

Aplikacija za evidenciju održane nastave

Bastijanić, Luka

Undergraduate thesis / Završni rad

2016

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Pula / Sveučilište Jurja Dobrile u Puli**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:137:821309>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-04-19**



Repository / Repozitorij:

[Digital Repository Juraj Dobrila University of Pula](#)



Sveučilište Jurja Dobrile u Puli
Odjel za informacijsko-komunikacijske tehnologije

LUKA BASTIJANIĆ

APLIKACIJA ZA EVIDENCIJU ODRŽANE NASTAVE

Završni rad

Pula, 2016. godine

Sveučilište Jurja Dobrile u Puli
Odjel za informacijsko-komunikacijske tehnologije

LUKA BASTIJANIĆ

APLIKACIJA ZA EVIDENCIJU ODRŽANE NASTAVE

Završni rad

JMBAG: 0303047679, redoviti student

Studijski smjer: preddiplomski studij Informatika

Predmet: Programiranje

Znanstveno područje: Društvene znanosti

Znanstveno polje: Informacijske i komunikacijske znanosti

Znanstvena grana: Informacijski sustavi i informatologija

Mentor: Doc. dr. sc. Tihomir Orehovački

Pula, rujan, 2016. godine



IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisani Luka Bastijanić, kandidat za prvostupnika informatike ovime izjavljujem da je ovaj Završni rad rezultat isključivo mogega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Završnog rada nije napisan na nedozvoljen način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student

U Puli, 19. rujna, 2016. godine



IZJAVA o korištenju autorskog djela

Ja, Luka Bastijanić dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj završni rad pod nazivom „Aplikacija za evidenciju održane nastave“ koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama.

Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, 19.09.2016.

Potpis

Sadržaj

1. UVOD	1
2. Evidencija održane nastave	3
2.1 Općenito o evidencijama	3
2.2 O evidenciji održane nastave	3
3. Aplikacija za evidenciju održane nastave	5
3.1 Motivacija za izradu aplikacije	5
3.2 Osnovni zahtjevi na sustav	5
3.3 Postojeće aplikacije	6
3.3.1 Aplikacija „Nastava“	6
3.3.2 Sustav „Quilt CMS“	7
3.3.3 EduPlan/EX	9
3.3.4 Analiza i usporedba postojećih aplikacija s razvijenom	9
4. Okruženje i tehnologije za razvoj aplikacije	11
4.1 Tipovi web stranica	11
4.1.1 Statičke web stranice	11
4.1.2 Dinamičke web stranice	12
4.2 Web aplikacije	13
4.2.1 Sustavi za upravljanje bazom podataka	13
4.2.2 Programski i skriptni jezici za razvoj web aplikacija	14
4.3 Korištene tehnologije za razvoj aplikacije	14
4.3.1 Hypertext Markup Language	15
4.3.2 Cascading Style Sheets	16
4.3.3 PHP: Hypertext Preprocessor	17
4.3.4 MySQL	19
4.3.5 JavaScript	21
4.3.6 jQuery	22

5. Hodogram korištenja aplikacije	23
6. Programsko rješenje	24
6.1 Baza podataka	24
6.1.1 Tablica Titula	25
6.1.2 Tablica Korisnik	25
6.1.3 Tablica Tip_predavanja	25
6.1.4 Tablica Studij	26
6.1.5 Tablica Kolegij	26
6.1.6 Tablica Korisnik_kolegij_tip	26
6.1.7 Tablica Evidencija.....	27
6.2 Rješenja specifičnih problema	27
6.2.1 Dodjeljivanje identifikacijskog broja korisniku	27
6.2.2 Generiranje lozinke.....	27
6.2.3 Automatizirano slanje e-mail-a	28
6.2.4 Prikaz listi	29
6.2.5 Provjera popunjenosti polja u obrascima.....	29
6.2.6 Dinamički kalendar	31
6.2.7 Izračun prekovremenih sati	31
6.2.8 Generiranje PDF izvješća	32
6.3 Korištenje web aplikacije.....	33
6.3.1 Korištenje web aplikacije u svojstvu administratora.....	38
7. Potencijalne buduće nadogradnje i poboljšanja sustava	42
7.1 Asynchronous JavaScript and XML	42
7.2 Generiranje više vrsta izvještaja	42
7.3 Integracija s bazom podataka rasporeda sati.....	42
8. ZAKLJUČAK.....	43
LITERATURA	44

