

SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# ZAVRŠNI RAD

**Matea Gredeļj**

Zagreb, 2018.

SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# ZAVRŠNI RAD

Mentor:

Prof. dr. sc. Mario Essert, dipl. ing.

Student:

Matea Gredelj

Zagreb, 2018.

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

Zahvaljujem se svom mentoru prof. dr. sc. Mariu Essertu što mi je omogućio izradu završnog rada, te odvojio svoje vrijeme kako bi korisnim savjetima i stručnom pomoći olakšao njegovu izradu.

Također se zahvaljujem svojoj obitelji na podršci tijekom studiranja, pogotovo ocu na vječnoj inspiraciji i motivaciji za ovaj završni rad.

Matea Gredelj



SVEUČILIŠTE U ZAGREBU  
**FAKULTET STROJARSTVA I BRODOGRADNJE**



Središnje povjerenstvo za završne i diplomske ispite  
Povjerenstvo za završne ispite studija strojarstva za smjerove:  
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment, inženjerstvo  
materijala i mehatronika i robotika

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

## ZAVRŠNI ZADATAK

Student: **Matea Gredelj** Mat. br. 0035195966

Naslov rada na hrvatskom jeziku: **Mrežni repozitorij digitalnih slika i video uradaka**

Naslov rada na engleskom jeziku: **The network repository of digital images and video clips**

Opis zadatka:

Za potrebe demonstracijskih vježbi i udaljene nastave potrebno je osmisliti, projektirati i izvesti mrežni repozitorij u kojem će skupina nastavnika (administratori) moći pohranjivati i editirati, a skupine prijavljenih i običnih korisnika (studenata) po različitim kriterijima dohvaćati i pregledavati digitalne slike i video uratke.

U radu je potrebno načiniti:

1. Objasniti WEB2PY tehnologiju koja povezuje *MVC* (eng. *model-view-controller*) paradigmu s *Python* programom, omogućujući na taj način izvođenje *Python* programa u pozadini mrežne aplikacije.
2. Osmisliti sučelje repozitorija kako za administratore i njihove grupe (obrada i analiza zaporki, stvaranje i upravljanje korisnicima), tako i za korisnike.
3. Izvesti sučelje u *HTML5/CSS3/jQuery/Bootstrap* tehnologiji
4. Osmisliti, projektirati i izvesti *Python* program koji će unutar *MVC* kontrolera izvoditi skladištenje i osvježavanje repozitorija digitalnih slika i uradaka.
5. Osmisliti osnovne funkcije obrade spremljenih materijala, kao što su povezivanja i razmjешtanja (eng. *mixing*) dijelova materijala u nove, izvedbene cjeline.

Zadatak zadan:  
30. studenog 2017.

Rok predaje rada:  
1. rok: 23. veljače 2018.  
2. rok (izvanredni): 28. lipnja 2018.  
3. rok: 21. rujna 2018.

Predviđeni datumi obrane:  
1. rok: 26.2. - 2.3. 2018.  
2. rok (izvanredni): 2.7. 2018.  
3. rok: 24.9. - 28.9. 2018.

Zadatak zadao:

Prof. dr. sc. Mario Essert

Predsjednik Povjerenstva:

Izv. prof. dr. sc. Branko Bauer

## SADRŽAJ

SADRŽAJ .....	I
POPIS SLIKA .....	II
SAŽETAK.....	IV
SUMMARY .....	V
1. UVOD.....	1
2. web2py.....	2
2.1. Pokretanje web2py-a.....	3
2.1.1. MVC .....	10
3. Tehnologije za izradu i uređenje sučelja .....	12
3.1. HTML5 .....	12
3.2. CSS3.....	13
3.3. JavaScript.....	14
3.4. jQuery.....	14
3.5. Bootstrap.....	15
4. Izrada mrežnog repozitorija digitalnih slika i video uradaka .....	16
4.1. Models.....	16
4.1.1. db.py.....	16
4.1.2. menu.py.....	21
4.2. Controllers.....	23
4.2.1. default.py .....	23
4.3. View.....	26
4.3.1. index.html .....	28
4.3.2. messert.html .....	30
4.3.3. messertedit.html .....	33
4.3.4. student.html.....	34
5. ZAKLJUČAK.....	39
LITERATURA.....	40
PRILOZI.....	41

## POPIS SLIKA

Slika 1. Repozitorij slika – web stranica .....	1
Slika 2. web2py logo .....	2
Slika 3. web2py sučelje za pokretanje aplikacije .....	3
Slika 4. Početno sučelje web2py-a .....	4
Slika 5 . Prijava za administrativno sučelje.....	5
Slika 6. Sučelje web2py aplikacije.....	6
Slika 7. Stvaranje nove aplikacije .....	6
Slika 8. Models.....	7
Slika 9. Controllers.....	7
Slika 10. View .....	8
Slika 11. Languages .....	8
Slika 12. Modules.....	9
Slika 13.Static.....	9
Slika 14. Plugins.....	9
Slika 15. MVC.....	10
Slika 16. Kreiranje baze podataka.....	16
Slika 17. Prikaz seta podataka iz baze.....	17
Slika 18. Kalendar iz baze podataka .....	18
Slika 19. Administracija baze podataka .....	19
Slika 20. Način dodavanja novih podataka u bazu kao administrator .....	20
Slika 21. Uređivanje baze podataka .....	20
Slika 22. Prikazivanje loga.....	21
Slika 23. Kreiranje menu-a.....	22
Slika 24. Traka izbornika mrežnog repozitorija .....	23
Slika 25. Kreiranje nove stranice .....	23
Slika 26. Definiranje nove stranice .....	24
Slika 27. Definiranje stranice mrežnih repozitorija .....	24
Slika 28. Definiranje stranice za profesore .....	25
Slika 29. Definiranje stranice za uređivanje.....	26
Slika 30. Prikaz CSS-a .....	27
Slika 31. CSS stvaranje okvira.....	28
Slika 32. Kod za prijavu na stranicu .....	29
Slika 33. Preusmjerenje nakon registracije/prijave/odjave .....	29
Slika 34. Sučelje za prijavu .....	29
Slika 35. CSS za tekst i tablicu .....	30
Slika 36. Unos novih podataka pomoću forme .....	30
Slika 37. Prikaz sučelja za unos podataka.....	31
Slika 38. Postojeće objave.....	31
Slika 39. Video uradak u postojećim objavama .....	32
Slika 40. Prikaz podataka iz baze.....	33
Slika 41. Stranica za uređivanje objave .....	34
Slika 42. Uređivanje objave .....	34
Slika 43. Postavke za tipke.....	35
Slika 44. <button> .....	36
Slika 45. <script> .....	36

---

Slika 46. Tipke .....	36
Slika 47. Setovi tipki .....	37
Slika 48. Prikaz repozitorija .....	38
Slika 49. Mrežni repozitorij video uradaka .....	38

## SAŽETAK

U ovom radu će biti objašnjen način izrade web stranica i web aplikacija pomoću web2py tehnologije: Python i HTML jezika. Stranica će biti kreirana pomoću MVC (*model-view-controller*) paradigme. Kreirana web stranica će biti repozitorij slika i video uradaka. Pomoću modela stvoriti će se baze podataka u koje će korisnici (profesori) moći pohranjivati datoteke. Kroz kontroler će se definirati stranice, te način na koji će se prenositi datoteke od baze podataka do korisnika. Te na kraju kroz pogled ćemo urediti izgled web stranice pomoću CSS-a. Također ćemo kroz biblioteke JavaScripta pomoću JQuerya i Bootstrapa unaprijediti našu stranicu pomoću raznih funkcija, animacija i slično.

Ključne riječi: web2py, MVC, HTML5, CSS3, jQuery, Bootstrap



## **SUMMARY**

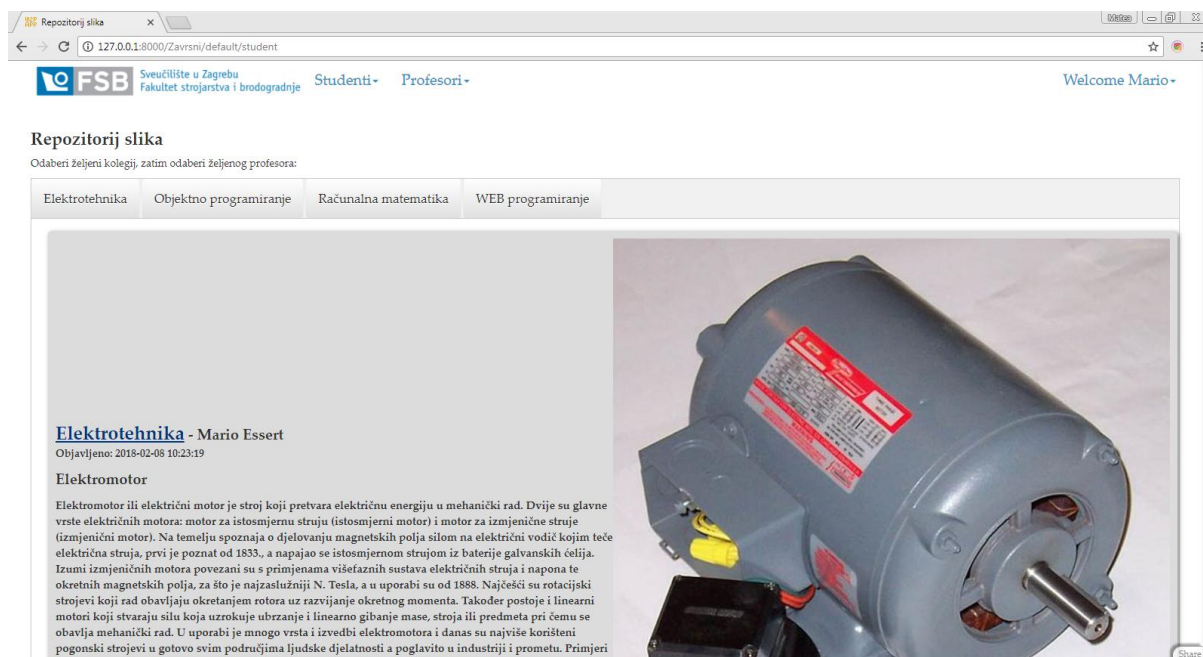
Throughout this piece it will be explained the way of creating web pages and web applications with the help of web2py technology : Python and HTML languages. The page will be created with MVC (model-view-controller) paradigm. Created web page will be a repository of pictures and videos. With the help of model databases will be created in which the users (profesors) will be able to save the data. Throughout controller the pages will be defined, just like the way of data transmission. At the end, through view, design of the page will be made with the help of CSS. Also throughout JavaScript libraries with jQuery and Bootstrap the page will be improved with a variety of functions, animations, ect.

Key words: web2py, MVC, HTML5, CSS3, jQuery, Bootstrap

## 1. UVOD

U ovom završnom radu biti će objašnjen način izrade web stranica ili web aplikacije pomoću web2py tehnologije. Pomoću web2py-a biti će napravljen repozitorij slika i video uradaka [Slika 1]. Repozitoriju se neće moći pristupiti ako osoba nije prijavljena, pa će se objasniti i način na koji će korisnici po raznim kriterijima moći pristupiti određenim stranicama. Uz to će biti napravljene i posebne stranice za skupinu korisnika (profesora) koji će moći dodavati slike i video uratke koji će kasnije biti drugoj skupini korisnika (studentima) dostupan. Uz web2py biti će korišteni i HTML5, CSS3, jQuery i Bootstrap kako bi stranica bila zanimljivija i animirana. Kako je web2py pisan Python programerskim jezikom on se i koristi njime. Također će biti prikazan način implementacije Python koda u web2py.

Web2py je veoma moćan alat i ovime samo pokazujemo jako malen dio mogućnosti koje nam pruža. Uz pomoć navedenih tehnologija, postoje i razne druge, no trenutno nam nisu potrebne pa ih nije potrebno ni spominjati.



Slika 1. Repozitorij slika – web stranica

## 2. web2py

web2py je web aplikacija otvorenog koda napisana u Python programskom jeziku. Stvorila ga je zajednica profesionalnih programera i profesora predvođenima Massimom Di Pierrom. Prva verzija aplikacije bila je puštena 2007. godine (web2py 1.0), a posljednja verzija web2py-a (2.15.4) 2017. godine. web2py omogućuje programiranje dinamičnog web sadržaja pomoću Pythona.



Slika 2. web2py logo

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

web2py je izgrađen za sigurnost. To znači da automatski rješava mnoge probleme koji mogu dovesti do sigurnosnih propusta. na primjer, potvrđuje sve unose, izbjegava sve izlaze, te preimenuje prenesene datoteke. web2py vodi brigu o sigurnosnim problemima, rako da programeri imaju manje izgleda za uvođenje ranjivosti.

web2py uključuje bazu podataka (DAL, eng. *Database Abstraction Layer*) koji dinamički piše SQL, tako da programer to ne mora. DAL zna kako SQL generirati 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

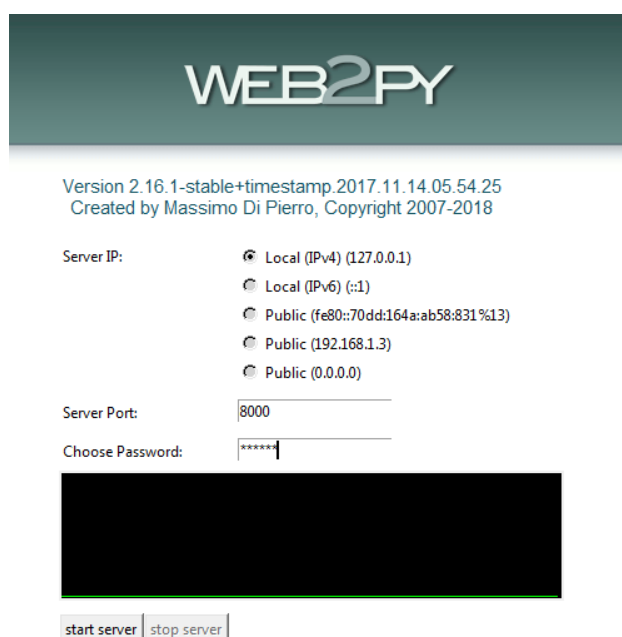
definira jedna ili više tablica baze podataka, web2py automatski generira potpuno funkcionalno sučelje administracije baze podataka na webu za pristup bazi podataka i njenim tablicama.

web2py se razlikuje od ostalih web kostura jer 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. Web2py se pokreće sa CPython-om (implementacija C-a) i PyPy-om (Pythonom pisanim u Pythonu) na Pythonu 2.7. [1]

## 2.1. Pokretanje web2py-a

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 [Slika 3] 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.



