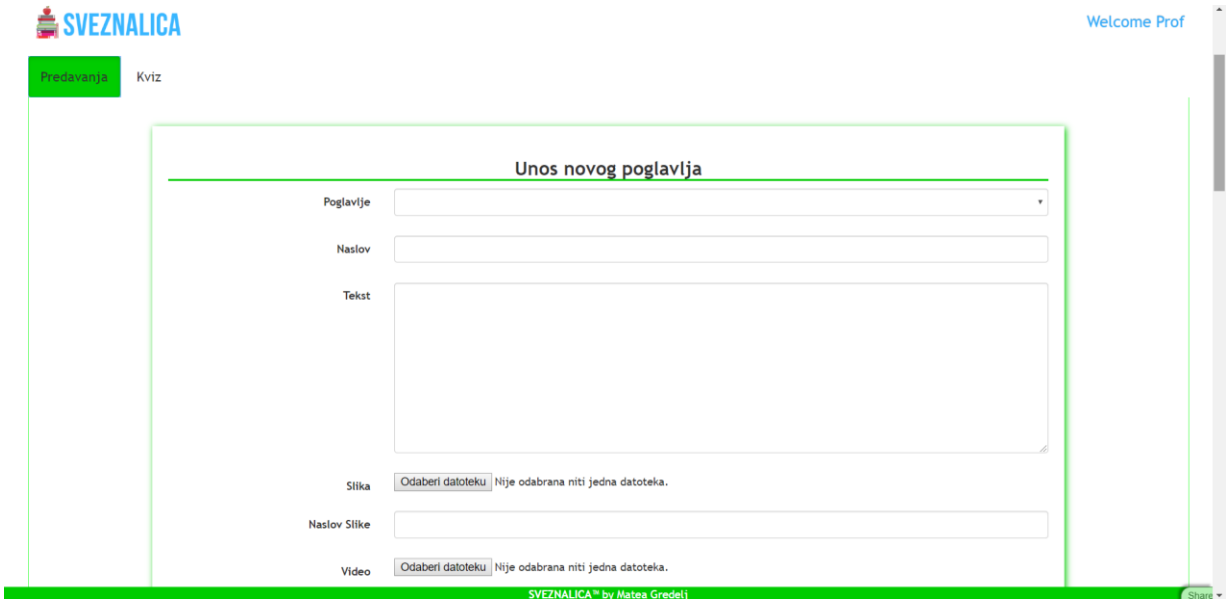
```
{{=forma}}.
```

No, kako bi se pozvala ta forma potrebno ju je definirati u upravitelju kao funkciju na način prikazan programskim kodom 25 koji se nalazi u *default.py*.

Programski kod 25. Definiranje forme

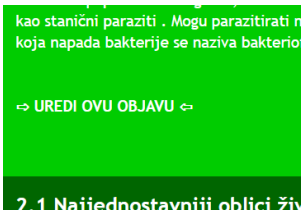
```
@auth.requires(auth.user_id==14 , requires_login=True)
def bioprof():
    response.title="Biologija"
    forma=SQLFORM(db.biologija, submit_button="Objavi")
    if forma.process(keepvalues=False).accepted:
        response.flash="Uspješno objavljeno!"
    elif forma.errors:
        response.flash="Došlo je do greške"
    return dict(formaforma=forma)
```

Kako bi samo profesor biologije mogao pristupiti ovoj stranici u prvoj liniji koda imamo to postavljeno kao zahtjev (profesor biologije u bazi korisnika ima id 14). Idući korak je definiranje forme pomoću SQLFORM-a koji automatski generira složeni obrazac za našu bazu podataka. U slučaju da su uneseni podaci ispravno uneseni oni se spremaju u bazu i stranica nam vraća poruku „Uspješno objavljeno!“, no ako dođe do greške stranica vraća poruku „Došlo je do greške!“. Izgled stranice sa otvorenom karticom „Predavanje“ možemo vidjeti na slici 48.



Slika 48. Izgled kartice za unos novog poglavlja

Nakon unesenih podataka u bazu ispod unosa nam se pojavljuju do sada već objavljena predavanja/vježbe te imamo ponuđenu mogućnost uređivanja te objave. Pritiskom na gumb za uređivanje objave (vidljiv na slici 49) otvara se nova stranica u kojoj se pojavljuje naš obrazac koji je već ispunjen podacima objave koju želimo urediti