POPIS SLIKA	46
POPIS KODA	47
SAŽETAK	48

1. UVOD

Sastavni dio svake obrazovne ustanove je nastava, zato je vrlo bitno voditi brigu u kojoj je mjeri ona održana. Stoga, cilj je omogućiti vođenje evidencije o održanoj nastavi svakoj visoko obrazovnoj ustanovi. Svrha takve evidencije je umanjiti potrebu za papirologijom, olakšati nastavnicima planiranje nastave, omogućiti obrazovnoj ustanovi uvid u održanu nastavu kao i kontrolu nad odrađenom normom pojedinog nastavnika.

Ovaj rad bavi se razvojem web aplikacije koja rješava taj problem, može služiti kao dobar način evidentiranja koliko je nastave održano ili kao kontrola u kojoj su mjeri nastavnici ispunili svoje nastavno opterećenje. Vanjski suradnici su tu posebno naglašeni jer oni nemaju propisano nastavno opterećenje već se njihov rad plaća po satu, stoga generirano izvješće njima predstavlja dokaz da su nastavu održali. Razvijena aplikacija prvenstveno je namijenjena visoko obrazovnim ustanovama, ali bi ju se vrlo lako moglo prilagoditi i ostalim tipovima obrazovnih ustanova poput osnovnih i srednjih škola. Aplikacija je namijenjena za korištenje većem broju korisnika svakodnevno.

Kroz rad je opisan teorijski dio koji obuhvaća opise termina poput evidencije i nastave, razradom kojih se dobiva uvid u ono što je problematika. Razrađeni su osnovni zahtjevi koje aplikacija mora zadovoljavati kako bi korisnici imali na raspolaganju sve potrebne funkcionalnosti. Predstavljena je analiza postojećih aplikacija na tržištu koje koriste druge obrazovne ustanove za osobne potrebe, te njihova usporedba s razvijenom aplikacijom. Obrađena je razlika između statičkih i dinamičkih web stranica kao i termin web aplikacije iz razloga što je odabrano da se aplikacija razvija za web okruženje. Pomno su odabrane i razrađene korištene tehnologije za razvoj aplikacije, među kojima su HTML, PHP i JavaScript. Prikazan je hodogram korištenja aplikacije koji pomaže pri boljem razumijevanju načina rada aplikacije.

Pored teorijskog dijela razrađen je i praktični dio razvoja aplikacije. Opisana je baza podataka i prikazana u ERD modelu, svaka tablica je posebno opisana kao i pripadajući atributi. Tijekom razvoja aplikacije bilo je potrebno koristiti specifična rješenja za pojedine probleme, poput generiranja lozinke ili izračuna prekovremenih

sati. Svaki taj specifični problem pomno je razrađen i mogu se vidjeti njihova rješenja. Detaljno je opisan način korištenja same aplikacije te je prikazan izgled dijelova aplikacije. Ujedno su predstavljena i moguća buduća poboljšanja i nadogradnje.

Rad je sastavljen od osam poglavlja, uvoda, šest glavnih poglavlja i zaključka. Drugo poglavlje sadrži opise osnovnih termina poput evidencije i nastave te njihovu razradu. Treće poglavlje posvećeno je osnovnim zahtjevima koje aplikacija mora zadovoljiti, opis postojećih aplikacija i usporedbu s razvijenom aplikacijom. Četvrto poglavlje opisuje tipove web stranica, definiciju web aplikacija i korištene tehnologije u razvoju aplikacije. Peto poglavlje sadrži hodogram korištenja aplikacije, dok šesto poglavlje opisuje razvijenu aplikaciju, odnosno njenu bazu podataka i rješenja specifičnih problema. Šesto poglavlje također sadrži opis aplikacije kao i način korištenja funkcionalnosti sustava. Sedmo poglavlje bavi se potencijalnim nadogradnjama i poboljšanjima sustava.