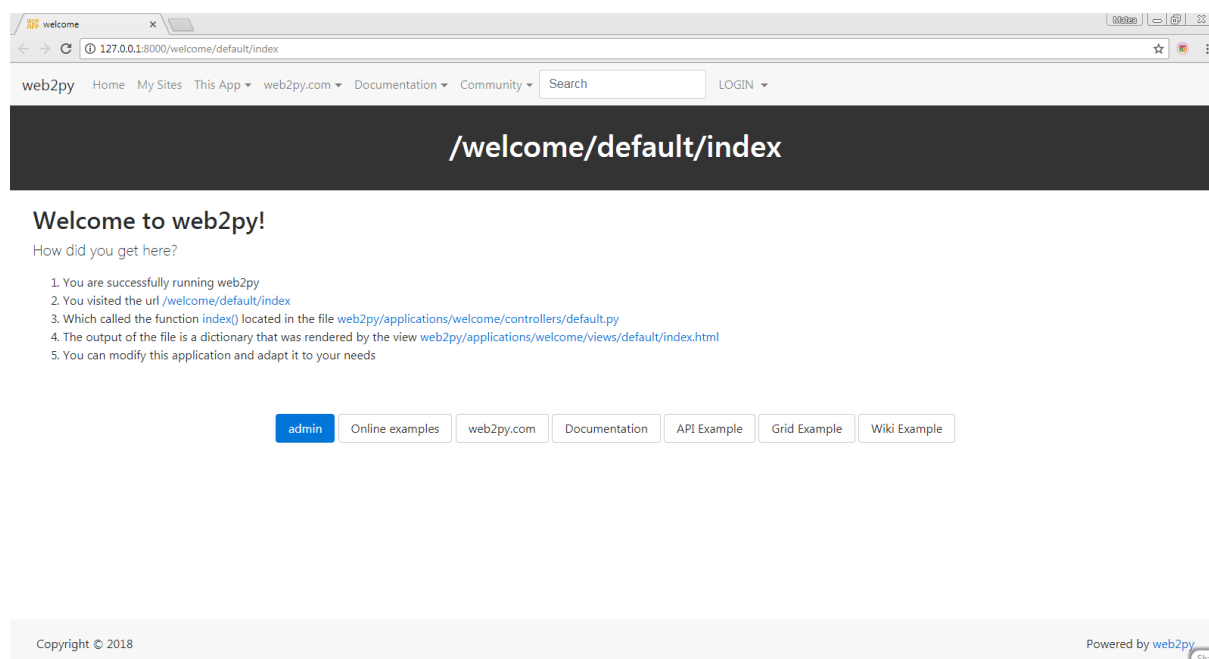
Slika 3. web2py sučelje za pokretanje aplikacije

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 [Slika 4] na stranici:

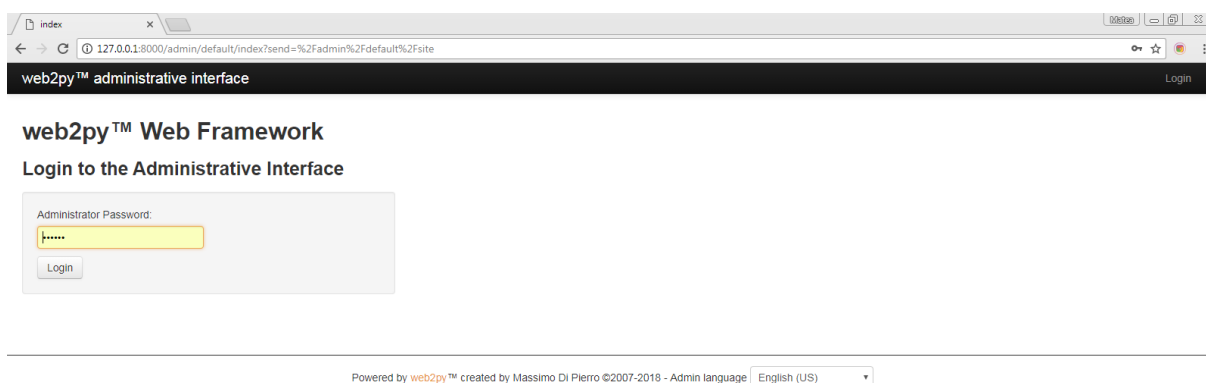
`http://127.0.0.1:8000/`

Ako računalo nema zadani preglednik, otvorite web-preglednik i unesite gore navedeni URL .



**Slika 4. Početno sučelje web2py-a**

Klikom na „administrativno sučelje“ dolazi se na prijavu za administrativno sučelje [Slika 5].



Questions?  
join our google group

### Slika 5 . Prijava za administrativno sučelje

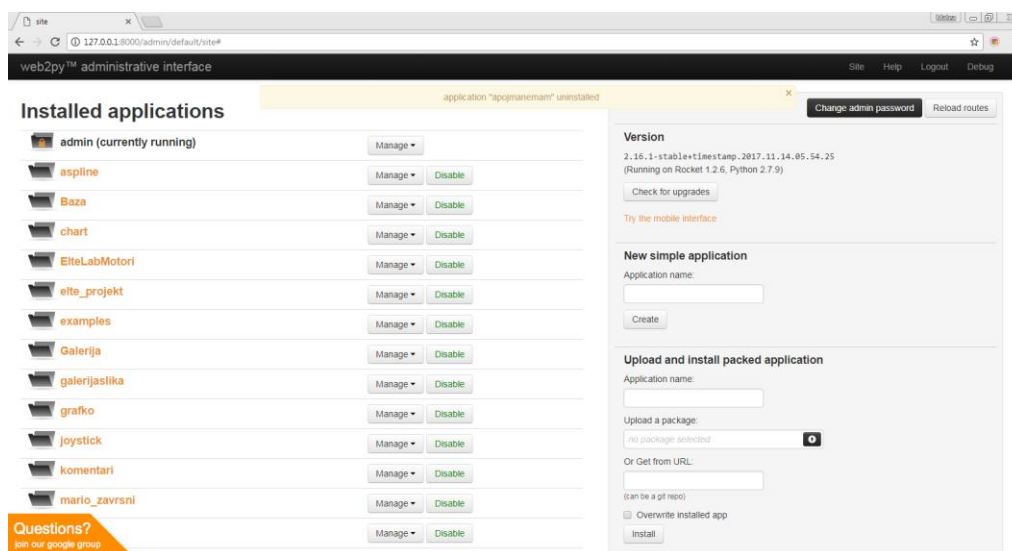
Administratorska lozinka je lozinka koja je odabrana prilikom pokretanja. Kao što se može primijetiti, postoji samo jedan administrator, pa samim time postoji i samo jedna administratorska lozinka. Iz sigurnosnih razloga, od programera se traži da odabere novu lozinku svaki put kada se upali web2py, osim ako nije navedena opcija *<recycle>*. No treba paziti, ovo nije jednako mehanizmu autentifikacije u web2py aplikacijama.

Ova stranica [Slika 6] navodi sve instalirane web2py programe i omogućuje administratoru da upravlja njima. web2py dolazi sa 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).

Na stranici administratorske aplikacije, mogu se izvršiti sljedeće aplikacije:

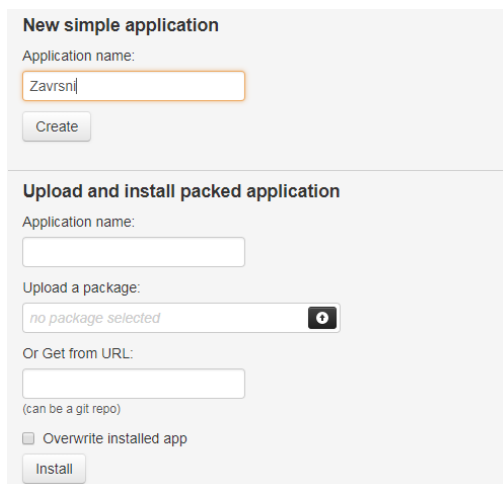
- Instalirati aplikaciju ispunjavanjem obrasca u donjem desnom dijelu stranice. Imenujemo novu aplikaciju, odaberemo datoteku koja sadrži pakiranu aplikaciju ili URL gdje se program nalazim, te stisnemo '*submit*'.
- Deinstalirati aplikaciju klikom na odgovarajući gumb. Također postoji stranica za potvrdu 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 svaki aplikaciju
- Uređivati aplikaciju [1]



Slika 6. Sučelje web2py aplikacije

U slučaju ovog rada, stvaramo novu aplikacijom naziva „Završni“ [Slika 7].



Slika 7. Stvaranje nove aplikacije

Nakon što se pritisne 'Create', aplikacija je stvorena, kao i kopija ugrađene aplikacije dobrodošlice.

Sada imamo kopiju aplikacije dobrodošlice. Kako bi ju uredili, moramo stisnuti 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 šest kategorija:

- „Models“ : opisuje prikaz podataka “ [Slika 8]



Slika 8. Models

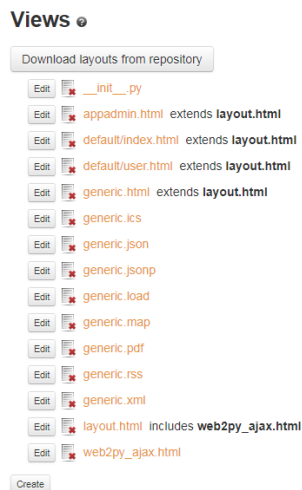
- „Controllers“: opisuje logiku aplikacije i tijek rada [Slika 9]



Slika 9. Controllers

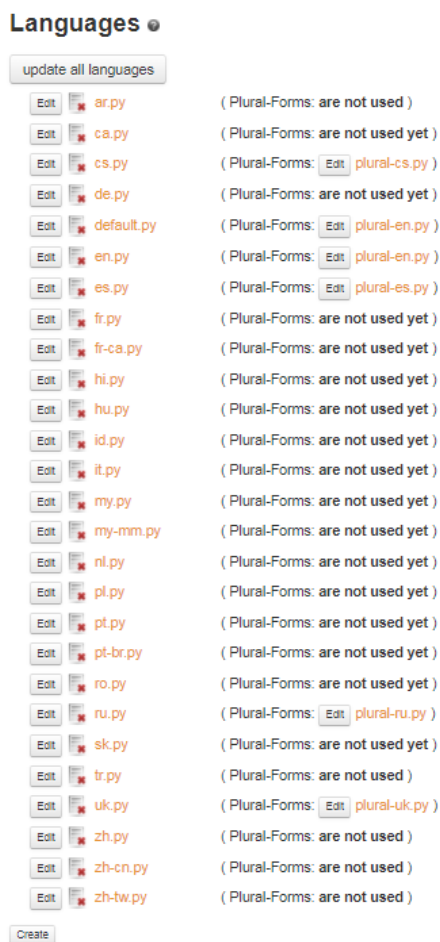


- „Views“: opisuje prezentaciju podataka [Slika 10]



Slika 10. View

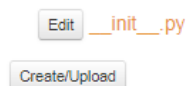
- „Languages“: opisuje kako prevoditi aplikaciju na druge jezike [Slika 11]



Slika 11. Languages

- „*Modules*“: Python moduli koji pripadaju aplikaciji [Slika 12]

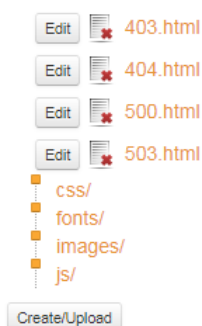
### Modules



Slika 12. Modules

- „*Static*“: slike, CSS datoteke, JavaScript datoteke i slično [Slika 13]

### Static

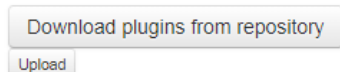


Slika 13. Static

- „*Plugins*“: skupine datoteka dizajnirane za zajednički rad [Slika 14]

### Plugins

There are no plugins

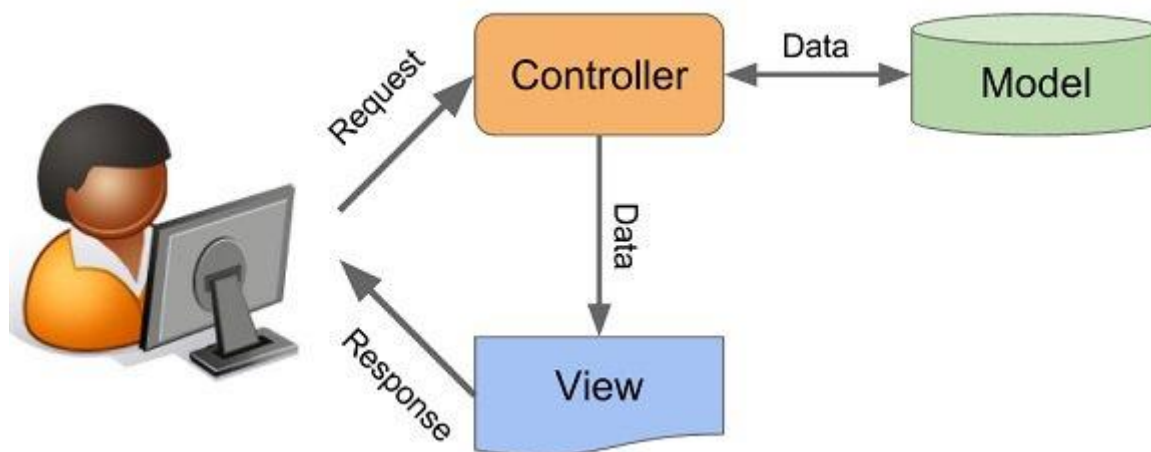


Slika 14. Plugins

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.

### 2.1.1. MVC

Web2py potiče programera da odvoji prikaz podataka (*model*), prezentaciju podatka (*view*) i tijek rada aplikacije (*kontroler*). Način rada MVC-a je prikazan na idućoj slici [Slika 15].



Slika 15. MVC

Model–View–Controller (MVC) je obrazac softverske arhitekture. 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 napatke (eng. *input*) i pretvara ih u naloge modelu ili pogledu. Ovakva arhitektura olakšava nezavisan razvoj, testiranje i održavanje određene aplikacije.

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]

### 3. Tehnologije za izradu i uređenje sučelja

Za izradu mrežnog repozitorija u web2py tehnologiji uz već objašnjenu MVC paradigmu i Python jezik, potrebne su nam i 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 bi smo ih bolje upoznali prije uporabe.

#### 3.1. HTML5

HTML je kratica za HyperText Markup Language, što znači prezentacijski jezik za izradu web stranica. Hipertekst dokument stvara se pomoću HTML jezika. HTML jezikom oblikuje se sadržaj i stvaraju se hiperveze hipertekst dokumenta. HTML 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. Html 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.

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 su ove 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. HTML5 standard nalazi se u statusu radnog dokumenta (eng. *draft*), a očekuje se da će postati službeno objavljen sredinom 2012. godine, dok bi konačne specifikacije trebale biti gotove u drugom kvartalu 2014. 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 iil Microsoftovog silverlight-a, mogućnost upravljanja pomoću tipkovnice i opcijama za bilo koju vrstu manipulacija, 'drag and drop', 'canvas' kao i ostali novi elementi. [3] [4]

### 3.2. CSS3

CSS je kratica od (eng.) *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.

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 prekomplikiran 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]

### 3.3. 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.

Kao multi-paradigmatski jezik, JavaScript podržava stilove programiranja temeljene na događajima, te funkcionalne i imperativne stilove programiranja (poput objektnog programiranja – Python). 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. [5]

### 3.4. jQuery

jQuery je biblioteka JavaScripta s više platformi dizajniranih da pojednostavljuje klijentima pisanje skripti HTML-a. 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 Ajaxove aplikacije. jQuery također pruža mogućnost razvojnim programerima za stvaranje dodataka na vrhu biblioteke JavaScripta. 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. [6]

### **3.5. Bootstrap**

Bootstrap je besplatna i otvorenog koda biblioteka za izradu web stranica i web aplikacija. 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). [7]



## 4. Izrada mrežnog repozitorija digitalnih slika i video uradaka

Kao što je već rečeno, mrežni repozitorij će se izvesti u web2py-u, pomoću ranije navedenih tehnologija za uređenje. Pomoću MVC paradigme izradit će se mrežni repozitorij za potrebe demonstracijskih vježbi i udaljene nastave. Pomoću modela (eng. *model*) će biti kreirane baze podataka – za svakog profesora jedna, te jedna koja će u sebi sadržavati sve korisnike koji su prijavljeni na tu stranicu. Također ćemo tu urediti menu. Zatim u kontroleru (eng. *controllers*) stvaramo nove stranice, te kod njih upisujemo podatke koje nam određena web stranica mora vratiti i zahtjeve za ulaz na web stranicu. Dok u pogledu (eng. *view*) stvaramo i oblikujemo sučelje, dizajniramo izgled, te programiramo funkcije koje će se izvršavati na stranici. Sve gore navedeno detaljno će biti objašnjeno u idućim poglavljima.