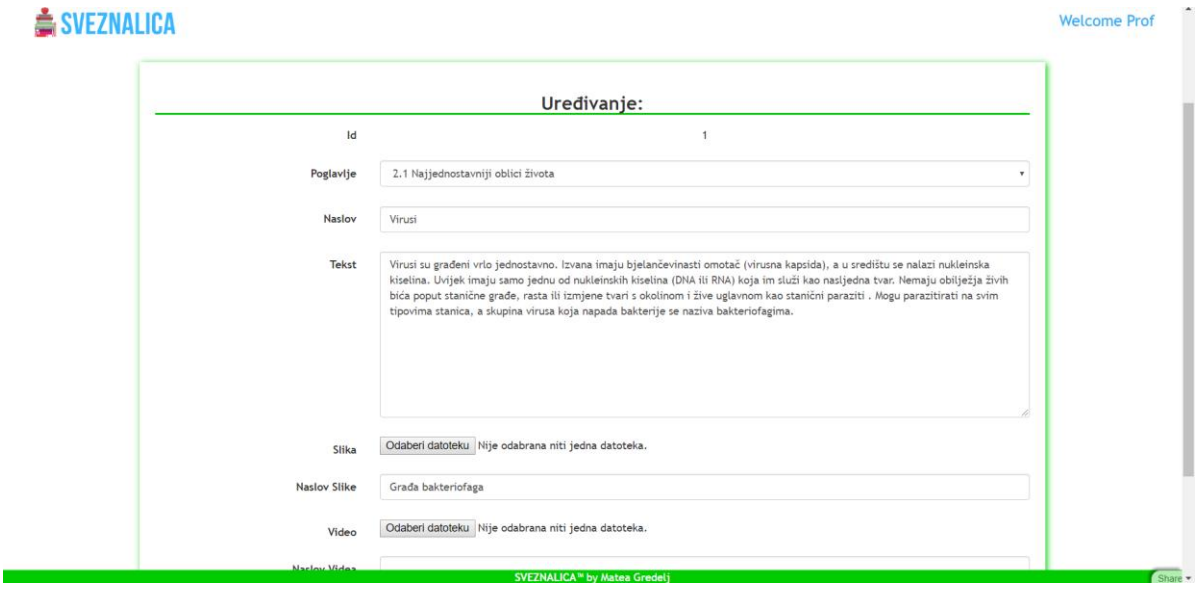
Slika 49. Gumb za uređivanje objave

Također, imamo ponuđenu mogućnost brisanja objave. To sve je također definirano funkcijom u upravitelju programskim kodom 26.

Programski kod 26. Definiranje forme sa podacima za izmjenu

```
1.  @auth.requires(auth.user_id==14 , requires_login=True)
2.  def bioprofedit():
3.      response.title = "Uređivanje objave"
4.      Id=eval(request.vars.Id)
5.      rekord=db(db.biologija.id==Id).select().first()
6.      forma=SQLFORM(db.biologija, record=rekord, submit_button="Promijeni", delete_lab
    el="Obriši", deletable=True)
7.      if forma.process().accepted:
8.          session.flash="Objava je uspješno promijenjena!"
9.          redirect(URL('bioprof'))
10.     elif forma.errors:
11.         response.flash="Došlo je do greške!"
12.     return dict(formaformaforma=forma)
```

Izgled stranice za uređivanje vidljiv je na slici 50.



Slika 50. Stranica za uređivanje objava

Pritiskom na karticu kviz otvara se tablica sa zadacima i mogućim odgovorima te poveznicom za uređenje. Programski kod ima istu logiku kao prethodno objašnjeni pa nije potrebno dodatno objašnjavati.

Za kolegij iz elektrotehnike kod je jednak ovome uz minimalne promjene u stilu (CSS) i nazivu iz baze podataka.

7. ZAKLJUČAK

Cilj ovog diplomskog rada bio je prikazati proces izrade web stranice koristeći aplikaciju web2py koja omogućuje programiranje dinamičnog web sadržaja pomoću programskog jezika Python. Web stranica prikazuje jednostavan način mrežnog upravljanja znanjem – baze podataka sa znanjem povezane sa web stranicama kako bi se to znanje moglo prenositi i provjeravati od strane profesora. Funkcija stranice nije dovoljna, potrebno ju je stilizirati te učiniti interaktivnom. Kako bi se to postiglo koriste se tehnologije poput CSS-a, JavaScript-a, Bootstrap-a te jQuery-a. Aplikacija web2py, kao i korištene tehnologije, razvijaju se na dnevnoj bazi stoga se potrebno konstantno educirati i pratiti novine iz tog područja.

U konkretnom primjeru prikazana je izrada web aplikacije „Sveznalica“ koja je predviđena za korištenje u obrazovnim ustanovama kao što su škole, odnosno fakulteti. Neke od mnogih prednosti ovog načina prenošenja znanja je interesantnije učenje za učenika/studenta koje je prilagođeno individui zbog mogućnosti kreiranja vlastitog tempa učenja, odnosno vraćanja na pojedine lekcije ako je to potrebno. Zatim, daju se korisniku otvorene ruke za samostalno istraživanje i otkrivanje novih znanja, prezentiranih kroz interaktivne i dinamične digitalne materijale te također brza povratna informacija o stupnju savladanosti gradiva.

Kako su mogućnosti danas dostupnih tehnologija izrazito velike, aplikaciju razvijenu u radu moguće je dodatno unaprijediti na razne načine. Neki od njih su na primjer dodavanje vremenskog ograničenja ispitima znanja, korištenje drugačijeg, kompleksnijeg sustava bodovanja, izrada statističkih grafova uspješnosti učenika/studenata za određeni ispit i slično tome. Također, sadržaj predavanja moguće je unaprijediti i dodavanjem na primjer zvučnih zapisa, interaktivnih mozgalica, simulacija i itd. Ovo su samo neke od potencijalnih mogućnosti poboljšanja aplikacije u trenutku izrade ovog rada, no uzevši u obzir činjenicu da se korištene tehnologije razvijaju svakodnevno, već sutra će zasigurno biti moguće činiti s njima nešto što danas još uvijek nije. Možemo tako samo nagađati što li će sve biti moguće za godinu, dvije ili deset.

LITERATURA

[1] Di Pierro, M.: web2py: Complete Reference Manual, 5th edition (2013.). Dostupno na: https://mdipierro.github.io/web2py/web2py_manual_5th.pdf. Zadnje pristupljeno 12. studenog 2019.

[2] Wikipedia: Model-view-controller (2015.). Dostupno na: https://hr.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller. Zadnje pristupljeno 01. studenog 2019.

[3] Wikipedia: HTML5, (2019). Dostupno na: https://en.wikipedia.org/wiki/HTML5. Zadnje pristupljeno 01. studenog 2019.

[4] Gasston, P.: Moderni Web. Zagreb: Dobar plan (2013.)

[5] W3schools.com (2019.). Dostupno na: https://www.w3schools.com/. Zadnje pristupljeno 15. studenog 2019.

[6] Wikipedia: JavaScript, (2019). Dostupno na: https://en.wikipedia.org/wiki/JavaScript. Zadnje pristupljeno 13. studenog 2019.