2. Evidencija održane nastave

2.1 Općenito o evidencijama

Evidencija je pojam koji označava pregled iz kojega se može iščitati da se nešto dogodilo, postojalo ili još uvijek postoji, te da je nešto napravljeno.

U svakodnevnoj upotrebi poznati su razni oblici evidencije, a najčešće oni koji se koriste u poslovne svrhe.

Neki od najčešćih oblika evidencije jesu:

1. **Evidencija radnog vremena** – bilježi dolaske, odlaske i tip rada (dnevni, noćni ili rad za praznik) svakog zaposlenika poduzeća te se naknadno prema njoj vrši isplata plaće.
2. **Evidencija zaposlenika** – sadrži popis svih bivših i sadašnjih zaposlenika s njihovim osobnim podacima poput imena, prezimena, datuma rođenja, mjesta stanovanja, itd.
3. **Evidencija zaliha** – prikazuje trenutno dostupne zalihe materijala na skladištu, te sadrži dokumente na kojima je vidljiv protok materijala u skladištu, poput primke (dokument koji bilježi zaprimljeni materijal) ili izdatnice (dokument koji bilježi materijal koji je izašao iz skladišta).
4. **Evidencija poslovanja** – pruža uvid u poslovanje poduzeća u svim segmentima, s ciljem dobivanja informacija o učinku poduzeća.
5. **Evidencija nestalih osoba** – služi kako bi se vodilo računa o broju i identitetu nestalih osoba.

2.2 O evidenciji održane nastave

Sastavni dio svake obrazovne ustanove (škole, fakulteta ili centra za edukaciju) je nastava.

Nastava je osnovni dio podučavanja, ona je organizirani napor koji služi za savladavanje propisanih sadržaja, stjecanje znanja, sposobnosti i navika te je njezin cilj osposobljavanje polaznika za rad u budućnosti.

Sastavni dijelovi nastave su: učenici (polaznici), nastavnik (prenositelj znanja), nastavna literatura i nastavni uvjeti (prostorija u kojoj se održava, materijali potrebni u nastavi i uređaji koji se koriste kao pomoć u nastavi).

Nastava se najčešće izvodi u obliku školskog sata koji traje 45 minuta, a definirana je rasporedom sati koji služi kao organizacijsko pomagalo kod održavanja nastave. Naime, nastavni program određuje kolegije i njihov broj sati koji će biti sadržani u rasporedu sati, dok raspored sati opisuje za pojedini kolegij tko će ga kada i kome održavati. Raspored sati je zbog toga osnovna polaznica za kreiranje evidencije održane nastave.

Svaki nastavnik u obrazovnoj ustanovi ima svoju normu (nastavno opterećenje) koju mora ispuniti u tekućoj školskoj ili akademskoj godini. Ona se definira ovisno o zvanju nastavnika, a posebno se boduje prema tipu nastave koja se održava.

Evidencija održane nastave primarno se koristi kako bi se vodilo računa da sve što je određeno nastavnim planom bude i ispunjeno. U nju se unose zapisi o odrađenom broju sati za dan nastave koji je određen rasporedom, a u tom obliku njome se prikazuje u kojoj mjeri je raspored za taj dan odrađen i postoje li nepravilnosti u smislu manjka ili viška odrađenih sati. Cilj evidencije je kontrola nastavnika, odnosno u kojoj mjeri je ispunio osobnu normu, te uvid u manjak sati koji nastavnik treba nadoknaditi kako polaznici ne bi bili zakinuti.

Evidenciju je potrebno voditi kako bi se mogli zbrajati održani sati, te kako bi se zbrajala odrađena norma sati. Na kraju godine provjerava se odrađenost norme i ukoliko postoji višak, nastavniku se isplaćuje naknada za prekovremeni rad.

3. Aplikacija za evidenciju održane nastave

Ovaj rad temeljen je na izradi aplikacije koja je upotrebljiva u visoko školskim ustanovama gdje je upotreba takve najpotrebnija, zbog visokog broja zvanja nastavnika, posebnom normiranju nastave, te vanjskih suradnika kojima se kroz uvid u evidenciju isplaćuje naknada za rad.