### 4.1. Models

#### 4.1.1. db.py

Izradu stranice započinjemo sa bazom podataka. Potrebno je kreirati baze podataka za svakog profesora, kako bi se pohranjivale slike i video uradci. Zbog jednostavnosti i skraćivanja koda, sve će biti prikazano samo za jednog korisnika pošto je izrada koda za ostale profesore analogna objašnjenom.

Za stvaranje nove baze koristimo funkciju `db.define_table()`. Kroz tu funkciju stvaramo tablicu koja se sastoji od raznih stupaca, vrste i naziva koje zadamo [Slika 16].

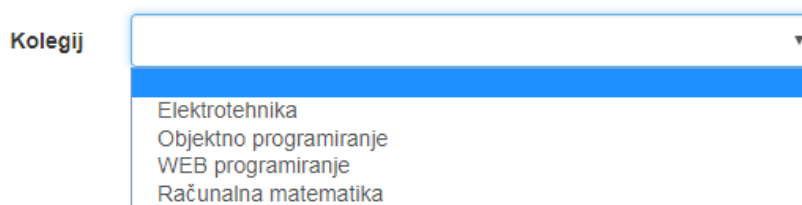
```
101
102 db.define_table('messert',
103     Field('kolegij', requires=IS_IN_SET(['Elektrotehnika', 'Objektno programiranje',
104     'WEB programiranje', 'Računalna matematika'])),
105     Field('naslov', 'string', unique=True, required=True, requires=IS_NOT_EMPTY()),
106     Field('datum', 'datetime', required=True, requires=IS_DATETIME(format=T('%Y-%m-%d %H:%M:%S'),
107     error_message='Mora biti YYYY-MM-DD HH:MM:SS!')),
108     Field('komentar', 'text', required=True, requires=IS_NOT_EMPTY()),
109     Field('slika', 'upload'),
110     Field('video', 'upload')
111 )
```

Slika 16. Kreiranje baze podataka

Kao što možemo vidjeti u gore navedenom kodu, nakon što koristimo funkciju za stvaranje tablice, prva stvar koju moramo definirati je ime te baze podataka, što je u ovom slučaju „messert“, jer se radi o bazi podataka za profesora Maria Esserta. Nakon toga definiramo

polja tablice – Field. Potrebna polja za izradu web stranica su nazivaju „kolegij“, „naslov“, „datum“, „komentar“, „slika“ i „video“. Iza naziva polja tablice baze podatka, nalazi se vrsta tog polja – vrijeme, tekst, upload i slično; a iza toga možemo još upisati neke zahtjeve poput jedinstvenosti u bazi podataka, ili da polje ne smije ostati prazno.

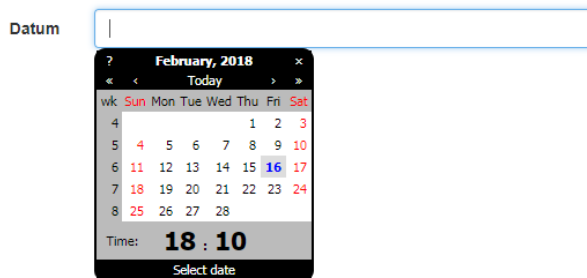
Prvo polje naziva „kolegij“ iza sebe ima zahtjev `=IS_IN_SET()` – da se mora izabrati jedna vrijednost od četiri zadane, tj. mora se izabrati jedan od četiri ponuđena kolegija. Ovo nam smanjuje mogućnost pogreške pri upisu imena kolegija, jer samo s jednim pogrešnim slovom, kasnije kada bi se kroz neke uvjete trebao pozvati podatak iz baze, ne bi bio pravilno pozvan zbog te gramatičke greške. Kada ćemo upisivati podatke u bazu zbog zahtjeva ćemo dobiti mali izbornik koji možemo vidjeti na idućoj slici [Slika 17].



**Slika 17. Prikaz seta podataka iz baze**

Drugo polje, 'naslov', je eng. *string*, tj. niz znakova, u našem slučaju to će biti kratak naslov koji opisuje sliku ili video uradak. Ovdje također imamo nekoliko zahtjeva. Ovo polje, naslov, mora biti jedinstveno. Ne možemo imati dvije objave sa istim imenom. To dobivamo pomoću zahtjeva koji definiramo istinitim za jedinstvenost (unikatnost). Također imamo zahtjev da ovo polje ne smije ostati prazno (`=IS_NOT_EMPTY()`).

Treće polje je 'datum', a tip je eng. *datetime*. Ovo nije potrebno previše objašnjavati, radi se polju u koje se će se upisati datum i vrijeme kada se bude objavila objava. Zahtjev (`=IS_DATETIME()`) nam govori da se datum mora upisati po točno određenoj formi, tj. mora biti: YYYY-MM-DD HH:MM:SS! (godina-mjesec-dan sat-minuta-sekunda). No, kako to ne bi bilo previše komplicirano za upisivanje, web2py nudi lakšu opciju, te kada se pritisne za upis datuma izbacuje kalendar u kojem jednostavno označite datum i vrijeme i on sam upiše točno traženu formu. Taj kalendar je vidljiv na idućoj slici [Slika 18].



**Slika 18. Kalendar iz baze podataka**

Iduće polje 'komentar' je doista jednostavno. Tip je *'text'*, što znači da ovdje imamo povećí prostor za opis slike ili video uratka. Također je postavljen zahtjev da se polje ne ostavi praznim.

Zadnja dva polja su istog tipa, *'upload'*. 'slika' i 'video' su razdvojeni u bazi kako bi se kasnije lakše pozivalo na web stranicu. Kao što je vidljivo, ovdje nema nikakvih zahtjeva zato što nisu potrebni.

Također uz sve ove zahtjeve postoji još mnogo drugih. Sve ovisi o našim potrebama pri izradi baze podataka, te u konačnici web stranice. Neki od drugih mogućih zahtjeva su i:

- = IS\_IN\_DB()
- = IS\_NOT\_IN\_DB()
- = IS\_INT\_IN\_RANGE()
- = IS\_ALPHANUMERIC()
- = IS\_MATCH()
- = IS\_UPPER()
- Itd.

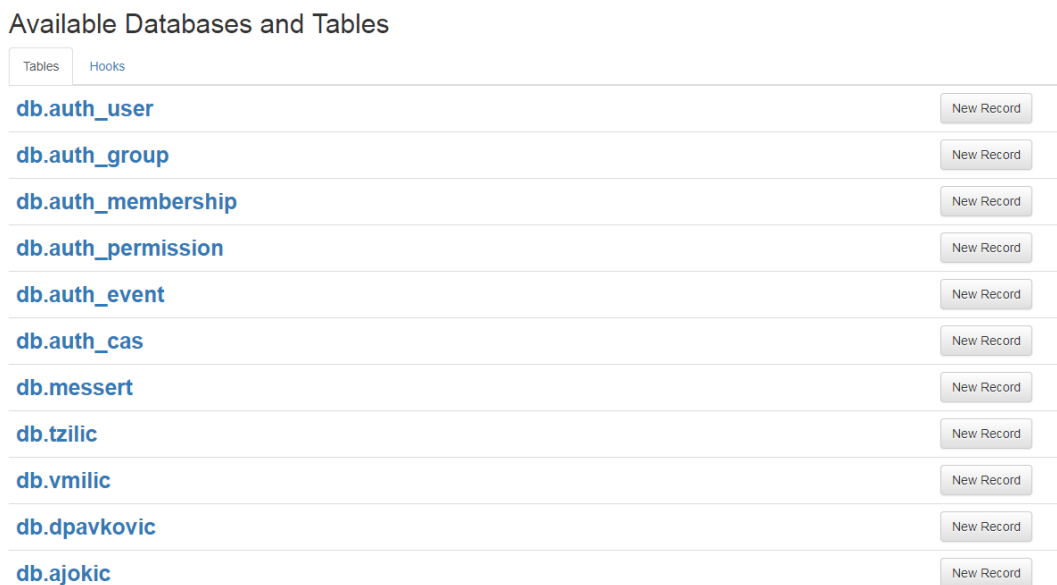
Kako ima raznih mogućnosti za zahtjeve, tako postoji i različiti tipovi polja baze podataka. Neki od su:

- boolean
- integer
- blob

- double
- time
- passwod
- json
- itd.

Kao što je i ranije navedeno, uz bazu podataka „messert“, napravljene su i baze podataka „tzilic“, „vmilic“, „dpavkovic“ i „ajokic“. Sadrže iste stupce kao i baza podataka „messert“.

Kada završimo s ispisivanjem funkcije za stvaranje baze podataka, bazi možemo pristupiti pomoću 'database administration-a'. Uz 6 baza podataka web2py-a, možemo vidjeti da su stvorene i naše nove baze [Slika 19].



Available Databases and Tables	
Tables	Hooks
<a href="#">db.auth_user</a>	New Record
<a href="#">db.auth_group</a>	New Record
<a href="#">db.auth_membership</a>	New Record
<a href="#">db.auth_permission</a>	New Record
<a href="#">db.auth_event</a>	New Record
<a href="#">db.auth_cas</a>	New Record
<a href="#">db.messert</a>	New Record
<a href="#">db.tzilic</a>	New Record
<a href="#">db.vmilic</a>	New Record
<a href="#">db.dpavkovic</a>	New Record
<a href="#">db.ajokic</a>	New Record

**Slika 19. Administracija baze podataka**

Baze podataka web2py-a su već postavljene kako bi nam olakšale kreiranje web stranice. Sve imaju sličan zadatak, upisivati korisnike kada se prijave na stranicu. Pomoću tih baza možemo korisnike svrstavati u razne grupe, te kasnije korisnicima iz tih grupa davati različite ovlasti i pristup stranicama.

Ako administrator stranice želi dodati nove podatke u bazu, stisnemo gumb 'New Record', te nam se otvori novi prozor.

Database db Table messert

New Record

Kolegij

Naslov

Datum

Komentar

Slika  Nije odabrana niti jedna datoteka.

Video  Nije odabrana niti jedna datoteka.

**Slika 20. Način dodavanja novih podataka u bazu kao administrator**

Na slici [Slika 20] možemo vidjeti da se traže sva polja koja smo ranije definirali. Administrator ove stranice ovdje može puniti bazu podataka. Također, pomoću 'data administration-a' možemo brisati ili uređivati određene podatke baze podataka [Slika 21].

Database db select

Rows in Table

Query:

Update:

Delete:

The "query" is a condition like "db.table1.field1==value". Something like "db.table1.field1=db.table2.field2" results in a SQL JOIN. Use (...)&(…) for AND, (...)|(…) for OR, and ~(…) for NOT to build more complex queries. "update" is an optional expression like "field1=newvalue". You cannot update or delete the results of a JOIN

0 selected

Import/Export

or import from csv file

Nije odabrana niti jedna datoteka.

**Slika 21. Uređivanje baze podataka**

#### 4.1.2. *menu.py*

Kao što i samo ime kaže, ovdje se bavimo izbornikom. U web2py-u je zadano da se traka s izbornikom nalazi na vrhu stranice. Radi jednostavnosti i lakšeg korištenja, tako će biti i za ovu web stranicu, no to se lako može promijeniti. Traka s izbornikom se može nalaziti na vrhu stranice, na dnu, lijevo, desno, ili kako god želimo.

U gornjem lijevom kutu se nalazi logo web2py-a, no njega ćemo zamijeniti sa logom Fakulteta strojarstva i brodogradnje. Prvo je potrebno pronaći sliku koju želimo za logo, te ju spremiti u datoteku naše aplikacije, u datoteku *'images'*, koja se nalazi u datoteci *'static'*. Bitno nam je znati gdje spremamo sliku kako bi kasnije znali stvoriti put kojim se slika poziva u stranicu. No, ne moramo sliku spremati, također možemo koristiti i link preko kojeg se može pozivati slika. Zbog sigurnosti, ipak ćemo sliku spremiti, jer uvijek postoji mogućnost da se dogodi nešto sa stranicom iz koje uzimamo sliku, te u tome slučaju nam se logo neće prikazati. Na idućoj slici možemo vidjeti kod za pozivanje loga [Slika 22].

```
8 response.logo = IMG(_src=URL('static','images/Logo.png'),
9                    _style="height:50px; width:auto; position:right; ",
10                   _alt=T('FSB-Logo'))
```

**Slika 22. Prikazivanje loga**

U osmom redu koda pozivamo logo, te definiramo da je slika i da ju pozivamo iz datoteke *'static'*. Zatim moramo definirati da se nalazi u poddatoteci *'images'*, te da joj je ime „Logo.png“. U devetom redu uređujemo sliku kako bi bolje pristajala uz našu traku izbornika. Postavljamo da visina slike bude 50 piksela, širinu je postavljena tako da se automatski prilagodi visini, te logo pozicioniramo lijevo. Deseti red nam je alternativa u slučaju da postoji neka greška i da se logo ne može učitati. Tada će nam se na mjestu loga ispisati *'FSB-logo'*. To je veoma korisno za gore navedene slučajeve, ako uzimamo sliku preko nekog linka.

Zatim se definira izbornik. U našem izborniku će se nalaziti tri manja izbornika. Prvi je izbornik za prijavu koji u sebi sadrži mogućnost prijave, odjave, te mogućnost promjene zaporke i pomoći ako smo zaboravili lozinku. Ovaj izbornik je također ugrađen u web2py te ga nećemo mijenjati jer nam je potreban. Idući manji izbornik nazvan je *'Studenti'*, te on je za

korištenje studenata i u njemu student može odabrati da li želi vidjeti repozitorij slika ili repozitorij video uradaka. Zadnji manji izbornik 'Profesori' se sastoji od imena profesora.

Svaki profesor ima svoju stranicu sa koje može puniti bazu i uređivati svoje objave, no o tome ćemo kasnije. Sada je samo bitno da znamo da pritiskom na ime profesora otvara se stranica kojoj pristupa taj profesor kako bi objavio nešto. Kod za traku izbornika možemo vidjeti na idućoj slici [Slika 23].

```
33 def _():
34     # shortcuts
35     app = request.application
36     ctr = request.controller
37     # useful links to internal and external resources
38     response.menu += [
39         (T('Studenti'), False, '#', [
40             (T('Repozitorij slika'), False, URL('Zavrsni', 'default', 'student')),
41             (T('Repozitorij videa'), False, URL('Zavrsni', 'default', 'student2')),
42         ]),
43         (T('Profesori'), False, '#', [
44             (T('Mario Essert'), False, URL('Zavrsni', 'default', 'mSSERT' )),
45             (T('Tihomir Žilić'), False, URL('Zavrsni', 'default', 'tzilic' )),
46             (T('Vladimir Milić'), False, URL('Zavrsni', 'default', 'vmillic' )),
47             (T('Danijel Pavković'), False, URL('Zavrsni', 'default', 'dpavkovic' )),
48             (T('Andrej Jokić'), False, URL('Zavrsni', 'default', 'ajokic' )),
49         ]),
50     ]
```

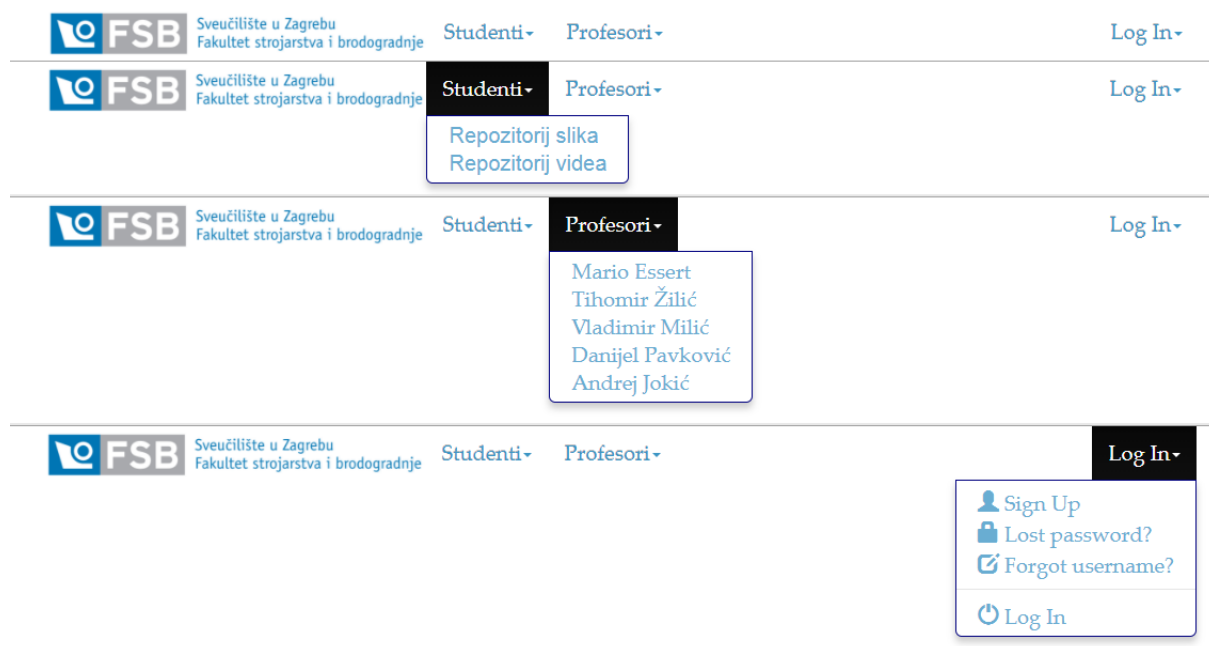
Slika 23. Kreiranje menu-a

Ovdje možemo vidjeti da postoje dva glavna izbornika: 'Studenti' i 'Profesori' koji se sastoje od gore navedenih mogućnosti izbora. Samim pritiskom na ta dva izbornika stranica se ne preusmjerava na neku drugu, nego samo otvara daljnji izbornik.