[7] Wikipedia: jQuery, (2019). Dostupno na: https://en.wikipedia.org/wiki/JQuery. Zadnje pristupljeno 03. studenog 2019.

[8] Wikipedia: Bootstrap (front-end framework), (2019). Dostupno na: https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework). Zadnje pristupljeno 04. studenog 2019.

PRILOZI

I. CD-R disc

II. Programski kod: CSS grafike

III. Programski kod: Sadržaj predmeta prikazan pomoću liste

IV. Programski kod: Periodni sustav

V. Programski kod: Kalkulator

VI. Programski kod: Kviz iz biologije

Prilog II

Programski kod: CSS grafike

```
a.button {
  position: relative;
  display: inline-block;
  width: 25.6em;
  height: 6.3em;
  overflow: hidden;
  vertical-align: middle;
  text-align: left;
  text-decoration: none;
  line-height: 3em;
  font-size: 1rem;
  color: #fff;
  cursor: pointer;
  -webkit-transform: translateZ(0);
          transform: translateZ(0);
  transition: -webkit-transform 0.5s;
  transition: transform 0.5s;
  transition: transform 0.5s, -webkit-transform 0.5s;
}
a.button svg {
  z-index: 1;
  position: absolute;
  top: 0;
  left: 0;
}
a.button svg g line,
a.button svg g polyline,
a.button svg g circle {
  fill: none;
  stroke: #33adff;
}
a.button svg g line,
a.button svg g polyline {
  stroke-miterlimit: 10;
}
a.button svg g circle {
  display: block;
  stroke-linecap: round;
  stroke-linejoin: round;
  stroke-width: 0.066em;
  stroke-dasharray: 200;
  stroke-dashoffset: 0;
  transition: stroke-dashoffset ease-out 0.5s;
}
a.button font {
  z-index: 2;
  position: relative;
  display: inline-block;
  text-transform: uppercase;
  color: #33adff;
  font-family: Trebuchet MS;
  font-size: 1.5em;
}
a.button:hover {
  -webkit-transform: translateZ(0) translateX(0.5em);
          transform: translateZ(0) translateX(0.5em);
}
a.button:hover svg g circle {
  stroke-dashoffset: 200;
}
```

Prilog III

Programski kod: Sadržaj predmeta prikazan pomoću liste

```
<div class="row">
    <ol1>
        <div class="col-sm-4">
            <a href="ups"><li1><b>Kemija - prirodna znanost:</b>
                <ol1>
                    <li1>Kemija - prirodna znanost</li1>
                    <li1>Pokus ili eksperiment</li1>
                    <li1>Kemijsko posuđe i pribor</li1>
                    <li1>Mjere opreza i zaštite na radu</li1>
                    <li1>Tvari</li>
                    <li1>Agregacijska stanja tvar</li1>
                    <li1>Svojstva tvari</li1>
                    <li1>Fizikalne i kemijske promjene</li1>
                    <li1>Biološko djelovanje tvari</li1>
                </ol1>
            </li1></a><br>
            <a href="ups"><li1><b>Vrste tvari:</b>
                <ol1>
                    <li1>Elementarne tvari i kemijski spojevi</li>
                    <li1>Vrste smjesa</li1>
                    <li1>Otopine</li1>
                    <li1>Kisele i lužnate otopine</li1>
                    <li1>Odjeljivanje sastojaka iz smjese</li1>
                    <li1>Iskazivanje sastava smjesa</li1>
                </ol1>
            </li1></a>
        </div>
        <div class="col-sm-4">
            <a href="ups"><li1><b>Zrak, voda, vodik:</b>
                <ol1>
                    <li1>Zemljina atmosfera</li1>
                    <li1>Kisik i ozon</li1>
                    <li1>Postanak Zemljine atmosfere</li1>
                    <li1>Ostali plinoviti sastojci zraka</li1>
                    <li1>Voda u prirodi</li1>
                    <li1>SVojstva kemijski čiste vode</li1>
                    <li1>Voda - temelj života</li1>
                    <li1>Vodik</li1>
                </ol1>
            </li1></a><br>
            <a href="ups"><li1><b>Građa tvari:</b>
                <ol1>
                    <li1>Građa atoma</li1>
                    <li1>Kemijski simboli i označavanje atoma</li1>
                    <li1>Periodni sustav elemenata</li1>
                    <li1>Relativna atomaska masa</li1>
                    <li1>Građa atoma i označavanje elementarnih tvari</li1>
                    <li1>Ioni i ionski spojevi</li1>
                    <li1>Valencije elemenata, imena i formule spojeva</li1>
                    <li1>Relativna molekulska masa</li1>
                </ol1>
            </li1></a>
        </div>
        <div class="col-sm-4">
            <a href="ups"><li1><b>Kemijske reakcije i kemijski elementi:</b>
                <ol1>
                    <li1>Kemijske reakcije</li1>
                    <li1>Zakon o očuvanju mase</li1>
                    <li1>Iskustveni zakon i atomska teorija</li1>
                    <li1>Kemijske jednadžbe</li1>
                    <li1>Kemijske reakcije i energija</li1>
```

```
                    <li1>Brzina kemijske reakcije</li1>
                    <li1>Kemijski elementi</li1>
                    <li1>Povijest pojma kemijski elementi</li1>
                    <li1>Halogeni elementi</li1>
                    <li1>Alkalijski metali</li1>
                </ol1>
            </li1></a><br>
            <a href="periodnisustav"><li1><b>PERIODNI SUSTAV</b>
            </li1></a>
        </div>
    </ol1>
</div>
```