3.1 Motivacija za izradu aplikacije

Najveća motivacija za izradu pomagala za pomoć pri vođenju evidencije, u ovom slučaju aplikacije, je nedostatak vođenja evidencije u pojedinim ustanovama ili vršenja iste na neadekvatan način, poput vođenja evidencije na papiru. Takav tip vođenja evidencije ne predstavlja prihvatljiv tip organizacije iz razloga što se papiri mogu zagubiti, organiziranje istih obično zbog nemara ne bude izvršeno na primjeren način i oduzima puno vremena.

Izrada ovakve aplikacije u velikoj mjeri može pomoći kao organizacijski alat. Uštede vremena za razliku od klasičnih metoda vršenja evidencije su zamjetne, a i uvelike pridonosi kod organizacije unutar obrazovnih ustanova.

3.2 Osnovni zahtjevi na sustav

Kako bi mogla služiti svrsi, aplikacija mora zadovoljiti osnovne zahtjeve. U to ulaze određene funkcionalnosti koje moraju biti podržane, te podaci koji su potrebni za rad sa sustavom. Zahtjevi su sljedeći:

1. mogućnost upravljanja korisnicima koji koriste sustav
2. uređivanje kolegija
3. uređivanje normi po zvanjima i cijene prekovremenih sati
4. mogućnost odabira kolegija i tipa predavanja za koje će se vršiti evidencija
5. evidencija opterećenja nastavnika
6. unos zapisa u evidenciju o održanoj nastavi po datumu, kolegiju i tipu nastave.

3.3 Postojeće aplikacije

Na tržištu postoji niz aplikacija koje obavljaju zadaću evidentiranja održane nastave, no to su većinom cjelovita rješenja koja uključuju izradu rasporeda, evidenciju svih zaposlenika, integraciju s vanjskim sustavima, planiranje nastave i mnoge druge funkcionalnosti. Također, postoje i aplikacije koje nisu komercijalne, odnosno razvijene su isključivo za zainteresiranu obrazovnu ustanovu, a najčešće od strane zaposlenika ili studenata. Iz tog razloga nije moguće dobiti uvid u njih jer su to interni sustavi tih ustanova.

U svrhu izrade aplikacije potrebno je proučiti i neka postojeća rješenja, kako ne bi izostavili neku od funkcionalnosti sustava, te kako bi uočili moguće nedostatke koje bi u vlastitoj aplikaciji mogli ispraviti.

3.3.1 Aplikacija „Nastava“

Web aplikacija „Nastava“ razvijena je isključivo za Fakultet organizacije i informatike Varaždin, a ima namjenu vođenja evidencije nastave i kreiranja rasporeda. Razvijena je uz pomoć niza tehnologija kao što su Groovy, Grails, GSP, HTML5, Javascript, Ajax, JQuery, CSS, PostgreSQL, Hibernate i Tomcat (Grd, 2016).

Sustav odlikuju sljedeće značajke:

- Korisniku je dodijeljena jedna od više mogućih uloga u sustavu (student, nastavnik, pročelnik katedre, administrator, superadmin)
- Određivanje ukupne norme sati za svakog pojedinog korisnika u određenoj akademskoj godini
- Omogućeno planiranje nastave, nositelj kolegija odabire nastavnike na kolegiju dok svaki nastavnik planira sate za taj kolegij
- Planiranje kolokvija
- Unos održane nastave u evidenciju
- Mogućnost pregleda različitih vrsta izvještaja ovisno o ulozi u sustavu
- Unos akademskog kalendara
- Kreiranje rasporeda nastave.

Aplikaciju „Nastava“ odlikuje vrlo atraktivan dizajn i preglednost. Na slici 3.1 vidljiv je pregled norme sati za sve korisnike sustava, te moguća navigacija po akademskim godinama kao i pretraživanje korisnika. Na slici 3.2 može vidjeti prikaz planirane

nastave za nastavnika te mogućnost generiranje izvješća.