Kod svih izbora možemo vidjeti da imaju „URL('Zavrsni', 'default', '\_\_\_\_\_')“. Taj dio koda nam govori da pritiskom na taj izbor otvara se nova stranica. Članovi u zagradi nam govore put kojim se poziva nova stranica. Zadnji član zagrade je jedinstven za svaku stranicu, jer je to naziv za tu stranicu. Detaljnije o tome će biti u idućim poglavljima.

Kada spremimo svoj novo napisani kod, izgled našeg novog izbornika možemo vidjeti na idućoj slici [Slika 24].

**NAPOMENA:** Ovdje je već uvršten CSS, te boje izbornika i fontovi slova su promijenjeni. No, to će biti pobliže objašnjeno u idućim poglavljima.

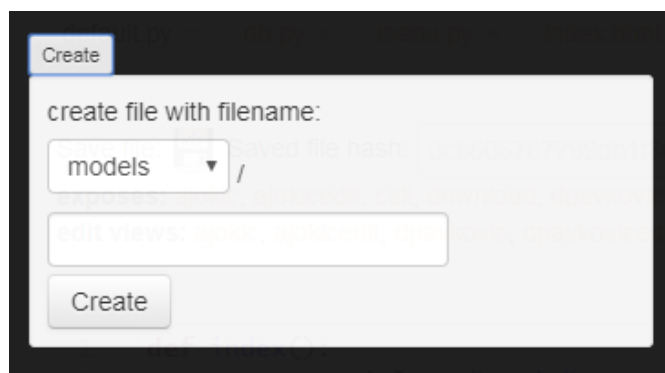


Slika 24. Traka izbornika mrežnog repozitorija

## 4.2. Controllers

### 4.2.1. *default.py*

U *default.py* možemo reći da stvaramo stranice i definiramo ih. Prvo je potrebno fizički kreirati novu stranicu. Stranicu možemo lako kreirati pomoću izbornika s lijeve strane kojeg otvorimo na kada dotaknemo lijevi rub. U tom izborniku postoji gumb 'Create' te njega pritisnemo, a on nam otvori maleni prozor prikazan na idućoj slici [Slika 25].



Slika 25. Kreiranje nove stranice

Ovdje prvo moramo odabrati u izborniku da želimo 'view', jer želimo novu stranicu, te zatim joj moramo dati ime, u ovom slučaju moramo napisati '*default/imestranice*'. Kao i u



prethodnim poglavljima, obraditi ćemo samo po jednu stranicu od studenata i profesora jer se ostale rade analogno onima koje će biti opisane.

Kada smo definirali naziv nove stranice, pritisnemo 'Create', te dobivamo novu stranicu. No ona nije u potpunosti definirana. Tu sada dolazi default.py. sada je potrebno definirati stranicu te što će nam ona vraćati.

Prva stranica koju ćemo definirati je stranica za prijavu. Naziv te stranice je „indeks“, što je već ranije bilo zadano od web2py-a, pa se može zadržati. Ovdje definiramo da ta stranica ima naziv 'Login' (vidljivo u traci preglednika) i da nam vraća formu za prijavu (više o tome će biti u idućim poglavljima) [Slika 26].

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

Slika 26. Definiranje nove stranice

Iduće dvije stranice koje definiramo su stranice na kojima će biti prikazani repozitoriji slika i video uradaka. Nazivi tih stranica su 'student' i 'student2', a imena vidljiva u pregledniku su 'Repozitorij slika' i 'Repozitorij video uradaka' [Slika 27]. Ovdje moramo postaviti uvjet da se na stranicu može ući samo ako je osoba prijavljena na stranicu. To se radi pomoću `@auth.requires_login()`, kojeg upišemo prije definiranja stranice. Time je osigurano da ne postoji mogućnost da netko tko nije student ili profesor uđe na stranicu. U slučaju da osoba nije prijavljena, zbog toga će se osoba morati vratiti na stranicu prijave i prijaviti se.

```
5 @auth.requires_login()
6 def student():
7     response.title = "Repozitorij slika"
8     return dict()
9
10 @auth.requires_login()
11 def student2():
12     response.title = "Repozitorij video uradaka"
13     return dict()
```

Slika 27. Definiranje stranice mrežnih repozitorija

Sada kada su definirane općenite stranice kojima mogu pristupiti svi prijavljeni korisnici, potrebno je napraviti stranice kojima mogu pristupiti samo profesori kako bi mogli

objavljivati nove sadržaje. Potrebno je definirati po dvije stranice za svakog profesora, jedna gdje će moći profesori stavljati slike/video uratke, te samim time puniti svoju bazu podataka i stranicu na kojoj će svoje objave moći uređivati.

Prvo je potrebno napraviti stranicu pomoću koje će profesori objavljivati slike/video uratke. No također kao i u prethodne dvije stranice, potrebno je postaviti uvjet kako bi samo taj određeni profesor mogao pristupiti toj stranici i objavljivati pod svojim korisničkim imenom. Ovdje koristimo modificirani gornji uvjet – uvjetujemo da stranici može pristupiti korisnik koji je prijavljen, te se u bazi nalazi pod svojim jedinstvenim id-om. U slučaju dolje prikazanom na slici to je broj 7 – Mario Essert (sedmi po redu je upisan u bazu podataka). S tim uvjetom smo odredili da ovoj stranici može pristupiti samo profesor Essert.

```
15 @auth.requires(auth.user_id==7 , requires_login=True)
16 def messert():
17     response.title = "Mario Essert"
18
19     forma=SQLFORM(db.messert, submit_button="Objavi")
20
21     if forma.process(keepvalues=False).accepted:
22         response.flash="Uspješno objavljeno!"
23     elif forma.errors:
24         response.flash="Došlo je do greške"
25     return dict(forma=forma)
```

### Slika 28. Definiranje stranice za profesore

Vidljivo je da je definirani naziv stranice „messert“, dok je vidljivo ime u pregledniku „Mario Essert“. U dvadeset i petom redu je vidljivo da nam stranica vraća formu. Formu zapravo dobivamo pomoću SQLFORM-a. To je automatski generirani složeni obrazac. Vidljivo je u devetnaestom redu da se SQLFORM povezuje na bazu podataka 'messenger'. Kao što smo ranije definirali u bazi podataka tip podatka (tekst, datum, vrijeme, upload...), ovaj konstruktor se bavi automatskim spremanjem unesenih polja u bazu podataka u odgovarajuće polje. SQLFORM konstruktor omogućuje različite prilagodbe, kao što su mijenjanje oznaka, ažuriranje podataka, brisanje podataka i slično. SQLFORM nam omogućava da u što kraćem vremenu napravimo formu i ažuriramo bazu podataka. Također je u njemu definirano da kada upišemo određene podatke za spremanje u bazu podataka možemo i te podatke objaviti pomoću gumba. Od dvadeset i prvog reda do dvadeset i četvrtog postavljamo uvjetnu naredbu. Ako su podaci koje smo upisali u formu prihvaćena u bazu podataka, u gornjem

desnom kutu javiti će nam se flash poruka u kojoj će pisati „Uspješno objavljeno!“. U suprotnom, ako ne prihvati pisati će „Došlo je do greške!“ [Slika 28].

Sličan slučaj imamo i kod stranice za uređivanje objava. Razlika je da ovdje uzimamo određenu objavu i nju mijenjamo ili brišemo [Slika 29]. Logika toga je ista prethodno objašnjennoj. Također ovoj stranici može pristupiti samo profesor Essert (u ovom slučaju). Kada se objava uspješno promijeni/izbriše, stranica se preusmjerava na stranicu za unos, u ovom slučaju „messert“.

```
27 @auth.requires(auth.user_id==7 , requires_login=True)
28 def messertedit():
29     response.title = "Uređivanje objave"
30     Id=eval(request.vars.Id)
31     rekord=db(db.messert.id==Id).select().first()
32     forma=SQLFORM(db.messert, record=rekord, submit_button="Promijeni", delete_label="Obriši", deletable=True)
33
34     if forma.process().accepted:
35         session.flash="Objava je uspješno promijenjena!"
36         redirect(URL('messert'))
37     elif forma.errors:
38         response.flash="Došlo je do greške!"
39     return dict(forma=forma)
```

### Slika 29. Definiranje stranice za uređivanje

default.py također sadrži i još neke zadane definirane stranice (mogućnosti) poput 'download' za skidanje određenih objekata sa stranice, 'user' za mogućnost prijave i obrade prijave, 'call' i slično.

## 4.3. View

Kao što sam naziv govor, 'view', tj. pogled je područje gdje se uređuje stranica, od rasporeda objekata na stranici do izgleda. Za početak potrebno je na svim stranicama urediti traku za izbornik. To radimo pomoću CSS-a.

Kako bi počeli pisati kod za CSS, potrebno je otvoriti oznaku (eng. *tag*) <style>. Sve što pišemo unutar toga web stranica to vidi kao CSS, te se to koristi za uređivanje stranice. Kroz ovaj dio koda opisujemo izgled trake izbornika. Kao što je ranije bilo napomenuto, ovdje tek se bavimo izgledom. Za ovu stranicu je odabrano da traka bude bijele boje, kao i pozadina stranica. Za font je odabran "Palatino Linotype", te veličina teksta je 20 piksela. Boja teksta je napisana pomoću heksadekaskog izraza. Svaka boja ima svoj određeni kod. Također neke od boja imaju i svoja imena, pa se umjesto koda može napisati i ime te boje. Treći način upisa boje je pomoću rgb (eng. *red-green-blue*), odnosno pomoću udjela primarnih boja, crvene,

zelene i plave u toj novoj boji. Pošto je traka za izbornik zapravo pravokutnik, pomoću CSS-a možemo kutove zaobliti i odabrati radijus ( u pikselima). Također možemo staviti sjenu iz tog pravokutnika („kutije“). Također možemo mijenjati poziciju teksta: lijevo, desno, centar... Sve ovo i više smo koristili kako bi se dobila traka za izbornik kakva trenutno je. Kod za to je idući [Slika 30].

```
5 <style type="text/css">
6
7     .navbar-inverse {
8         background-image: linear-gradient(to bottom, white, white);
9         background-repeat: repeat-x;
10        border: transparent;
11        border-radius: 7px 7px 7px 7px;
12        min-height: 40px;
13        padding-left: 20px;
14        padding-right: 20px;
15    }
16
17    .dropdown-menu {
18        background-clip: padding-box;...
19        background-color: #FFFFFF;...
20        border: 1px solid navy;
21        border-radius: 6px 6px 6px 6px;
22        box-shadow: 0 5px 10px rgba(0, 0, 0, 0.2);
23        display: none;
24        float: left;
25        left: 0;
26        list-style: none outside none;
27        margin: 2px 0 0;
28        min-width: 160px;
29        padding: 5px 0;
30        position: absolute;
31        top: 100%;
32        z-index: 1000;
33    }
34
35    .navbar .nav > li > a {
36        color: #3D96C5;
37        float: none;
38        padding: 15px 15px;
39        text-decoration: none;
40        font-family: "Palatino Linotype",Palatino,sans-serif;
41        font-size:20px;
42    }
43
44    .dropdown-menu li > a {
45        display: block;
46        padding: 3px 20px;
47        clear: both;
48        font-weight: normal;
49        line-height: 20px;
50        color: #69ADD2;
51        white-space: nowrap;
52        font-family: "Palatino Linotype",Palatino,sans-serif;
53        font-size: 20px
54    }
55    .footer {
56        background-color:white;
57        height: 20px;
58    }
59 </style>
```

Slika 30. Prikaz CSS-a

### 4.3.1. *index.html*

Prvo što je potrebno ovdje je ubaciti gore navedeni kod za izgled trake izbornika. Ova stranica je početna stranica koja se prva otvara. Zapravo, ova stranica je stranica za prijavu. Stranica je zamišljena tako da se na njoj nalazi samo forma za upisivanje korisničkog imena i lozinke. Prvo pomoću CSS-a kreiramo jedan pravokutnik kako bi u njega stavili formu za prijavu. Nazvali smo ga 'frame'. Ovo nije potrebno, no koristimo samo zbog ljepšeg izgleda i usklađenosti stranice. Pravokutnik i njegov okvir su bijele boje pa se ne vide, iako u pravilu postoje. Ovdje također određujemo širinu i visinu pomoću postotka u odnosu na širinu i visinu ekrana na kojem je web stranica učitana. Kod za to se nalazi na idućoj slici [Slika 31].

```
61 <style>
62     div#frame {
63         background: white;
64         border: 3px solid white;
65         padding: 100px;
66         margin: auto;
67         width: 40%;
68         height: 30%;
69         position: relative;
70         top: 80px;
71     }
72 </style>
```

Slika 31. CSS stvaranje okvira

Iduće što je potrebno je napraviti formu za prijavu. Iako to možemo raditi od nule, web2py se već pobrinuo za to pa im svoju ugrađenu formu koju možemo koristiti. Ovdje smo spojili HTML i CSS, te u oznaci `<div>` pišemo CSS, font, boja i veličina fonta. Zatim pišemo kod za formu koji je već zadan web2py-u. Kod je veoma jednostavan [Slika 32] i govori nam da ako niste u bazi podataka, ne možete pristupiti web stranici. Također tu postoji mogućnost zaborava lozinke pa imamo tipku s kojom se može pomoći oko toga.

```

75 <div id="frame" style="font-family:Palatino; color:#3D96C5; font-size:20px">
76   {{extend 'layout.html'}}
77 <h2>{{T( request.args(0).replace('_', ' ').capitalize() )}}</h2>
78 <div id="web2py_user_form">
79   {{=form}}
80   {{if request.args(0)=='login':}}
81     {{if not 'register' in auth.settings.actions_disabled:}}
82       {{pass}}
83     {{if not 'request_reset_password' in auth.settings.actions_disabled:}}
84       <br/>
85       <a href="{{=URL(args='request_reset_password')}}">Zaboravili ste lozinku?</a>
86     {{pass}}
87   {{pass}}
88 </div>
89 </div>

```

Slika 32. Kod za prijavu na stranicu

No kako radimo sa MVC, potrebno je sada vratiti se na model, tj. na bazu podataka. Tamo smo morali odrediti na koji link je potrebno stranicu preusmjeriti kada s osoba registrira, prijavi ili odjavi [Slika 33].