Prilog IV

Programski kod: Periodni sustav

```
<!DOCTYPE html>
<html>
<head>
    <title>Periodni sustav</title>

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></
script>

    <style>
        footer {
              background-color: #8000ff;
              text-align: center;
              color: white;
              position: fixed;
              left: 0;
              bottom: 0;
              width: 100%;
            }
        h1 {
            text-align: center;
            color: #8000ff;
            font-family: Trebuchet MS;
            font-size: 40px;
        }
        a {
            text-decoration: none;
        }
    * { box-sizing: border-box; }
        body {
            background:linear-gradient(#dbeaf8, #dbeaf8);
            color:#222;
            font-family:Trebuchet MS;
        }

        .wrapper {
            width:1100px;
            margin:30px auto;
            position:relative;
        }

        .main:after {
            clear:both;
            content:"";
            display:table;
        }

        .main li {
            width:56px;
            height:60px;
            border:1px solid rgba(0,0,0,0.3);
            box-shadow:inset 0 1px 0 rgba(255,255,255,0.4);
            position:relative;
            font-size:18px;
            float:left;
            padding:0 2px;
            margin:1px;
            cursor:pointer;
            padding-top:20px;
            overflow:hidden;
            transition:all .2s ease-in;
            text-shadow:0 1px 0 rgba(255,255,255,0.2);
```

```
        }

        .main li span {
            display:block;
            line-height:1;
            font-size:9px;
            color:black;
            padding-top:9px;
        }

        .main .empty {
            border:none;
            box-shadow:none;
            cursor:default;
        }

        .deactivate {
            opacity:0.5;
            -webkit-filter:grayscale(70%);
            filter:grayscale(70%);
        }

        .main li:before,
        .main li:after {
            content:attr(data-pos);
            position:absolute;
            top:3px;
            left:3px;
            color:black;
            font-size:8px;
            line-height:1;
        }

        .main li:after {
            content:attr(data-nb);
            left:auto;
            right:3px;
        }
        .main li:hover {
                    transform:scale(2);
                    z-index:100;
                    box-shadow: 0 0 10px rgba(0,0,0,0.3);
                }

                .main .empty:hover {
                    box-shadow:none;
                    transform:none;
                }

                .type-0 { background:#fff; }
                .type-1 { background:#dddddd; }
                .type-2 { background:#a8bffa; }
                .type-3 { background:#F8B707; }
                .type-4 { background:#f3f300; }
                .type-5 { background:#3bd93b; }
                .type-6 { background:#dd9999; }
                .type-7 { background:#4CAFFA; }
                .type-8 { background:#ffaa88; }
                .type-9 { background:#ddaacc; }

                .cat-0 { color:#222; }
                .cat-1 { color:#0000dc; }
                .cat-2 { color:#b10601; }
                .cat-3 { color:#555; }

                .legend {
```

```
                    position:absolute;
                    top:0;
                    left:20%;
                    padding:10px;
                    font-size:11px;
                    background:#f1edec;
                    border:1px solid rgba(0,0,0,0.2);
                    border-radius:15px;
                    box-shadow:
                        inset 0 1px 1px white,
                        inset 0 -5px 3px #dddcdb,
                        0 0 10px rgba(0,0,0,0.2);
                }

                .legend:after {
                    content:"";
                    display:table;
                    clear:table;
                }

                .legend ul { float:left; }

                .legend .list-2 {
                    margin-left:40px;
                    width:200px;
                }

                .legend .list-1 li { margin:7px 0; }
                .legend .list-1 li:first-of-type { margin-top: 2px; }

                .legend .list-1 span {
                    border:1px solid black;
                    display:inline-block;
                    padding:3px;
                    width:20px;
                    text-align:center;
                    height:20px;
                    margin-right:5px;
                }

                .legend .list-2 li {
                    margin:2px;
                    padding:3px;
                    float:left;
                    border:1px solid rgba(0,0,0,0.2);
                    width:48%;
                    cursor:pointer;
                }</style>
            </head>
            <body>
                <div>
                    <h1><b>PERIODNI SUSTAV</b></h1>
                </div>

                <div class="wrapper">
                    <ul class="main">


                        <li data-pos="1" data-nb="1.0079" class="type-1 cat-
    2">H<span>Vodik</span></li>
                        <li class="empty"></li>
                        <li class="empty"></li>
                        <li class="empty"></li>
                        <li class="empty"></li>
                        <li class="empty"></li>
                        <li class="empty"></li>
```

```
                        <li class="empty"></li>
                        <li class="empty"></li>
                        <li class="empty"></li>
                        <li class="empty"></li>
                        <li class="empty"></li>
                        <li class="empty"></li>
                        <li class="empty"></li>
                        <li class="empty"></li>
                        <li class="empty"></li>
                        <li data-pos="2" data-nb="4.0026" class="type-2 cat-2">He<span>Helij</span></li>

                        <li data-pos="3" data-nb="6.941" class="type-3 cat-0">Li<span>Litij</span></li>