Norma sati

Plan

Održana nastava

Izvještaji

Administracija

Norma sati

Odaberite akademsku godinu

2013/2014

Dohvati

Prikazuje se 10 zapisa po stranici

Korisnik	Norma sati	Akademski godina
	0	2013/2014
	450	2013/2014
	300	2013/2014
	300	2013/2014
	300	2013/2014
	225	2013/2014
	150	2013/2014
	0	2013/2014
	150	2013/2014
	0	2013/2014

Prikazuje se od 71. do 80. od ukupno 165 zapisa

Prva

Slika 3.1 Pregled korisnika i njihovih normi sati za određenu akademsku godinu
(Grd, 2016)

3.3.2 Sustav „Quilt CMS“

Sustav „Quilt CMS“ dio je e-Campus rješenja specifično napravljenog za potrebe akademskog osoblja, studenata i institucija. Razvijan je na Fakultetu elektrotehnike i računarstva Sveučilišta u Zagrebu od 2001. godine, a njegov razvoj i održavanje traje još i danas. Za razvoj su korištene tehnologije otvorenog koda. Velika prednost ovog sustava je integracija sa sustavima poput ISVU-a i AAI@Edu.hr s kojih prikuplja potrebne podatke za rad (informacije o nastavnicima, djelatnicima, studentima te drugim dostupnim sustavima integriranim u e-Campus). „Quilt CMS“ sustav trenutno se koristi unutar nekoliko obrazovnih institucija, a to su: Fakultet elektrotehnike i računarstva u Zagrebu, Ekonomski Fakultet Zagreb (koji sustav koristi u Intranet¹ obliku), Sveučilište u Zagrebu, Filozofski fakultet u Splitu, Vojni studiji, Studij inženjerstva, Fakultet ekonomije i turizma dr. Mijo Mirković u Puli i Studij Energetske učinkovitosti i obnovljivih izvora u Šibeniku. Sustav je web platforma koja sadrži veći broj web aplikacija koje služe za potporu nastavi pa se stoga radi o cjelovitom rješenju.

¹ Privatna računalna mreža unutar organizacije koja omogućava svojim zaposlenicima sigurnu razmjenu informacija ili obavljanje potrebnih radnji vezanih za organizaciju

Neke od tih web aplikacija jesu:

- Informacijski paket
- Red predavanja
- Rezervacija dvorana
- Moodle
- Knjižnica
- Satnica
- Završni radovi
- Materijalne potrebe
- Honoriranje studenata za pomoć u nastavi
- Projekti
- Nastavne aktivnosti
- Inventura
- Upravljanje radnim nalogima
- Evidencija nastave i opterećenja nastavnika.

Posebno zanimljiva je web aplikacija za evidenciju nastave i opterećenja nastavnika. Ona omogućava evidentiranje održane nastave ovisno o tipu (predavanje, seminar ili vježbe), unos termina, grupe studenata (i njihov broj), nastavnika, vrijeme trajanja i obrađenu temu. Ukoliko je više nastavnika zaduženo za isti kolegij, ova aplikacija omogućuje da budu upućeni u ono što će predavati ubuduće, a i kao kontrola je li sve što je predviđeno ujedno i obrađeno. Aplikacija također omogućava pregled svih nastavnih aktivnosti po predmetima i nastavnicima (Fakultet elektrotehnike i računarstva, 2016).

Na slici 3.3 prikazan je zapis u evidenciji nastave, gdje se za određeni predmet nastave može vidjeti koliko je sati održano po tipu nastave. Na konkretnom primjeru vidimo da je za predmet „Osnove informatike“ održano 9 sati predavanja što je preračunato u norma sate 18 sati koliko je do tog trenutka održano.

Slika 3.4 prikazuje obrazac za unos održane nastave u evidenciju, a potrebno je izabrati tjedan u kojem se nastava održala, grupu, točan dan i vrijeme održavanja, trajanje nastave, broj prisutnih studenata, predavača te održano gradivo.

3.3.3 EduPlan/EX

Eduplan/EX je još jedno cjelovito rješenje za planiranje nastave, razvijeno od strane tvrtke Lama d.o.o. korištenjem Microsoftovih tehnologija. Sustav omogućava planiranje nastave, evidenciju prisustva nastavi, evidentiranje održane nastave i izradu rasporeda sati. Sustav je namijenjen svakoj obrazovnoj ustanovi kojoj je potreban jednostavan način za upravljanje nastavom i resursima, te kreiranje raznih izvještaja poput onih o održanoj nastavi i nastavnom osoblju ili izvještaja koje su dužni dostaviti Ministarstvu obrazovanja. Posebna pogodnost je mogućnost integracije s RFID tehnologijom² pogotovo ukoliko se želi implementirati evidentiranje posjećenosti nastave (Lama Informacijske Tehnologije, 2016).