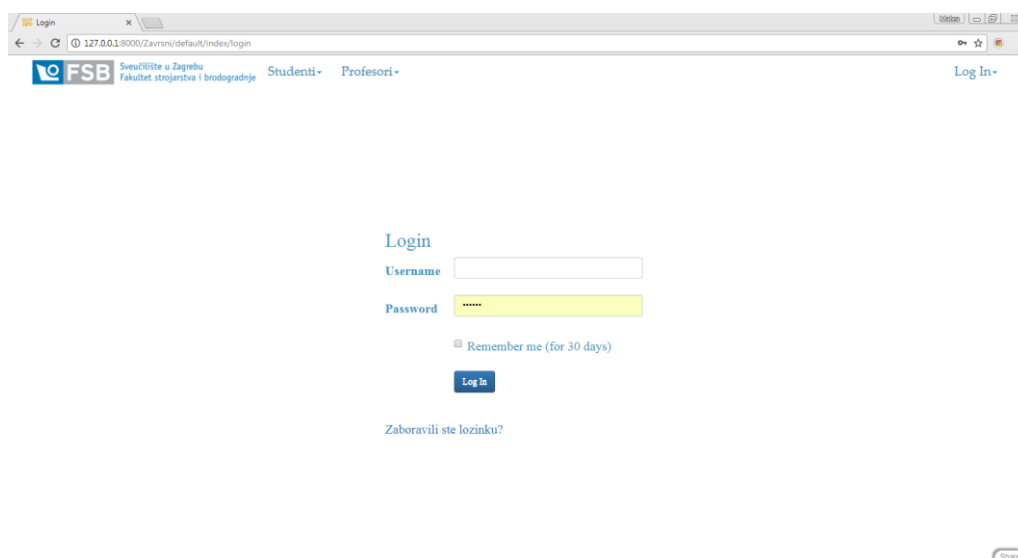
```

98 auth.settings.register_next = URL(r=request,c='default/student')
99 auth.settings.login_next = URL(r=request,c='default/student')
100 auth.settings.logout_next = URL(r=request,c='default/index')

```

Slika 33. Preusmjeravanje nakon registracije/prijave/odjave

Na kraju dobivamo funkcionalnu web stranicu pomoću koje se može prijaviti za rad dalje. Izgled stranice vidljiv je na idućoj slici [Slika 34].



Slika 34. Sučelje za prijavu

### 4.3.2. messert.html

Kao i kod svake stranice i ovdje prvo ubacujemo CSS za traku izbornika. Zatim moramo postaviti i CSS za ovu stranicu [Slika 35], točnije, moramo postaviti pravokutnik u kojem će se ispisivati podaci uneseni od strane korisnika (profesora).

```
60 <style>
61 .tekst{
62     box-shadow:2px 2px 8px rgba(0,0,0,0.5);
63     border-radius: 5px;
64     background: gainsboro;
65     margin:10px;
66     padding:10px;
67 }
68 .tekst table{
69     width:100%;
70     font-size:16px;
71 }
72 .tekst td{
73     padding: 5px;
74     width:40%;
75 }
76 </style>
```

Slika 35. CSS za tekst i tablicu

Iduće što je potrebno na stranici je napraviti formu preko koje će profesori puniti svoju bazu podataka sa slikama i video uradcima. Nju pozivamo pomoću forme koju smo definirali u kontroleru. Ovdje možemo vidjeti cijeli MVC proces. Zahtjev za podacima od korisnika dolazi iz pogleda, te on šalje taj zahtjev u kontroler, koji iz baze podataka (modela) vuče podatke i šalje ih nazad u pogled. Formu pozivamo veoma jednostavno jer je ona već ranije definirana [Slika 36].

```
78 <div style="margin:auto; width:80%; box-shadow:2px 2px 8px #cce0ff; padding:20px
79     ; margin-top:30px; margin-bottom:50px;">
80     <h3 style="border-bottom:2px solid #3D96C5;">
81         <b>Unos nove slike/videoa:</b>
82     </h3>
83     <div style="width:50%;">
84         {{=forma}}
85     </div>
```

Slika 36. Unos novih podataka pomoću forme

Pomoću oznake `<h3>` povlačimo crtu kako bi odvojili naslov od forme. Na kraju dobivamo pogled koji je vidljiv na idućoj slici [Slika 37].

**Unos nove slike/videa:**

Kolegij

Naslov

Datum

Komentar

Slika  Nije odabrana niti jedna datoteka.

Video  Nije odabrana niti jedna datoteka.

Slika 37. Prikaz sučelja za unos podataka


No, nakon toga je zamišljen i pregled dosadašnjih objava kako bi korisnik mogao uređivati i brisati svoje objave. Radi ljepšeg pregleda, koristit ćemo oznaku `<table>` kako bi napravili tablicu s jednim redom, te ju punili sa podacima iz baze podataka. Također moramo postaviti neke zahtjeve. Za slučaj da još objava nema, korisnik mora dobiti povratnu informaciju o tome. Tada će se na stranici ispisivati: „Trenutno ne postoji niti jedna objava!“. No ako postoje objave, tada će se koristiti petlja koja će uzimati podatke po redcima baze podataka, ovisno o traženom podatku ispisivati ga na određeno mjesto [Slika 38.].

**Dosadašnje objave:**

**Elektrotehnika**

**Elektromotor**

Elektromotor ili električni motor je stroj koji pretvara električnu energiju u mehanički rad. Dvije su glavne vrste električnih motora: motor za istosmjernu struju (istosmjerni motor) i motor za izmjenične struje (izmjenični motor). Na temelju spoznaja o djelovanju magnetskih polja silom na električni vodič kojim teče električna struja, prvi je poznat od 1833. a napucao se istosmjernom strujom iz baterije galvanskih ćelija. Izumi izmjeničnih motora povezani su s primjenama višefaznih sustava električnih struja i napona te okretnih magnetskih polja, za što je najpoznatiji N. Tesla, a u uporabi su od 1888. Najčešći su rotacijski strojevi koji rad obavljaju okretanjem rotora uz razvijanje okretnog momenta. Također postoje i linearni motori koji stvaraju silu koja uzrokuje ubrzanje i linearno gibanje mase, stroja ili predmeta pri čemu se obavlja mehanički rad. U uporabi je mnogo vrsta i izvedbi elektromotora i danas su najviše korišteni pogonski strojevi u gotovo svim područjima ljudske djelatnosti a poglavito u industriji i prometu. Primjen korištenja u prometu su lokomotive vlakova, tramvaj, trolejbusi, električni automobili.



[Preuzmi sliku]

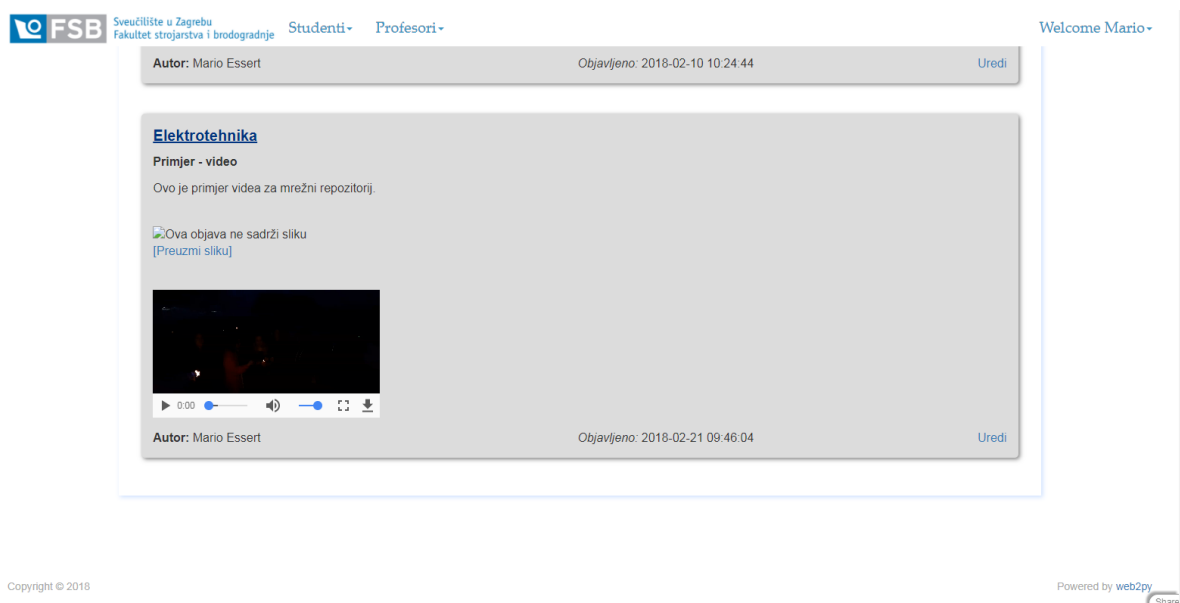
0:00

Autor: Mario Essert Objavljeno: 2018-02-08 10:23:19

Slika 38. Postojeće objave



Kod tekstualnih podataka nije potrebno previše objašnjavati. Ispisuju se istim fontom, no različitim bojama i debljini, ovisno o kriterijima potrebnima za ispis. Ovdje se pojavljuje objavljivanje slika i video uradaka. Za objavljivanje slike potrebna je oznaka `<img>`, te pomoću CSS-a određujemo veličinu i poziciju slike. Također za slučaj da objava ne sadrži sliku, imamo alternativu pomoću atributa `alt`. Tu upisujemo tekst koji će se prikazati u slučaju da nema slike u objavi. Atribut `src` nam pomaže da datoteku preuzmemo iz baze podataka, dok atribut `'href'` nam omogućuje preuzimanje datoteke sa stranice na računalo [Slika 40]. Za objavljivanje video uratka veoma je sličan način. Razlika je u oznaci koja je ovdje `<video>`, te ovdje imamo atribut `controls`. Ovaj atribut nam omogućava da dobijemo kontrole, te da možemo puštati i zaustavljati video kada želimo, mijenjati glasnoću, te prikazivati video preko cijelog ekrana. Ovdje se mogu koristiti i neki drugi atributi poput `'autoplay'`, `'loop'`, `'muted'` i slično. Prikaz videa u dosadašnjim objavama možemo vidjeti na idućoj slici. [Slika 38.]



Slika 39. Video uradak u postojećim objavama

```

88     <b> Dosadašnje objave:</b>
89 </h3>
90 <div>
91     {{if db(db.messert.id>0).isempty()==False:}}
92     {{for redak in db(db.messert.id>0).select():}}
93     <div class="tekst">
94         <table>
95             <tr>
96                 <td><b style="color: #003380; font-size:20px; text-decoration: underline ">
97                     {{=redak.kolegij}}</b></td><td align="right">
98
99                 </tr>
100                <tr>
101                    <td colspan="1" valign="top"><div style="min-height:25px;">
102                        <b>{{=redak.naslov}}</b>
103                    </div></td>
104                </tr>
105                <tr>
106                    <td colspan="2" valign="top"><div style="min-height:50px;">
107                        {{=redak.komentar}}
108                    </div></td>
109                </tr>
110                <tr>
111                    <td colspan="2" valign="top"><div style="min-height:75px;">
112                        <br>
114                        <a href="{{=URL(f='download', args=redak.slika)}}>[Preuzmi sliku]</a>
115                    </div></td>
116                </tr>
117                <tr>
118                    <td colspan="2" valign="top"><div style="min-height:75px;">
119                        <video style="width:300px" src="{{=URL('download', args=redak.video)}}>
120                        controls</video><br>
121                    </div></td>
122                </tr>
123                <tr>
124                    <td align="left"><b>Autor: </b>Mario Essert</td>
125                    <td align="center"><i>Objavljeno: </i>{{=redak.datum}}</td>
126                    <td align="right"><a href="{{=URL('default', 'messertedit',
127                        vars=dict(Id=redak.id))}}>Uredi</a></td>
128                </tr>
129            </table>
130        </div>
131        <br>
132        {{pass}}
133        {{else:}},
134        <p>
135            Trenutno ne postoji niti jedana objava!
136        </p>
137        {{pass}}
138    </div>
</div>

```

Slika 40. Prikaz podataka iz baze

U sto dvadeset i petom redu imamo link koji vodi na drugu stranicu, stranicu „messertedit“ na kojoj uređujemo i brišemo dosadašnje objave – više o tome može se vidjeti u idućem poglavlju.

#### 4.3.3. *messertedit.html*

Ova stranica je zapravo veoma slična prethodnoj. Jedina razlika je u tome što pritiskom na gumb 'Uredi' u prethodnom poglavlju, otvara se forma već popunjena sa određenom objavom.

[Slika 41.] Ovdje je potrebno samo promijeniti objavu, ili u krajnjem slučaju obrisati je [Slika 42.].

**Slika 41. Stranica za uređivanje objave**

```

59 <div style="margin:auto; width:80%; box-shadow:2px 2px 8px #cce0ff;
60     padding:20px; margin-top:30px; margin-bottom:50px;">
61     <h3 style="border-bottom:2px solid #0066ff;">
62         <b>Uređivanje objave:</b>
63     </h3>
64     <div style="width:50%;">
65         {{=forma}}
66     </div>
67     <hr>
68 
```

**Slika 42. Uređivanje objave**

#### 4.3.4. student.html

Zadnju stranicu, repozitorij slika za studente, započinje kao i sve ostale, sa CSS-om za traku izbornika. Zatim koristimo oznaku `<meta>`. Oznaka `<meta>` sadrži metapodatke o HTML dokumentu. Metapodaci neće biti prikazani na stranici, već će se moći analizirati. Meta elementi obično se koriste za određivanje opisa stranice, ključnih riječi, autora dokumenta, zadnje izmijenjene i druge metapodatke. Preglednik može koristiti metapodatke

(kako prikazati sadržaj ili ponovno učitati stranicu), tražilice (ključne riječi) ili druge web usluge.

Zatim pomoću CSS-a uređujemo način na koji će izgledati tipke pomoću kojih kroz razne kriterije dohvaćamo podatke [Slika 43].

```
73 <meta name="viewport" content="width=device-width, initial-scale=1">
74 <style>
75 body {font-family: Palatino Linotype;}
76
77 .tab {
78     overflow: hidden;
79     border: 1px solid #ccc;
80     background-color: #f1f1f1;
81 }
82
83 .tab button {
84     background-color: inherit;
85     float: left;
86     border: none;
87     outline: none;
88     cursor: pointer;
89     padding: 14px 16px;
90     transition: 0.3s;
91     font-size: 17px;
92 }
93
94 .tab button:hover {
95     background-color: #ddd;
96 }
97
98 .tab button.active {
99     background-color: #ccc;
100 }
101
102 .tabcontent {
103     display: none;
104     padding: 6px 12px;
105     border: 1px solid #ccc;
106     border-top: none;
107 }
108 </style>
```

**Slika 43. Postavke za tipke**

Sljedeće što je potrebno je napisati kod koji će pomoću određenih kriterija otvoriti određene repozitorije. Prvo je potrebno stvoriti četiri tipke pomoću kojih uz određene kriterije će se otvoriti nove tipke. To radimo pomoću oznake `<button>`. Sada napokon dolazi korištenje JavaScripta, tj. jQuery-a i Bootstrapa. Kako bi na pritisak jedne od navedenih tipki se nešto dogodilo (otvorile novi set tipki), potrebno je napraviti funkciju koja to izvodi. Ovdje se ta funkcija zove 'otvorikolegij' [Slika 44].

```

112 <h1 style="font-size:23px">
113 <b>Repozitorij slika</b>
114 </h1>
115 <p>Odaberi željeni kolegij, zatim odaberi željenog profesora:</p>
116
117 <div class="tab">
118 <button class="tablinks" onclick="otvorikolegij(event, 'elektroteh')">Elektrotehnika</button>
119 <button class="tablinks" onclick="otvorikolegij(event, 'op')">Objektno programiranje</button>
120 <button class="tablinks" onclick="otvorikolegij(event, 'racmat')">Računalna matematika</button>
121 <button class="tablinks" onclick="otvorikolegij(event, 'webp')">WEB programiranje</button>
122 </div>

```

Slika 44. &lt;button&gt;

Funkciju opisujemo pomoću oznake <script>. Ta oznaka se koristi za definiranje skripte na strani klijenta (JavaScript). Uobičajene upotrebe za JavaScript su manipuliranje slikom, provjera forme i dinamičke promjene sadržaja, kao što je to u ovom slučaju. Kod navedene funkcije se nalazi na idućoj slici [Slika 45].