                        <li data-pos="4" data-nb="9.0122" class="type-4 cat-0">Be<span>Berilij</span></li>
                        <li class="empty"></li>
                        <li class="empty"></li>
                        <li class="empty"></li>
                        <li class="empty"></li>
                        <li class="empty"></li>
                        <li class="empty"></li>
                        <li class="empty"></li>
                        <li class="empty"></li>
                        <li class="empty"></li>
                        <li data-pos="5" data-nb="10.811" class="type-5 cat-0">B<span>Bor</span></li>
                        <li data-pos="6" data-nb="12.011" class="type-5 cat-0">C<span>Ugljik</span></li>
                        <li data-pos="7" data-nb="14.007" class="type-5 cat-2">N<span>Dušik</span></li>
                        <li data-pos="8" data-nb="15.999" class="type-5 cat-2">O<span>Kisik</span></li>
                        <li data-pos="9" data-nb="18.998" class="type-5 cat-2">F<span>Fluor</span></li>
                        <li data-pos="10" data-nb="20.180" class="type-2 cat-2">Ne<span>Neon</span></li>

                        <li data-pos="11" data-nb="22.990" class="type-3 cat-0">Na<span>Natrij</span></li>
                        <li data-pos="12" data-nb="24.305" class="type-4 cat-0">Mg<span>Magnezij</span></li>
                        <li class="empty"></li>
                        <li class="empty"></li>
                        <li class="empty"></li>
                        <li class="empty"></li>
                        <li class="empty"></li>
                        <li class="empty"></li>
                        <li class="empty"></li>
                        <li class="empty"></li>
                        <li class="empty"></li>
                        <li class="empty"></li>
                        <li data-pos="13" data-nb="26.982" class="type-7 cat-0">Al<span>Aluminij</span></li>
                        <li data-pos="14" data-nb="28.086" class="type-5 cat-0">Si<span>Silicij</span></li>
                        <li data-pos="15" data-nb="30.974" class="type-5 cat-0">P<span>Fosfor</span></li>
                        <li data-pos="16" data-nb="32.065" class="type-5 cat-0">S<span>Sumpor</span></li>
                        <li data-pos="17" data-nb="35.453" class="type-5 cat-2">Cl<span>Klor</span></li>
                        <li data-pos="18" data-nb="39.948" class="type-2 cat-2">Ar<span>Argon</span></li>
```

```
241.
242.                         <li data-pos="19" data-nb="39.098" class="type-3 cat-
    0">K<span>Kalij</span></li>
243.                         <li data-pos="20" data-nb="40.078" class="type-4 cat-
    0">Ca<span>Kalcij</span></li>
244.                         <li data-pos="21" data-nb="44.956" class="type-6 cat-
    0">Sc<span>Skandij</span></li>
245.                         <li data-pos="22" data-nb="47.867" class="type-6 cat-
    0">Ti<span>Titanij</span></li>
246.                         <li data-pos="23" data-nb="50.942" class="type-6 cat-
    0">V<span>Vanadij</span></li>
247.                         <li data-pos="24" data-nb="51.996" class="type-6 cat-
    0">Cr<span>Krom</span></li>
248.                         <li data-pos="25" data-nb="54.938" class="type-6 cat-
    0">Mn<span>Mangan</span></li>
249.                         <li data-pos="26" data-nb="55.845" class="type-6 cat-
    0">Fe<span>Železo</span></li>
250.                         <li data-pos="27" data-nb="58.933" class="type-6 cat-
    0">Co<span>Kobalt</span></li>
251.                         <li data-pos="28" data-nb="58.693" class="type-6 cat-
    0">Ni<span>Nikal</span></li>
252.                         <li data-pos="29" data-nb="63.546" class="type-6 cat-
    0">Cu<span>Bakar</span></li>
253.                         <li data-pos="30" data-nb="65.39" class="type-7 cat-
    0">Zn<span>Cink</span></li>
254.                         <li data-pos="31" data-nb="69.723" class="type-7 cat-
    0">Ga<span>Galij</span></li>
255.                         <li data-pos="32" data-nb="72.64" class="type-7 cat-
    0">Ge<span>Germanij</span></li>
256.                         <li data-pos="33" data-nb="74.922" class="type-5 cat-
    0">As<span>Arsen</span></li>
257.                         <li data-pos="34" data-nb="78.96" class="type-5 cat-
    0">Se<span>Selenij</span></li>
258.                         <li data-pos="35" data-nb="79.904" class="type-5 cat-
    1">Br<span>Brom</span></li>
259.                         <li data-pos="36" data-nb="83.80" class="type-2 cat-
    2">Kr<span>Kripton</span></li>
260.
261.                         <li data-pos="37" data-nb="85.468" class="type-3 cat-
    0">Rb<span>Rubidij</span></li>
262.                         <li data-pos="38" data-nb="87.62" class="type-4 cat-
    0">Sr<span>Stroncij</span></li>
263.                         <li data-pos="39" data-nb="88.906" class="type-6 cat-
    0">Y<span>Iritij</span></li>
264.                         <li data-pos="40" data-nb="91.224" class="type-6 cat-
    0">Zr<span>Cirkonij</span></li>
265.                         <li data-pos="41" data-nb="92.906" class="type-6 cat-
    0">Nb<span>Niobij</span></li>
266.                         <li data-pos="42" data-nb="95.94" class="type-6 cat-
    0">Mo<span>Molibden</span></li>
267.                         <li data-pos="43" data-nb="(96)" class="type-6 cat-
    0">Tc<span>Tehnecij</span></li>
268.                         <li data-pos="44" data-nb="101.07" class="type-6 cat-
    0">Ru<span>Rutenij</span></li>
269.                         <li data-pos="45" data-nb="102.91" class="type-6 cat-
    0">Rh<span>Rodij</span></li>
270.                         <li data-pos="46" data-nb="106.42" class="type-6 cat-
    0">Pd<span>Paladij</span></li>
271.                         <li data-pos="47" data-nb="107.87" class="type-6 cat-
    0">Ag<span>Srebro</span></li>
272.                         <li data-pos="48" data-nb="112.41" class="type-7 cat-
    0">Cd<span>Kadmij</span></li>
273.                         <li data-pos="49" data-nb="114.82" class="type-7 cat-
    0">In<span>Indij</span></li>
274.                         <li data-pos="50" data-nb="118.71" class="type-7 cat-