Glavne značajke ovog sustava vezane za evidenciju nastave jesu:

- Automatsko generiranje ugovora za vanjske suradnike u skladu s nastavnim planom
- Povezanost sustava s kadrovskim sustavom
- Povezanost sustava s ISVU-om
- Centralizirano praćenje svih događaja
- Pregled angažmana nastavnog osoblja na više fakulteta
- Praćenje izvršenja nastave (planska i neplanska)
- Evidencija posjećenosti predavanja
- Automatsko generiranje izvještaja po planiranoj i/ili izvršenoj nastavi
- Automatsko generiranje zakonskih izvještaja prema Ministarstvu obrazovanja
- Centralna evidencija i baza podataka.

3.3.4 Analiza i usporedba postojećih aplikacija s razvijenom

Iz pregleda postojećih aplikacija uočljivo je da uglavnom dijele iste značajke. Sve su web aplikacije i većinom su cjelovita rješenja za planiranje nastave. Kada se uzme u obzir segment evidencije održane nastave, sve aplikacije sadržavaju mogućnost upisa ukupne norme za određenog korisnika, unos zapisa o održanoj nastavi u evidenciju i generiranje različitih izvještaja i dokumenata (EduPlan/EX automatski generira ugovore za vanjske suradnike). Još jedna zajednička prednost im je povezanost s rasporedom sati što olakšava proces unosa zapisa u evidenciju.

² RFID je tehnologija koja koristi radio frekvenciju za razmjenu informacija između prijenosnih uređaja i računala

Razvijena aplikacija sadrži sve funkcije vezane za evidenciju nastave kao i postojeća rješenja, osim što je moguće generirati samo izvješće o održanoj nastavi. Za razliku od postojećih, razvijena aplikacija ima mogućnost praćenja prekovremenih sati.

4. Okruženje i tehnologije za razvoj aplikacije

Kako bi započeo razvoj aplikacije potrebno je odrediti u kojem okruženju se ona treba izvršavati, hoće li se pristupati s jednog računala na kojem se vodi cijela evidencija ili s bilo kojeg računala i lokacije u svako doba. U slučaju aplikacije za evidenciju održane nastave nužna je dostupnost svakome s odobrenjem stoga je logičan izbor izrada web aplikacije, odnosno dinamičke web stranice, zato što pristup mora biti omogućen većem broju korisnika.

4.1 Tipovi web stranica

Postoje dva tipa web stranica, statičke i dinamičke, a razlikuju se po načinu generiranja web stranice.

4.1.1 Statičke web stranice

Statičke web stranice (Košak i sur., 2015) su stranice koje se odvijaju u statičkom okruženju i realizirane su osnovnim web tehnologijama HyperText Markup Language-om (HTML) i Cascading Style Sheets-om (CSS). HTML se koristi za dodavanje sadržaja dok CSS prvenstveno služi kao alat za vizualno oblikovanje. Takav tip web stranice karakterizira statičan HTML sadržaj spremljen na poslužitelju, koji osluškuje vanjske upite i dostavlja traženi sadržaj. Sama izmjena sadržaja nije moguća preko web preglednika, neovisno o korisnikovim zahtjevima prilikom posjete, već se izmjena sadržaja i ažuriranje vrši tako što se ažurirane datoteke ponovno postavljaju na poslužitelja.

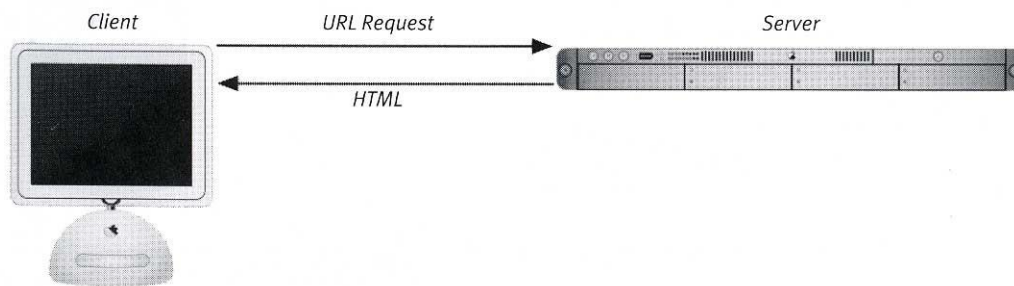
Prednosti korištenja ovog tipa web stranice jesu:

- potpuna kontrola nad sadržajem
- brzina odziva, jer nema potrebe za komunikacijom s bazom podataka
- jednostavniji hosting na webu.