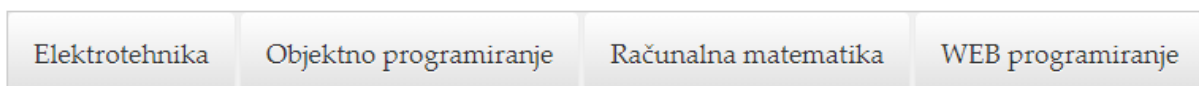
```

495 <script>
496 function otvorikolegij(evt, kolegij) {
497     var i, tabcontent, tablinks;
498     tabcontent = document.getElementsByClassName("tabcontent");
499     for (i = 0; i < tabcontent.length; i++) {
500         tabcontent[i].style.display = "none";
501     }
502     tablinks = document.getElementsByClassName("tablinks");
503     for (i = 0; i < tablinks.length; i++) {
504         tablinks[i].className = tablinks[i].className.replace(" active", "");
505     }
506     document.getElementById(kolegij).style.display = "block";
507     evt.currentTarget.className += " active";
508 }
509 </script>

```

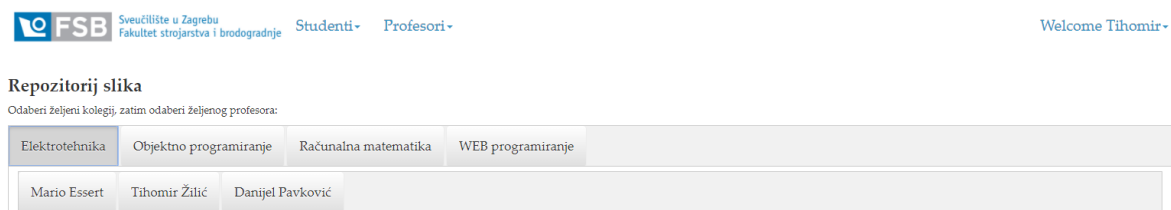
Slika 45. &lt;script&gt;

Funkcija ove skripte nam govori što će se dogoditi u slučaju pritiska na određenu gore zadanu tipku [Slika 46.]. Pritiskom na tipku otvarati će se različite kartice, te aktivna kartica će biti prikazana drugačije. Ovisno o daljnjem kodu, za svaku trenutnu karticu, tj. tipku postoji određen broj novih tipki koje će biti prikazane pritiskom na početne kartice.



Slika 46. Tipke

Potpuno istom analogijom dobivamo drugi set tipki [Slika 47], kako bi pri pretraživanju repozitorija dobili preciznije rješenje. Prvi set tipki je bio kako bi se odabrao kolegij, dok kod drugog seta biramo profesora čije objave želimo vidjeti. Kao i u prvom slučaju. Potrebno je pomoću skripte napraviti funkciju koja će otvarati repozitorije. Ovdje funkciju nazivamo 'otvoriprof'.



**Slika 47. Setovi tipki**

Kada smo odabrali kolegij i profesora čije objave želimo vidjeti, potrebno je postaviti uvjete pomoću kojih se iz baze uzimaju podaci. Objasniti ćemo na prvom primjeru u kodu, dok za druge sve vrijedi analogno. Definirano je funkcijom da na pritisak tipke 'Elektrotehnika' otvaraju se imena profesora, zatim pritiskom na tipku 'Mario Essert', otvara se novi prozor koji povezujemo pomoću id-a 'elektromeesert'. Koristimo tablicu sa dva stupca radi urednijeg prikaza. U lijevi stupac stavljamo tekst, dok u desnom prikazujemo sliku i mogućnost njenog preuzimanja. No, kako bi dobili točno objave od profesora Esserta iz elektrotehnike, potrebno je postaviti neke uvjete. Prvo je potrebno ispitati da li u bazi profesora postoji objava za taj kolegij, te ako postoji potrebno je provjeriti da li postoji objava sa slikom (u ovom slučaju), a ne s videom. Zatim pomoću CSS-a i HTML-a ponovno ubacujemo podatke na stranicu kao što smo to učinili i kod osobne stranice profesora. Jedina razlika je da ovdje ubacujemo sa tablicom sa dva stupca zbog bolje preglednosti [Slika 48].

Razlika repozitorija slika i repozitorija video uradaka je samo u jednom dijelu koda. Tamo gdje se nalazi kod za slike sa znakom `<img>`, mijenjamo sa oznakom `<video>` i pripadnim kodom koji je već prethodno naveden i objašnjen.

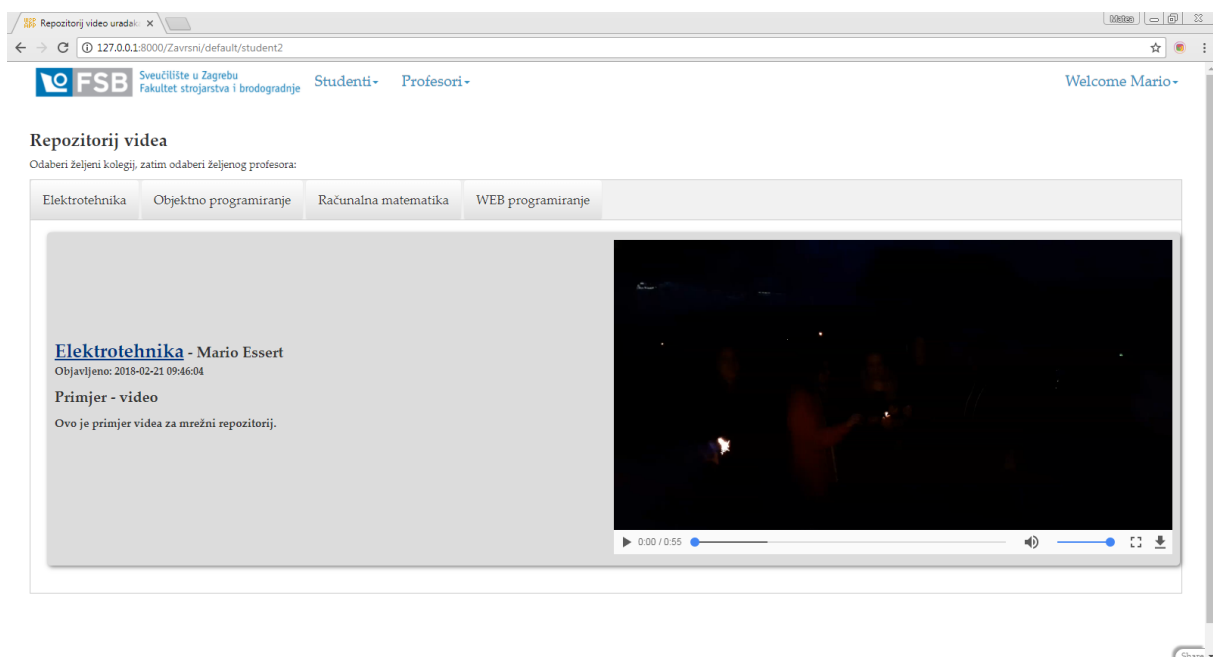
```

132 <div id="elektromessert" class="tabcontent" >
133     {{if db(db.messert.kolegij=='Elektrotehnika').isempty()==False:}}
134     {{for redak in db(db.messert.kolegij=='Elektrotehnika' and db.messert.video==' ').select():}}
135     <div class="tekst">
136         <table>
137             <tr>
138                 <th width=50%><b style="color: #003380; font-size:25px; text-decoration: underline ">
139                     {{=redak.kolegij}}</b><b style="font-size:20px"> - Mario Essert</b>
140                 <p>Objavljeno: {{=redak.datum}}</p>
141
142                 <p style="min-height:25px; font-size:20px "><b>{{=redak.naslov}}</b></p>
143                 <p style="min-height:50px;font-size:15px">{{=redak.komentar}}</p>
144                 </th>
145                 <th width=50%><div style="min-height:75px">
146                     <br>
147                     <a href="{{URL(f='download', args=redak.slika)}}">[Preuzmi sliku]</a>
148                 </div></th>
149             </tr>
150         </table>
151     </div>
152     <br>
153     {{pass}}
154     {{else:}}
155     <p>
156         Trenutno ne postoji niti jedana objava!
157     </p>
158     {{pass}}
159 </div>
160

```

Slika 48. Prikaz repozitorija

Modificiranjem ovog koda dobivamo potpuno funkcionalne repozitorije [Slika 1.] [Slika 49.]. Uz odabir kolegija i profesora olakšavamo si pretragu repozitorija i ciljano tražimo potrebne podatke. Potpuni kod ove stranice može se vidjeti u prilogu.



Slika 49. Mrežni repozitorij video uradaka

## 5. ZAKLJUČAK

Radeći ovu web stranicu, uspjela sam sve svoje znanje izrade web stranica i Python programiranja spojiti i napraviti web stranicu kojom se veoma ponosim. Radeći ovaj mrežni repozitorij shvatila sam da se želim time baviti u svome životu. web2py veoma olakšava način izrade web stranica i uz to i moje znanje Python programerskog jezika, nebo je samo granica. Ovaj završni rad pokazuje samo površinu mogućnosti koje nam daje web2py. Uz Python i HTML jezik, koji su veoma važni, dajem ipak prednost CSS-u i JavaScriptu. Smatram da u današnje doba, puno je bitnija mogućnost korisnika da kroz zanimljivu interakciju sa stranicom dobiva povratnu informaciju. Zato je web2py za to savršen. On se koristi MVC paradigmom i time na brz i efikasan način rješava probleme koji su veoma bitni kod drugih načina programiranja web stranica. Također rješava druge probleme s već nekim zadanim vrijednostima unutar sebe i time s programera skida teret programiranja od nule. Naravno ako je potrebno, sve se može napraviti od nule, tako da postoji i ta mogućnost. Ništa nije konačno. Kao ni web2py.



## LITERATURA

- [1] <http://web2py.com/books/default/chapter/29>
- [2] <https://hr.wikipedia.org/wiki/Model-view-controller>
- [3] <https://en.wikipedia.org/wiki/HTML5>
- [4] Gasston, P.: Moderni Web, Dobar plan, 2013.
- [5] <https://en.wikipedia.org/wiki/JavaScript>
- [6] <https://en.wikipedia.org/wiki/JQuery>
- [7] [https://en.wikipedia.org/wiki/Bootstrap\\_\(front-end\\_framework\)](https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework))
- [8] <https://www.w3schools.com>

## **PRILOZI**

- I. CD-R disc
- II. Kod za web stranicu

## Prilog 1. Kod za web stranico

### 1.1. db.py

```

from gluon.contrib.appconfig import AppConfig

myconf = AppConfig(reload=True)

if not request.env.web2py_runtime_gae:
    db = DAL(myconf.take('db.uri'), pool_size=myconf.take('db.pool_size',
    cast=int), check_reserved=['all'])
else:
    db = DAL('google:datastore+ndb')
    session.connect(request, response, db=db)

response.generic_patterns = ['*'] if request.is_local else []
response.formstyle = myconf.take('forms.formstyle')
response.form_label_separator = myconf.take('forms.separator')

from gluon.tools import Auth, Service, PluginManager

auth = Auth(db)
service = Service()
plugins = PluginManager()

auth.define_tables(username=True, signature=False)

mail = auth.settings.mailer
mail.settings.server = 'logging' if request.is_local else
myconf.take('smtp.server')
mail.settings.sender = myconf.take('smtp.sender')
mail.settings.login = myconf.take('smtp.login')

auth.settings.registration_requires_verification = False
auth.settings.registration_requires_approval = False
auth.settings.reset_password_requires_verification = True
auth.settings.registration_requires_approval = True

from gluon.tools import Auth
auth = Auth(db)
auth.define_tables(username=True, signature=False)

auth.settings.register_next = URL(r=request, c='default/student')
auth.settings.login_next = URL(r=request, c='default/student')
auth.settings.logout_next = URL(r=request, c='default/index')

db.define_table('mSSERT',
    Field('kolegij', requires=IS_IN_SET(['Elektrotehnika',
    'Objektno programiranje',
    'WEB programiranje', 'Računalna matematika'])),
    Field('naslov', 'string', unique=True, required=True,
    requires=IS_NOT_EMPTY()),
    Field('datum', 'datetime', required=True,
    requires=IS_DATETIME(format=T('%Y-%m-%d %H:%M:%S'),
    error_message='Mora biti YYYY-MM-DD HH:MM:SS!')),

```

```

        Field('komentar', 'text', required=True,
requires=IS_NOT_EMPTY()),
        Field('slika', 'upload'),
        Field('video', 'upload')
    )
db.define_table('tzilic',
    Field('kolegij', requires=IS_IN_SET(['Elektrotehnika',
'Objektno programiranje', 'WEB programiranje', 'Računalna matematika'])),
    Field('naslov', 'string', unique=True, required=True,
requires=IS_NOT_EMPTY()),
    Field('datum', 'datetime', required=True,
requires=IS_DATETIME(format=T('%Y-%m-%d %H:%M:%S'), error_message='Mora
biti YYYY-MM-DD HH:MM:SS!')),
    Field('komentar', 'text', required=True,
requires=IS_NOT_EMPTY()),
    Field('slika', 'upload'),
    Field('video', 'upload')
    )
db.define_table('vmilic',
    Field('kolegij', requires=IS_IN_SET(['Objektno
programiranje', 'Računalna matematika'])),
    Field('naslov', 'string', unique=True, required=True,
requires=IS_NOT_EMPTY()),
    Field('datum', 'datetime', required=True,
requires=IS_DATETIME(format=T('%Y-%m-%d %H:%M:%S'), error_message='Mora
biti YYYY-MM-DD HH:MM:SS!')),
    Field('komentar', 'text', required=True,
requires=IS_NOT_EMPTY()),
    Field('slika', 'upload'),
    Field('video', 'upload')
    )
db.define_table('dpavkovic',
    Field('kolegij', requires=IS_IN_SET(['Elektrotehnika', ])),
    Field('naslov', 'string', unique=True, required=True,
requires=IS_NOT_EMPTY()),
    Field('datum', 'datetime', required=True,
requires=IS_DATETIME(format=T('%Y-%m-%d %H:%M:%S'), error_message='Mora
biti YYYY-MM-DD HH:MM:SS!')),
    Field('komentar', 'text', required=True,
requires=IS_NOT_EMPTY()),
    Field('slika', 'upload'),
    Field('video', 'upload')
    )
db.define_table('ajokic',
    Field('kolegij', requires=IS_IN_SET(['Računalna
matematika'])),
    Field('naslov', 'string', unique=True, required=True,
requires=IS_NOT_EMPTY()),
    Field('datum', 'datetime', required=True,
requires=IS_DATETIME(format=T('%Y-%m-%d %H:%M:%S'), error_message='Mora
biti YYYY-MM-DD HH:MM:SS!')),
    Field('komentar', 'text', required=True,
requires=IS_NOT_EMPTY()),
    Field('slika', 'upload'),
    Field('video', 'upload')
    )