    0">Sn<span>Kositar</span></li>
```

```
275.                        <li data-pos="51" data-nb="121.76" class="type-7 cat-
    0">Sb<span>Antimon</span></li>
276.                        <li data-pos="52" data-nb="127.60" class="type-5 cat-
    0">Te<span>Telurij</span></li>
277.                        <li data-pos="53" data-nb="126.90" class="type-5 cat-
    0">I<span>Jod</span></li>
278.                        <li data-pos="54" data-nb="131.29" class="type-2 cat-
    2">Xe<span>Ksenon</span></li>
279.
280.                        <li data-pos="55" data-nb="132.91" class="type-3 cat-
    0">Cs<span>Cezij</span></li>
281.                        <li data-pos="56" data-nb="137.33" class="type-4 cat-
    0">Ba<span>Barij</span></li>
282.                        <li data-pos="57-71" class="type-8">La-
    Lu<span>Lantanoidi</span></li>
283.                        <li data-pos="72" data-nb="178.49" class="type-6 cat-
    0">Hf<span>Hafnij</span></li>
284.                        <li data-pos="73" data-nb="180.95" class="type-6 cat-
    0">Ta<span>Tantal</span></li>
285.                        <li data-pos="74" data-nb="183.64" class="type-6 cat-
    0">W<span>Volfram</span></li>
286.                        <li data-pos="75" data-nb="186.21" class="type-6 cat-
    0">Re<span>Renij</span></li>
287.                        <li data-pos="76" data-nb="190.23" class="type-6 cat-
    0">Os<span>Osmij</span></li>
288.                        <li data-pos="77" data-nb="192.22" class="type-6 cat-
    0">Ir<span>Iridij</span></li>
289.                        <li data-pos="78" data-nb="195.08" class="type-6 cat-
    0">Pt<span>Platina</span></li>
290.                        <li data-pos="79" data-nb="196.97" class="type-6 cat-
    0">Au<span>Zlato</span></li>
291.                        <li data-pos="80" data-nb="200.59" class="type-7 cat-
    1">Hg<span>Živa</span></li>
292.                        <li data-pos="81" data-nb="204.38" class="type-7 cat-
    0">Tl<span>Talij</span></li>
293.                        <li data-pos="82" data-nb="207.2" class="type-7 cat-
    0">Pb<span>Olovo</span></li>
294.                        <li data-pos="83" data-nb="208.96" class="type-7 cat-
    0">Bi<span>Bizmut</span></li>
295.                        <li data-pos="84" data-nb="(209)" class="type-7 cat-
    0">Po<span>Polonij</span></li>
296.                        <li data-pos="85" data-nb="(210)" class="type-5 cat-
    0">At<span>Astat</span></li>
297.                        <li data-pos="86" data-nb="(222)" class="type-2 cat-
    2">Rn<span>Radon</span></li>
298.
299.                        <li data-pos="87" data-nb="(223)" class="type-3 cat-
    0">Fr<span>Francij</span></li>
300.                        <li data-pos="88" data-nb="(226)" class="type-4 cat-
    0">Ra<span>Radij</span></li>
301.                        <li data-pos="89-103" class="type-9">Ac-
    Lr<span>Aktinoidi</span></li>
302.                        <li data-pos="104" data-nb="(261)" class="type-6 cat-
    3">Rf<span>Rutherfodij</span></li>
303.                        <li data-pos="105" data-nb="(262)" class="type-6 cat-
    3">Db<span>Dubnij</span></li>
304.                        <li data-pos="106" data-nb="(266)" class="type-6 cat-
    3">Sg<span>Seaborgij</span></li>
305.                        <li data-pos="107" data-nb="(264)" class="type-6 cat-
    3">Bh<span>Bohrij</span></li>
306.                        <li data-pos="108" data-nb="(277)" class="type-6 cat-
    3">Hs<span>Hassij</span></li>
307.                        <li data-pos="109" data-nb="(268)" class="type-6 cat-
    3">Mt<span>Meitnerij</span></li>
308.                        <li data-pos="110" data-nb="(281)" class="type-6 cat-
    3">Ds<span>Darmstadtij</span></li>
```

```
                        <li data-pos="111" data-nb="(272)" class="type-6 cat-3">Rg<span>Roentgenij</span></li>
                        <li data-pos="112" data-nb="(285)" class="type-7 cat-3">Cn<span>Kopernicij</span></li>
                        <li data-pos="113" data-nb="(285)" class="type-7 cat-3">Nh<span>Nihonij</span></li>
                        <li data-pos="114" data-nb="(289)" class="type-7 cat-3">Fl<span>Flerovij</span></li>
                        <li data-pos="115" data-nb="(289)" class="type-7 cat-3">Mc<span>Moskovij</span></li>
                        <li data-pos="115" data-nb="(289)" class="type-7 cat-3">Lv<span>Livermorij</span></li>
                        <li data-pos="115" data-nb="(289)" class="type-0 cat-3">Ts<span>Tenesin</span></li>
                        <li data-pos="115" data-nb="(289)" class="type-2 cat-3">Og<span>Oganeson</span></li>
                    </ul>
                </div><br><br>
                <div>
                    <a href="kemija">
                        <span style='font-size:80px; color:#8000ff;'>⇦</span>
                    </a>
                </div>
                <footer>
                    <p style="font-family: Trebuchet MS;"><b>SVEZNALICA</b>™ by Matea Gredelj</p>
                </footer>
                <script>
                    $(function(){
                        $('li[class^="type-"]').mouseover(function(){
                            var currentClass = $(this).attr('class').split(' ')[0];
                            if(currentClass != 'empty'){
                                $('.main > li').addClass('deactivate');
                                $('.' + currentClass).removeClass('deactivate');
                            }
                        });
                        $('li[class^="cat-"]').mouseover(function(){
                            var currentClass = $(this).attr('class').split(' ')[0];
                            $('.main > li').addClass('deactivate');
                            $('.' + currentClass).removeClass('deactivate');
                        });
                        $('.main > li').mouseout(function(){
                            var currentClass = $(this).attr('class').split(' ')[0];
                            $('.main > li').removeClass('deactivate');
                        });
                    });
                </script>
            </body>
```