Nedostaci su:

- povećanje kompleksnosti s povećanjem količine sadržaja
- potrebna je stručna osoba za izmjenu sadržaja
- kod promjene detalja koji se nalaze u svakoj datoteci potrebno je izmijeniti svaku posebno te ponovno postaviti na poslužitelja.

Na slici 4.1 prikazan je način komunikacije klijenta i servera kod statičkih web stranica.



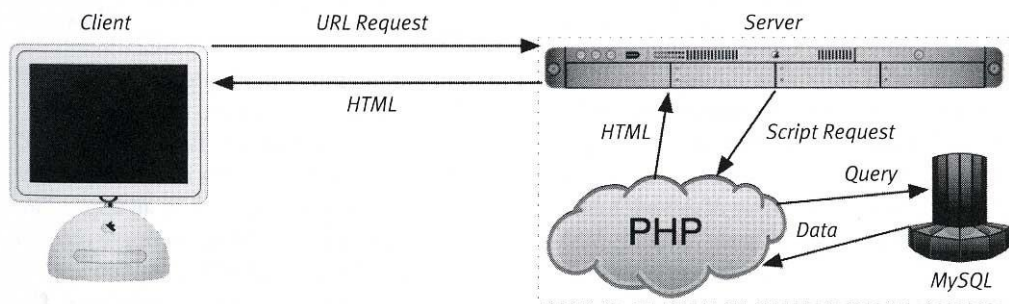
Slika 4.1 Komunikacija između klijenta i servera kod statičkih web stranica
(Ullman, 2012)

4.1.2 Dinamičke web stranice

Glavna karakteristika dinamičkih web stranica jest da one nisu eksplicitno napisane u HTML jeziku i spremljene na poslužitelju, već se dinamički generiraju prilikom pozivanja uz pomoć server-side programskih jezika. Glavne komponente koje su potrebne za ostvarivanje komunikacije s korisnikom su web preglednik, baza podataka i web poslužitelj.

Web poslužitelj je podijeljen na dva sloja, aplikacijski i podatkovni (baza podataka), dok korisnikov web preglednik predstavlja prezentacijski sloj. Aplikacijski sloj sastoji se od server-side programskog jezika koji je povezan s bazom podataka iz koje čita potrebne informacije. Nakon što se kod u aplikacijskom sloju izvrši i završi komunikaciju s bazom podataka, poslužitelj isporučuje podatke web pregledniku odnosno prezentacijskom sloju, i na taj način omogućuje konačnu prezentaciju zatražene stranice.

Osnovna prednost ovakvog tipa web stranice je mogućnost dinamičkog upravljanja web sadržajem, što ujedno znači i jednostavniju izmjenu sadržaja. Nedostatak pronalazimo u činjenici da je hosting skuplji kako se radi o kompleksnijim sustavima (Ullman, 2012). Na slici 4.2 prikazan je način komunikacije klijenta i servera kod dinamičkih web stranica.



Slika 4.2 Komunikacija između klijenta i servera kod dinamičkih web stranica
(Ullman, 2012)

4.2 Web aplikacije

Web aplikacije su programska (softverska) rješenja kojima se pristupa putem web preglednika koristeći internet. Njihova glavna prednost je mogućnost pristupa s bilo koje lokacije u svakom trenutku i s bilo kojeg uređaja (npr. računalo ili mobilni uređaj) koji ima neku vrstu web preglednika. Još jedna vrlo važna prednost je ta što se takve aplikacije ne trebaju prilagođavati različitim platformama već samo standardima postojećih web preglednika i međunarodnih organizacija.