```

## 1.2. menu.py

```

response.logo = IMG(_src=URL('static','images/Logo.png'),
                    _style="height:50px; width:auto; position:right; ",
                    _alt=T('FSB-logo'))
response.title = request.application.replace('_', ' ').title()
response.subtitle = ''

response.meta.author = 'Your Name <you@example.com>'
response.meta.description = 'a cool new app'
response.meta.keywords = 'web2py, python, framework'
response.meta.generator = 'Web2py Web Framework'

response.google_analytics_id = None

DEVELOPMENT_MENU = True

def _():
    app = request.application
    ctr = request.controller
    response.menu += [
        (T('Studenti'), False, '#', [
            (T('Repozitorij slika'), False, URL('Zavrsni', 'default',
'student')),
            (T('Repozitorij videa'), False, URL('Zavrsni', 'default',
'student2')),
        ]),
        (T('Profesori'), False, '#', [
            (T('Mario Essert'), False, URL('Zavrsni', 'default', 'messert'
)),
            (T('Tihomir Žilić'), False, URL('Zavrsni', 'default', 'tzilic'
)),
            (T('Vladimir Milić'), False, URL('Zavrsni', 'default',
'vmillic' )),
            (T('Danijel Pavković'), False, URL('Zavrsni', 'default',
'dpavkovic' )),
            (T('Andrej Jokić'), False, URL('Zavrsni', 'default', 'ajokic'
)),
        ]),
    ]

if DEVELOPMENT_MENU: _()

if "auth" in locals(): auth.wikimenu()

```

## 1.3. default.py

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

@auth.requires_login()
def student():
    response.title = "Repozitorij slika"
    return dict()

@auth.requires_login()
def student2():
    response.title = "Repozitorij video uradaka"
    return dict()

@auth.requires(auth.user_id==7 , requires_login=True)
def messert():
    response.title = "Mario Essert"

    forma=SQLFORM(db.messert, submit_button="Objavi")

    if forma.process(keepvalues=False).accepted:
        response.flash="Uspješno objavljeno!"
    elif forma.errors:
        response.flash="Došlo je do greške"
    return dict(forma=forma)

@auth.requires(auth.user_id==7 , requires_login=True)
def messertedit():
    response.title = "Uređivanje objave"
    Id=eval(request.vars.Id)
    rekord=db(db.messert.id==Id).select().first()
    forma=SQLFORM(db.messert, record=rekord, submit_button="Promijeni",
delete_label="Obriši", deletable=True)
    if forma.process().accepted:
        session.flash="Objava je uspješno promijenjena!"
        redirect(URL('messert'))
    elif forma.errors:
        response.flash="Došlo je do greške!"
    return dict(forma=forma)

@auth.requires(auth.user_id==8 , requires_login=True)
def tzilic():
    response.title = "Tihomir Žilić"

    forma=SQLFORM(db.tzilic, submit_button="Objavi")

    if forma.process(keepvalues=False).accepted:
        response.flash="Uspješno objavljeno!"
    elif forma.errors:
        response.flash="Došlo je do greške"
    return dict(forma=forma)
    return dict()

@auth.requires(auth.user_id==8 , requires_login=True)
def tzilicedit():
    response.title = "Uređivanje objave"
    Id=eval(request.vars.Id)
```

```
rekord=db(db.tzilic.id==Id).select().first()
forma=SQLFORM(db.tzilic, record=rekord, submit_button="Promijeni",
delete_label="Obriši", deletable=True)

if forma.process().accepted:
    session.flash="Objava je uspješno promijenjena!"
    redirect(URL('tzilic'))
elif forma.errors:
    response.flash="Došlo je do greške!"
return dict(forma=forma)

@auth.requires(auth.user_id==9 , requires_login=True)
def vmilic():
    response.title = "Vladimir Milić"

    forma=SQLFORM(db.vmilic, submit_button="Objavi")

    if forma.process(keepvalues=False).accepted:
        response.flash="Uspješno objavljeno!"
    elif forma.errors:
        response.flash="Došlo je do greške"
    return dict(forma=forma)
    return dict()

@auth.requires(auth.user_id==9 , requires_login=True)
def vmilicedit():
    response.title = "Uređivanje objave"
    Id=eval(request.vars.Id)
    rekord=db(db.vmilic.id==Id).select().first()
    forma=SQLFORM(db.vmilic, record=rekord, submit_button="Promijeni",
delete_label="Obriši", deletable=True)

    if forma.process().accepted:
        session.flash="Objava je uspješno promijenjena!"
        redirect(URL('vmilic'))
    elif forma.errors:
        response.flash="Došlo je do greške!"
    return dict(forma=forma)

@auth.requires(auth.user_id==11 , requires_login=True)
def dpavkovic():
    response.title = "Danijel Pavković"

    forma=SQLFORM(db.dpavkovic, submit_button="Objavi")

    if forma.process(keepvalues=False).accepted:
        response.flash="Uspješno objavljeno!"
    elif forma.errors:
        response.flash="Došlo je do greške"
    return dict(forma=forma)
    return dict()

@auth.requires(auth.user_id==11 , requires_login=True)
def dpavkovicedit():
    response.title = "Uređivanje objave"
    Id=eval(request.vars.Id)
    rekord=db(db.dpavkovic.id==Id).select().first()
    forma=SQLFORM(db.dpavkovic, record=rekord, submit_button="Promijeni",
delete_label="Obriši", deletable=True)
```

```

    if forma.process().accepted:
        session.flash="Objava je uspješno promijenjena!"
        redirect(URL('dpavkovic'))
    elif forma.errors:
        response.flash="Došlo je do greške!"
    return dict(forma=forma)

@auth.requires(auth.user_id==10 , requires_login=True)
def ajokic():
    response.title = "Andrej Jokić"

    forma=SQLFORM(db.ajokic, submit_button="Objavi")

    if forma.process(keepvalues=False).accepted:
        response.flash="Uspješno objavljeno!"
    elif forma.errors:
        response.flash="Došlo je do greške"
    return dict(forma=forma)
    return dict()

@auth.requires(auth.user_id==10 , requires_login=True)
def ajokicedit():
    response.title = "Uređivanje objave"
    Id=eval(request.vars.Id)
    rekord=db(db.ajokic.id==Id).select().first()
    forma=SQLFORM(db.ajokic, rekord=rekord, submit_button="Promijeni",
delete_label="Obriši", deletable=True)

    if forma.process().accepted:
        session.flash="Objava je uspješno promijenjena!"
        redirect(URL('ajokic'))
    elif forma.errors:
        response.flash="Došlo je do greške!"
    return dict(forma=forma)

def user():
    """
    exposes:
    http://.../\[app\]/default/user/login
    http://.../\[app\]/default/user/logout
    http://.../\[app\]/default/user/register
    http://.../\[app\]/default/user/profile
    http://.../\[app\]/default/user/retrieve\_password
    http://.../\[app\]/default/user/change\_password
    http://.../\[app\]/default/user/manage\_users (requires membership in
    http://.../\[app\]/default/user/bulk\_register
    use @auth.requires_login()
        @auth.requires_membership('group name')
        @auth.requires_permission('read','table name',record_id)
    to decorate functions that need access control
    """
    return dict(form=auth())

@cache.action()
def download():
    """
    allows downloading of uploaded files
    http://.../\[app\]/default/download/\[filename\]
    """

```



```
return response.download(request, db)
```

```
def call():  
    """  
    exposes services. for example:  
    http://.../\[app\]/default/call/jsonrpc  
    decorate with @services.jsonrpc the functions to expose  
    supports xml, json, xmlrpc, jsonrpc, amfrpc, rss, csv  
    """  
    return service()
```

## 1.4. default/indeks.html

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

{{block head}}

<style type="text/css">

    .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: #FFFFFF;
border: 1px solid navy;
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: #3D96C5;
float: none;
padding: 15px 15px;
text-decoration: none;
font-family: "Palatino Linotype", Palatino, sans-serif;
font-size: 20px;
}

.dropdown-menu li > a {
display: block;
padding: 3px 20px;
clear: both;
font-weight: normal;
line-height: 20px;
color: #69ADD2;
white-space: nowrap;
font-family: "Palatino Linotype", Palatino, sans-serif;
font-size: 20px
}

.footer {
background-color: white;
height: 20px;
```

```
    }
</style>

<style>
  div#frame {
    background: white;
    border: 3px solid white;
    padding: 100px;
    margin: auto;
    width: 40%;
    height: 30%;
    position: relative;
    top: 80px;
  }
</style>
{{end}}

<div id="frame" style="font-family:Palatino; color:#3D96C5; font-size:20px"
>
  {{extend 'layout.html'}}
<h2>{{=T( request.args(0).replace('_', ' ').capitalize() )}}</h2>
<div id="web2py_user_form" >
  {{=form}}
  {{if request.args(0)=='login':}}
    {{if not 'register' in auth.settings.actions_disabled:}}
      {{pass}}
    {{if not 'request_reset_password' in auth.settings.actions_disabled:}}
      <br/>
      <a href="{{=URL(args='request_reset_password')}}">Zaboravili ste
lozinku?</a>
      {{pass}}
    {{pass}}
</div>
</div>
```

## 1.5. default/messert.html

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

<style type="text/css">

    .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: #FFFFFF;
        border: 1px solid navy;
        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: #3D96C5;
        float: none;
        padding: 15px 15px;
        text-decoration: none;
        font-family: "Palatino Linotype", Palatino, sans-serif;
        font-size: 20px;
    }

    .dropdown-menu li > a {
        display: block;
        padding: 3px 20px;
        clear: both;
        font-weight: normal;
        line-height: 20px;
        color: #69ADD2;
        white-space: nowrap;
        font-family: "'Palatino Linotype", Palatino, sans-serif;
        font-size: 20px
    }

    .footer {
        background-color: white;
        height: 20px;
    }

</style>
```

```

<style>
.tekst{
    box-shadow:2px 2px 8px rgba(0,0,0,0.5);
    border-radius: 5px;
    background: gainsboro;
    margin:10px;
    padding:10px;
}
.tekst table{
    width:100%;
    font-size:16px;
}
    .tekst td{
        padding: 5px;
        width:40%;
    }
</style>

<div style="margin:auto; width:80%; box-shadow:2px 2px 8px #cce0ff;
padding:20px; margin-top:30px; margin-bottom:50px;">
    <h3 style="border-bottom:2px solid #3D96C5;">
        <b>Unos nove slike/videoa:</b>
    </h3>
    <div style="width:50%;">
        {{=forma}}
    </div>
    <hr>
    <h3>
        <b> Dosadašnje objave:</b>
    </h3>
    <div>
        {{if db(db.messert.id>0).isempty()==False:}}
        {{for redak in db(db.messert.id>0).select():}}
        <div class="tekst">
            <table>
                <tr>
                    <td><b style="color: #003380; font-size:20px; text-
decoration: underline ">{{=redak.kolegij}}</b></td><td align="right">
                </tr>
                <tr>
                    <td colspan="1" valign="top"><div style="min-
height:25px;">
                        <b>{{=redak.naslov}}</b>
                    </div></td>
                </tr>
                <tr>
                    <td colspan="2" valign="top"><div style="min-
height:50px;">
                        {{=redak.komentar}}
                    </div></td>
                </tr>
                <tr>
                    <td colspan="2" valign="top"><div style="min-
height:75px;">
                        <br>
                        <a href="{{=URL(f='download',
args=redak.slika)}}">[Preuzmi sliku]</a>
                    </div></td>

```

```
        </tr>
        <tr>
            <td colspan="2" valign="top"><div style="min-
height:75px;">
                <video style="width:300px" src="{=URL('download',
args=redak.video)}" controls></video><br>
                </div></td>
        </tr>
        <tr>
            <td align="left"><b>Autor: </b>Mario Essert</td>
            <td align="center"><i>Objavljeno:
</i>{=redak.datum}</td>
            <td align="right"><a
href="{=URL('default','messertedit',
vars=dict(Id=redak.id))}">Uredi</a></td>
        </tr>
    </table>
</div>
<br>
{{pass}}
{{else:}},
<p>
    Trenutno ne postoji niti jedana objava!
</p>
{{pass}}
</div>
</div>
```

## 1.6. default/messertedit.html

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

<style type="text/css">

    .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: #FFFFFF;
        border: 1px solid navy;
        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: #3D96C5;
        float: none;
        padding: 15px 15px;
        text-decoration: none;
        font-family: "Palatino Linotype", Palatino, sans-serif;
        font-size: 20px;
    }

    .dropdown-menu li > a {
        display: block;
        padding: 3px 20px;
        clear: both;
        font-weight: normal;
        line-height: 20px;
        color: #69ADD2;
        white-space: nowrap;
        font-family: "'Palatino Linotype", Palatino, sans-serif;
        font-size: 20px
    }

    .footer {
        background-color: white;
        height: 20px;
    }

</style>
```

```
<div style="margin:auto; width:80%; box-shadow:2px 2px 8px #cce0ff;
padding:20px; margin-top:30px; margin-bottom:50px;">
  <h3 style="border-bottom:2px solid #0066ff;">
    <b>Uređivanje objave:</b>
  </h3>
  <div style="width:50%;">
    {=forma}
  </div>
```



## 1.7. default/student.html

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

<style type="text/css">

    .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: #FFFFFF;
        border: 1px solid navy;
        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: #3D96C5;
        float: none;
        padding: 15px 15px;
        text-decoration: none;
        font-family: "Palatino Linotype", Palatino, sans-serif;
        font-size: 20px;
    }

    .dropdown-menu li > a {
        display: block;
        padding: 3px 20px;
        clear: both;
        font-weight: normal;
        line-height: 20px;
        color: #69ADD2;
        white-space: nowrap;
        font-family: "Palatino Linotype", Palatino, sans-serif;
        font-size: 20px
    }

    .footer {
        background-color: white;
        height: 20px;
    }

</style>
```

```
<style>
.tekst{
    box-shadow:2px 2px 8px rgba(0,0,0,0.5);
    border-radius: 5px;
    background: gainsboro;
    margin:10px;
    padding:10px;
    width: 100%;
}
</style>

<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {font-family: Palatino Linotype;}

.tab {
    overflow: hidden;
    border: 1px solid #ccc;
    background-color: #f1f1f1;
}

.tab button {
    background-color: inherit;
    float: left;
    border: none;
    outline: none;
    cursor: pointer;
    padding: 14px 16px;
    transition: 0.3s;
    font-size: 17px;
}

.tab button:hover {
    background-color: #ddd;
}

.tab button.active {
    background-color: #ccc;
}

.tabcontent {
    display: none;
    padding: 6px 12px;
    border: 1px solid #ccc;
    border-top: none;
}
</style>
</head>
<body>

<h1 style="font-size:23px">
    <b>Repozitorij slika</b>
</h1>

<p>Odaberi željeni kolegij, zatim odaberi željenog profesora:</p>

<div class="tab">
    <button class="tablinks" onclick="otvorikolegij(event,
'elektroteh')">Elektrotehnika</button>
    <button class="tablinks" onclick="otvorikolegij(event, 'op')">Objektno
programiranje</button>
```

```

    <button class="tablinks" onclick="otvorikolegij(event,
'racmat')">Računalna matematika</button>
    <button class="tablinks" onclick="otvorikolegij(event, 'webp')">WEB
programiranje</button>
</div>

<div id="elektroteh" class="tabcontent">
    <div class="tab">
        <button class="tablinks" onclick="otvoriprof(event,
'elektromessert')">Mario Essert</button>
        <button class="tablinks" onclick="otvoriprof(event,
'elektrotzilic')">Tihomir Žilić</button>
        <button class="tablinks" onclick="otvoriprof(event,
'elektropavkovic')">Danijel Pavković</button>
    </div>
</div>