Prilog V

Programski kod: Kalkulator

```
<!DOCTYPE html>
<html>
<head>
    <title>Kalkulator</title>

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></
script>

    <style>
        footer {
              background-color: #cc0066;
              text-align: center;
              color: white;
              position: fixed;
              left: 0;
              bottom: 0;
              width: 100%;
            }
        h1 {
            text-align: center;
            color: #cc0066;
            font-family: Trebuchet MS;
            font-size: 40px;
        }
        a {
            text-decoration: none;
        }
        form,input { margin:0px; }
        body {
            background-color: rgb(255, 230, 240);
        }
        p {
          margin: 0;
          font-family: Trebuchet MS;
          position: relative;
          left: 5%;
          color: #cc0066;
        }
        .button {
            font-family: Trebuchet MS;
            color: #cc0066;
        }
        .calculator {
          position: relative;
          margin: 1em auto;
          padding: 1em 0;
          display: block-inline;
          width: 350px;
          background-color: #e6e6e6;
          border-radius: 25px;
          box-shadow: 5px 5px 15px 3px #4d001f;
          font-family: Trebuchet MS;
        }
        .calc-row {
          text-align: center;
        }
        .calc-row div.screen {
          font-family: Trebuchet MS;
          display: table;
          width: 85%;
          background-color: white;
```

```
          text-align: right;
          font-size: 2em;
          min-height: 1.2em;
          margin-left: 0.5em;
          padding-right: 0.5em;
          border: 1px solid #888;
          color: #333;
        }

        .calc-row div {
          text-align: center;
          display: inline-block;
          font-weight: bold;
          border: 1px solid #555;
          background-color: #eee;
          padding: 10px 0;
          margin: 7px 5px;
          border-radius: 15px;
          box-shadow: 2px 2px 1px 1px #4d001f;
          width: 50px;
        }
        </style>
    </head>
    <body>
        <div>
            <h1><b>KALKULATOR</b></h1>
        </div><br><br><br>
        <div class="calculator">
            <p><b>Kalkulator </b></p>
        <div class="calc-row">
            <div class="screen">0</div>
        </div>
        <div class="calc-row">
            <div class="button" style= "cursor:pointer">1</div>
            <div class="button" style= "cursor:pointer">2</div>
            <div class="button" style= "cursor:pointer">3</div>
            <div class="button operator" style= "cursor:pointer">+</div>
        </div>
        <div class="calc-row">
                    <div class="button" style= "cursor:pointer">4</div>
                    <div class="button" style= "cursor:pointer">5</div>
                    <div class="button" style= "cursor:pointer">6</div>
                    <div class="button operator" style= "cursor:pointer">-</div>
                </div>
                <div class="calc-row">
                    <div class="button" style= "cursor:pointer">7</div>
                    <div class="button" style= "cursor:pointer">8</div>
                    <div class="button" style= "cursor:pointer">9</div>
                    <div class="button operator" style= "cursor:pointer">*</div>
                </div>
                <div class="calc-row" >
                    <div class="button clear" style= "cursor:pointer">C</div>
                    <div class="button zero" style= "cursor:pointer">0</div>
                    <div class="button equal" style= "cursor:pointer">=</div>
                    <div class="button operator" style= "cursor:pointer">/</div>
                </div>
                </div><br><br><br><br><br><br>

                <div>
                    <a href="matematika2">
                        <span style='font-size:80px; color:#cc0066;'>⇦</span>
                    </a>
                </div>
                <footer>
                    <p1 style="font-
    family: Trebuchet MS;"><b>SVEZNALICA</b>™ by Matea Gredelj</p1>
```

```
126.                 </footer>
127.                 <script type="text/javascript">
128.                     $(document).ready(function() {
129.                         var operacija = null;
130.                         var flag = false;
131.                         $('.button').on('click', function(evt) {
132.
133.                             var tekst = $(this).text();
134.
135.                             if ($(this).hasClass("operator")){
136.                                 operacija = $(".screen").text()+tekst;
137.                                 $(".screen").fadeTo(250, .2).fadeTo(250, 1);
138.                                 flag = true;
139.                             }
140.
141.                             else if ($(this).hasClass("clear")){
142.                                 $(".screen").text("0");
143.                                 operacija = null;
144.                             }
145.
146.                             else if ($(this).hasClass("equal")){
147.                                 if (operacija!=null && operacija!=$(".screen").text()
    ){
148.                                     $(".screen").text( eval( operacija+$(".screen").t
    ext() ) );
149.                                     operacija = null;
150.                                     flag = true;
151.                                 }
152.                             }
153.
154.                             else{
155.                                 if (flag){
156.                                     $(".screen").text( tekst );
157.                                     flag = false;
158.                                 }
159.                                 else{
160.                                     $(".screen").text( parseInt( $(".screen").text()+
    tekst ) );
161.                                 }
162.                             }
163.                         });
164.                     });
165.                 </script>
166.             </body>
167.         </html>
```