<div id="elektromessert" class="tabcontent" >
    {{if db(db.messert.kolegij=='Elektrotehnika').isempty()==False:}}
    {{for redak in db(db.messert.kolegij=='Elektrotehnika' and
db.messert.video=='').select():}}
        <div class="tekst">
            <table>
                <tr>
                    <th width=50%><b style="color: #003380; font-size:25px;
text-decoration: underline ">{{=redak.kolegij}}</b><b style="font-
size:20px"> - Mario Essert</b>
                    <p>Objavljeno: {{=redak.datum}}</p>
                    <p style="min-height:25px; font-size:20px
"><b>{{=redak.naslov}}</b></p>
                    <p style="min-height:50px;font-
size:15px">{{=redak.komentar}}</p>
                </th>
                <th width=50%><div style="min-height:75px">
                    <br>
                    <a href="{{=URL(f='download',
args=redak.slika)}}">[Preuzmi sliku]</a>
                </div></th>
            </tr>
        </table>
        <br>
        {{pass}}
        {{else:}}
        <p>
            Trenutno ne postoji niti jedana objava!
        </p>
        {{pass}}
    </div>

<div id="elektrotzilic" class="tabcontent" >
    {{if db(db.tzilic.kolegij=='Elektrotehnika').isempty()==False:}}
    {{for redak in db(db.tzilic.kolegij=='Elektrotehnika' and
db.tzilic.video=='').select():}}
        <div class="tekst">
            <table>
                <tr>

```

```

        <th width=50%><b style="color: #003380; font-size:25px;
text-decoration: underline ">{{=redak.kolegij}}</b><b style="font-
size:20px"> - Tihomir Žilić</b>
        <p>Objavljeno: {{=redak.datum}}</p>

        <p style="min-height:25px; font-size:20px
"><b>{{=redak.naslov}}</b></p>
        <p style="min-height:50px;font-
size:15px">{{=redak.komentar}}</p>
        </th>
        <th width=50%><div style="min-height:75px">
        <br>
        <a href="{{=URL(f='download',
args=redak.slika)}}">[Preuzmi sliku]</a>
        </div></th>
    </tr>
</table>
</div>
<br>
{{pass}}
{{else:}}
<p>
    Trenutno ne postoji niti jedana objava!
</p>
{{pass}}
</div>

<div id="elektropavkovic" class="tabcontent" >
    {{if db(db.dpavkovic.kolegij=='Elektrotehnika').isempty()==False:}}
    {{for redak in db(db.dpavkovic.kolegij=='Elektrotehnika' and
db.dpavkovic.video=='').select():}}
        <div class="tekst">
            <table>
                <tr>
                    <th width=50%><b style="color: #003380; font-size:25px;
text-decoration: underline ">{{=redak.kolegij}}</b><b style="font-
size:20px"> - Danijel Pavković</b>
                    <p>Objavljeno: {{=redak.datum}}</p>

                    <p style="min-height:25px; font-size:20px
"><b>{{=redak.naslov}}</b></p>
                    <p style="min-height:50px;font-
size:15px">{{=redak.komentar}}</p>
                    </th>
                    <th width=50%><div style="min-height:75px">
                        <br>
                        <a href="{{=URL(f='download',
args=redak.slika)}}">[Preuzmi sliku]</a>
                        </div></th>
                </tr>
            </table>
        </div>
        <br>
        {{pass}}
        {{else:}}
        <p>
            Trenutno ne postoji niti jedana objava!
        </p>
    </div>

```

```

        {{pass}}
    </div>

<div id="op" class="tabcontent">
    <div class="tab">
        <button class="tablinks" onclick="otvoriprof(event, 'opmessert')">Mario
        Essert</button>
        <button class="tablinks" onclick="otvoriprof(event, 'optzilic')">Tihomir
        Žilić</button>
        <button class="tablinks" onclick="otvoriprof(event, 'opvmilic')">Vladimir
        Milić</button>
    </div>
</div>

<div id="opmessert" class="tabcontent" >
    {{if db(db.messert.kolegij=='Objektno
programiranje').isempty()==False:}}
    {{for redak in db(db.messert.kolegij=='Objektno programiranje' and
db.messert.video=='').select():}}
        <div class="tekst">
            <table>
                <tr>
                    <th width=50%><b style="color: #003380; font-size:25px;
text-decoration: underline ">{{=redak.kolegij}}</b><b style="font-
size:20px"> - Mario Essert</b>
                    <p>Objavljeno: {{=redak.datum}}</p>
                    <p style="min-height:25px; font-size:20px
"><b>{{=redak.naslov}}</b></p>
                    <p style="min-height:50px;font-
size:15px">{{=redak.komentar}}</p>
                </th>
                <th width=50%><div style="min-height:75px">
                    <br>
                    <a href="{{=URL(f='download',
args=redak.slika)}}">[Preuzmi sliku]</a>
                </div></th>
            </tr>
        </table>
    </div>
    <br>
    {{pass}}
    {{else:}}
    <p>
        Trenutno ne postoji niti jedana objava!
    </p>
    {{pass}}
</div>

<div id="optzilic" class="tabcontent" >
    {{if db(db.tzilic.kolegij=='Objektno
programiranje').isempty()==False:}}
    {{for redak in db(db.tzilic.kolegij=='Objektno programiranje' and
db.tzilic.video=='').select():}}
        <div class="tekst">
            <table>
                <tr>

```

```

        <th width=50%><b style="color: #003380; font-size:25px;
text-decoration: underline ">{{=redak.kolegij}}</b><b style="font-
size:20px"> - Tihomir Žilić</b>
        <p>Objavljeno: {{=redak.datum}}</p>

        <p style="min-height:25px; font-size:20px
"><b>{{=redak.naslov}}</b></p>
        <p style="min-height:50px;font-
size:15px">{{=redak.komentar}}</p>
        </th>
        <th width=50%><div style="min-height:75px">
        <br>
        <a href="{{=URL(f='download',
args=redak.slika)}}">[Preuzmi sliku]</a>
        </div></th>
    </tr>
</table>
</div>
<br>
{{pass}}
{{else:}}
<p>
    Trenutno ne postoji niti jedana objava!
</p>
{{pass}}
</div>

<div id="opvmilic" class="tabcontent" >
    {{if db(db.vmilic.kolegij=='Objektno
programiranje').isempty()==False:}}
    {{for redak in db(db.vmilic.kolegij=='Objektno programiranje' and
db.vmilic.video=='').select():}}

        <div class="tekst">
            <table>
                <tr>
                    <th width=50%><b style="color: #003380; font-size:25px;
text-decoration: underline ">{{=redak.kolegij}}</b><b style="font-
size:20px"> - Vladimir Milić</b>
                    <p>Objavljeno: {{=redak.datum}}</p>

                    <p style="min-height:25px; font-size:20px
"><b>{{=redak.naslov}}</b></p>
                    <p style="min-height:50px;font-
size:15px">{{=redak.komentar}}</p>
                    </th>
                    <th width=50%><div style="min-height:75px">
                        <br>
                        <a href="{{=URL(f='download',
args=redak.slika)}}">[Preuzmi sliku]</a>
                        </div></th>
                </tr>
            </table>
        </div>
        <br>
        {{pass}}
        {{else:}}
        <p>

```

```

        Trenutno ne postoji niti jedana objava!
    </p>
    {{pass}}
</div>

<div id="racmat" class="tabcontent">
    <div class="tab">
        <button class="tablinks" onclick="otvoriprof(event,
'racmatmesseract')">Mario Essert</button>
        <button class="tablinks" onclick="otvoriprof(event,
'racmattzilic')">Tihomir Žilić</button>
        <button class="tablinks" onclick="otvoriprof(event,
'racmatvmilic')">Vladimir Milić</button>
        <button class="tablinks" onclick="otvoriprof(event,
'racmatvmilic')">Andrej Jokić</button>
    </div>
</div>

<div id="racmatmesseract" class="tabcontent" >
    {{if db(db.messeract.kolegij=='Računalna
matematika').isempty()==False:}}
    {{for redak in db(db.messeract.kolegij=='Računalna matematika' and
db.messeract.video=='').select():}}
        <div class="tekst">
            <table>
                <tr>
                    <th width=50%><b style="color: #003380; font-size:25px;
text-decoration: underline ">{{=redak.kolegij}}</b><b style="font-
size:20px"> - Mario Essert</b>
                    <p>Objavljeno: {{=redak.datum}}</p>
                    <p style="min-height:25px; font-size:20px
"><b>{{=redak.naslov}}</b></p>
                    <p style="min-height:50px;font-
size:15px">{{=redak.komentar}}</p>
                </th>
                <th width=50%><div style="min-height:75px">
                    <br>
                    <a href="{{=URL(f='download' ,
args=redak.slika)}}">[Preuzmi sliku]</a>
                </div></th>
                </tr>
            </table>
        </div>
        <br>
        {{pass}}
        {{else:}}
        <p>
            Trenutno ne postoji niti jedana objava!
        </p>
        {{pass}}
    </div>

<div id="racmattzilic" class="tabcontent" >
    {{if db(db.tzilic.kolegij=='Računalna
matematika').isempty()==False:}}
    {{for redak in db(db.tzilic.kolegij=='Računalna matematika' and
db.tzilic.video=='').select():}}

```

```

    <div class="tekst">
        <table>
            <tr>
                <th width=50%><b style="color: #003380; font-size:25px;
text-decoration: underline ">{{=redak.kolegij}}</b><b style="font-
size:20px"> - Tihomir Žilić</b>
                    <p>Objavljeno: {{=redak.datum}}</p>

                    <p style="min-height:25px; font-size:20px
"><b>{{=redak.naslov}}</b></p>
                    <p style="min-height:50px;font-
size:15px">{{=redak.komentar}}</p>
                </th>
                <th width=50%><div style="min-height:75px">
                    <br>
                    <a href="{{=URL(f='download',
args=redak.slika)}}">[Preuzmi sliku]</a>
                </div></th>
            </tr>
        </table>
    </div>
    <br>
    {{pass}}
    {{else:}}
    <p>
        Trenutno ne postoji niti jedana objava!
    </p>
    {{pass}}
</div>

<div id="racmatvmilic" class="tabcontent" >
    {{if db(db.vmilic.kolegij=='Računalna
matematika').isempty()==False:}}
    {{for redak in db(db.vmilic.kolegij=='Računalna matematika' and
db.vmilic.video=='').select():}}

        <div class="tekst">
            <table>
                <tr>
                    <th width=50%><b style="color: #003380; font-size:25px;
text-decoration: underline ">{{=redak.kolegij}}</b><b style="font-
size:20px"> - Vladimir Milić</b>
                        <p>Objavljeno: {{=redak.datum}}</p>

                        <p style="min-height:25px; font-size:20px
"><b>{{=redak.naslov}}</b></p>
                        <p style="min-height:50px;font-
size:15px">{{=redak.komentar}}</p>
                    </th>
                    <th width=50%><div style="min-height:75px">
                        <br>
                        <a href="{{=URL(f='download',
args=redak.slika)}}">[Preuzmi sliku]</a>
                    </div></th>
                </tr>
            </table>
        </div>
    <br>

```



```

        {{pass}}
        {{else:}}

        <p>
            Trenutno ne postoji niti jedana objava!
        </p>
        {{pass}}
    </div>

<div id="racmatajokic" class="tabcontent" >
    {{if db(db.ajokic.kolegij=='Računalna
matematika').isempty()==False:}}
    {{for redak in db(db.ajokic.kolegij=='Računalna matematika' and
db.ajokic.video=='').select():}}
    <div class="tekst">
        <table>
            <tr>
                <th width=50%><b style="color: #003380; font-size:25px;
text-decoration: underline ">{{=redak.kolegij}}</b><b style="font-
size:20px"> - Andrej Jokić</b>
                <p>Objavljeno: {{=redak.datum}}</p>

                <p style="min-height:25px; font-size:20px
"><b>{{=redak.naslov}}</b></p>
                <p style="min-height:50px;font-
size:15px">{{=redak.komentar}}</p>
            </th>
            <th width=50%><div style="min-height:75px">
                <br>
                <a href="{{=URL(f='download',
args=redak.slika)}}">[Preuzmi sliku]</a>
            </div></th>
        </tr>
    </table>
    </div>
    <br>
    {{pass}}
    {{else:}}
    <p>
        Trenutno ne postoji niti jedana objava!
    </p>
    {{pass}}
</div>

<div id="webp" class="tabcontent">
    <div class="tab">
        <button class="tablinks" onclick="otvoriprof(event, 'webpmessert')">Mario
Essert</button>
        <button class="tablinks" onclick="otvoriprof(event,
'webptzilic')">Tihomir Žilić</button>
    </div>
</div>

<div id="webpmessert" class="tabcontent" >
    {{if db(db.messert.kolegij=='WEB
programiranje').isempty()==False:}}
    {{for redak in db(db.messert.kolegij=='WEB programiranje' and
db.messert.video=='').select():}}

```

```

    <div class="tekst">
        <table>
            <tr>
                <th width=50%><b style="color: #003380; font-size:25px;
text-decoration: underline ">{{=redak.kolegij}}</b><b style="font-
size:20px"> - Mario Essert</b>
                <p>Objavljeno: {{=redak.datum}}</p>

                <p style="min-height:25px; font-size:20px
"><b>{{=redak.naslov}}</b></p>
                <p style="min-height:50px;font-
size:15px">{{=redak.komentar}}</p>
            </th>
            <th width=50%><div style="min-height:75px">
                <br>
                <a href="{{=URL(f='download',
args=redak.slika)}}">[Preuzmi sliku]</a>
            </div></th>
        </tr>
    </table>
</div>
<br>
{{pass}}
{{else:}}
<p>
    Trenutno ne postoji niti jedana objava!
</p>
{{pass}}
</div>

<div id="webptzilic" class="tabcontent" >
    {{if db(db.tzilic.kolegij=='WEB programiranje').isempty()==False:}}
    {{for redak in db(db.tzilic.kolegij=='WEB programiranje' and
db.tzilic.video=='').select():}}
        <div class="tekst">
            <table>
                <tr>
                    <th width=50%><b style="color: #003380; font-size:25px;
text-decoration: underline ">{{=redak.kolegij}}</b><b style="font-
size:20px"> - Tihomir Žilić</b>
                    <p>Objavljeno: {{=redak.datum}}</p>

                    <p style="min-height:25px; font-size:20px
"><b>{{=redak.naslov}}</b></p>
                    <p style="min-height:50px;font-
size:15px">{{=redak.komentar}}</p>
                </th>
                <th width=50%><div style="min-height:75px">
                    <br>
                    <a href="{{=URL(f='download',
args=redak.slika)}}">[Preuzmi sliku]</a>
                </div></th>
            </tr>
        </table>
    </div>
    <br>
    {{pass}}
    {{else:}}

```

```
<p>
    Trenutno ne postoji niti jedana objava!
</p>

    {{pass}}
</div>

<script>
function otvorikolegij(evt, kolegij) {
    var i, tabcontent, tablinks;
    tabcontent = document.getElementsByClassName("tabcontent");
    for (i = 0; i < tabcontent.length; i++) {
        tabcontent[i].style.display = "none";
    }
    tablinks = document.getElementsByClassName("tablinks");
    for (i = 0; i < tablinks.length; i++) {
        tablinks[i].className = tablinks[i].className.replace(" active",
    "");
    }
    document.getElementById(kolegij).style.display = "block";
    evt.currentTarget.className += " active";
}
</script>

<script>
function otvoriprof(evt, prof) {
    var i, tabcontent, tablinks;
    tabcontent = document.getElementsByClassName("tabcontent");
    for (i = 0; i < tabcontent.length; i++) {
        tabcontent[i].style.display = "none";
    }
    tablinks = document.getElementsByClassName("tablinks");
    for (i = 0; i < tablinks.length; i++) {
        tablinks[i].className = tablinks[i].className.replace(" active",
    "");
    }
    document.getElementById(prof).style.display = "block";
    evt.currentTarget.className += " active";
}
</script>
</body>
```