Prilog VI

Programski kod: Kviz iz biografije

```
{{extend 'layout.html'}}

<head>
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

    <style>
        .navbar-inverse {
            background-image: linear-gradient(to bottom, white, white);
            background-repeat: repeat-x;
            border: transparent;
            border-radius: 7px 7px 7px 7px;
            min-height: 40px;
            padding-left: 20px;
            padding-right: 20px;
        }
        .dropdown-menu {
            background-clip: padding-box;
            background-color: white;
            border: 1px solid #004d80;
            border-radius: 6px 6px 6px 6px;
            box-shadow: 0 5px 10px rgba(0, 0, 0, 0.2);
            display: none;
            float: left;
            left: 0;
            list-style: none outside none;
            margin: 2px 0 0;
            min-width: 160px;
            padding: 5px 0;
            position: absolute;
            top: 100%;
            z-index: 1000;
        }
        .navbar .nav > li > a {
            color: #33adff;
            float: none;
            padding: 15px 15px;
            text-decoration: none;
            font-family: Trebuchet MS;
            font-size:20px;
        }
        .dropdown-menu li > a {
            display: block;
            padding: 3px 20px;
            clear: both;
            font-weight: normal;
            line-height: 20px;
            color: #33adff;
            white-space: nowrap;
            font-family: Trebuchet MS;
            font-size: 20px
        }
        footer {
            background-color: #00cc00;
            text-align: center;
            color: white;
            position: fixed;
            left: 0;
            bottom: 0;
```

```
            width: 100%;
        }
        #quiz {
                margin-left:0px;
                background:#00cc00;
                color: white;
                padding:10px 20px 10px 20px;
                margin: auto;
                width:50%;
                border-radius:20px;
                font-family: Trebuchet MS;
                }
            h1 {
                text-align:center;
                font-family: Trebuchet MS;
                color: #00cc00;
            }
            #textbox {
                height:22px;
                width:120px;
                font-size:16px;
                border-radius:5px;
                border:none;
                padding-left:5px;
            }
            #button {
                background:green;
                border:none;
                border-radius:5px;
                padding:10px;
                color:white;
                font-size:16px;
                transition-duration:.5s;
                margin-top:15px;
            }
            #button.hover {
                background:white;
                border:1px solid green;
                color:black;
                cursor:pointer;
                   }
                   #after_submit {
                       visibility:hidden;
                   }
                   #message {
                       font-size:20px;
                       text-align:center;
                   }
                   #number_correct{
                       font-size:30px;
                       text-align:center;
                  }
            </style>
        </head>

        <body>
            <div style="text-align: center;">
                <img src="{{=URL('static','images/biologija2.png')}}" alt="Biologija"
    style="width:20%">
            </div><br>

            <div>
                <h1><b>2. Najjednostavniji oblici života i stanica</b></h1>
                <form id="quiz" name="quiz">
```

```
123.                 Kod svakoga pitanja potrebno je označiti <b>samo jedan</b> točan
    odgovor. Kada odgovorite na sva pitanja pritisnite gumb "<b>Predaj!</b>" na kraju st
    ranice.<br><br>
124.                 <div>
125.                     {{for redak in db((db.bioquiz.poglavlje=='2. Najjednostavniji
    oblici života i stanica')).select():}}
126.                     <p><b>{{=redak.id}}. {{=redak.pitanje}}</b></p>
127.                     <input type="radio" name="question{{=redak.id}}" value="a"> {
    {=redak.odgovor_a}}<br>
128.                     <input type="radio" name="question{{=redak.id}}" value="b"> {
    {=redak.odgovor_b}}<br>
129.                     <input type="radio" name="question{{=redak.id}}" value="c"> {
    {=redak.odgovor_c}}<br>
130.                     <input type="radio" name="question{{=redak.id}}" value="d"> {
    {=redak.odgovor_d}}<br>
131.                     <br>
132.                     <p id="x{{=redak.id}}" style="color:red; font-
    weight: bold;"></p>
133.                     {{pass}}
134.
135.                     <input id="button" type="button" value="Predaj!" onclick="che
    ck();">
136.                 </div>
137.
138.                 <div id="after_submit" style="text-align:center;"></div>
139.                 <p id="number_correct"></p>
140.                 <p id="message"></p>
141.             </form><br><br>
142.         </div><br><br><br>
143.
144.         <a href="bio2" style="font-size:15px;font-
    family: Trebuchet MS;"><b>⇦ Prethodna</b> </a>
145.         </div><br><br>
146.
147.         {{block footer}}
148.         <footer>
149.             <p1 style="font-
    family: Trebuchet MS;"><b>SVEZNALICA</b>™ by Matea Gredelj</p1>
150.         </footer>
151.
152.         <script>
153.             function check(){
154.
155.                 var question1 = document.quiz.question1.value;
156.                 var question2 = document.quiz.question2.value;
157.                 var question3 = document.quiz.question3.value;
158.                 var question4 = document.quiz.question4.value;
159.                 var question5 = document.quiz.question5.value;
160.                 var question6 = document.quiz.question6.value;
161.                 var question7 = document.quiz.question7.value;
162.                 var question8 = document.quiz.question8.value;
163.                 var question9 = document.quiz.question9.value;
164.                 var question10 = document.quiz.question10.value;
165.
166.                 var correct = 0;
167.
168.                 {{for redak in db((db.bioquiz.poglavlje=='2. Najjednostavniji obl
    ici života i stanica')).select():}}
169.                 if (question{{=redak.id}} == "{{=redak.tocno}}"){
170.                     correct++;
171.                 }
172.                 else {
173.                     document.getElementById("x{{=redak.id}}").innerHTML = "Krivi
    odgovor!";}
174.                 {{pass}}
175.
```

```
                    var messages = ["Odlično!", "Vrlo dobro!", "Dobro!", 'Dovoljno!', 'Nedovoljno!'];
                    var range;

                    if (correct < 5) {
                        range = 4;
                    }
                    if (correct > 5 && correct < 7) {
                        range = 3;
                    }
                    if (correct > 6 && correct < 9) {
                        range = 2;
                    }
                    if (correct > 8 && correct < 10) {
                        range = 1;
                    }
                    if (correct > 9) {
                        range = 0;
                    }

                    document.getElementById("after_submit").style.visibility = "visible";
                    document.getElementById("message").innerHTML = messages[range];
                    document.getElementById("number_correct").innerHTML = "Imaš "+correct+"/10 točno!";

                    document.getElementById("button").disabled = true;
                    document.getElementById("button").style.backgroundColor = "lightgrey";
                }
            </script>
        </body>
```