Za rad web aplikacije potrebno je osigurati neke temeljne komponente, a to su:

- Hardver (računalo/poslužitelj) koji podržava mrežnu komunikaciju
- Operacijski sustav na hardveru
- Web server (koji se nalazi na hardveru odnosno operacijskom sustavu)
- Sustav za upravljanje bazom podataka
- Programski ili skriptni jezik.

Kod razvoja web aplikacija, kao bitne komponente potrebno je izdvojiti sustav za upravljanje bazom podataka, te programski ili programski skriptni jezik (Radovan, 2009).

4.2.1 Sustavi za upravljanje bazom podataka

Sustavi za upravljanje bazom podataka su računalni programi uz pomoć kojih se izgrađuje i upravlja bazom podataka kao i podacima koji se nalazi u njoj. Baza podataka sadrži organizirani skup strukturiranih podataka koji su potrebni za rad web aplikacije.

Postoji više različitih tipova sustava za upravljanje bazom podataka među kojima su

mrežni, hijerarhijski, logički i relacijski (sustav za upravljanje relacijskom bazom podataka³) koji je trenutno najzastupljeniji. U relacijskoj bazi podataka podaci su organizirani u tablicama koje se kreiraju, uređuju i spajaju jedna s drugom po određenim načelima i pravilima. Najpoznatije inačice relacijskih sustava su MySQL, PostgreSQL, Oracle, Microsoft SQL Server i mnogi drugi (Radovan, 2009).

4.2.2 Programski i skriptni jezici za razvoj web aplikacija

Realizacija web aplikacija moguća je upotrebom programskih ili programskih skriptnih jezika. Programski jezici predstavljaju skup simbola i pravila koje računalo razumije i pretvara u strojni jezik izvršavanjem koda. Najpoznatiji primjeri programskih jezika jesu: C, C++, C# i Java. Iz programskih jezika razvili su se skriptni jezici koji se ne razlikuju bitno od standardnih programskih jezika osim po načinu korištenja i izvođenja programa/skripti. Programi su najčešće samostalni elementi većeg sustava, a skripte se obično ugrađuju u druge softverske elemente. Najpoznatiji primjeri skriptnih jezika su PHP, Perl, Python, Ruby i Javascript.

4.3 Korištene tehnologije za razvoj aplikacije

Nakon odabira okruženja nužno je utvrditi koje tehnologije će se koristiti za razvoj aplikacije. Postoji niz kriterija koje treba uzeti u obzir kod odabira tehnologije poput:

- predznajna programera u određenim tehnologijama
- poznavanje okruženja za koje se aplikacija razvija
- dobro poznavanje zahtjeva
- potrebna količina interakcije sa sustavom
- broj korisnika.

U slučaju aplikacije za evidenciju održane nastave ona mora podržavati:

- rad s udaljenih lokacija
- rad većeg broja korisnika
- moraju postojati mehanizmi za laganu izmjenu sadržaja
- aplikacija mora biti uvijek dostupna zbog učestalosti korištenja.

Uzevši sve kriterije u obzir izabrane su sljedeće tehnologije:

- 1) HTML
- 2) CSS

³ Engl. RDBMS - Relational Database Management System

- 3) PHP
- 4) MySQL
- 5) JavaScript
- 6) jQuery.

4.3.1 Hypertext Markup Language

Hypertext Markup Language (skraćeno HTML) je jezik za označavanje i povezivanje dokumenata hiper-poveznicama. Označavanje dokumenata vrši se korištenjem tagova kojima se stvaraju, povezuju i strukturiraju elementi HTML dokumenta. Korišteni tagovi upućuju web preglednik na koji način će prikazivati tekst. Kreiranje HTML dokumenata moguće je korištenjem bilo kojeg tekst editora, a ekstenzija za HTML datoteku može biti .htm ili .html (W3Schools, 2016).

Nastao je 1993. godine, razvili su ga W3C konzorcij⁴ i grupa WHATWG⁵, a u trenutnoj upotrebi je HTML verzija 5 (HTML 5).

Svaka HTML datoteka započinje tagom `<html>`, a završava s `</html>`. Postoji i niz drugih tagova koji označavaju elemente jedne takve datoteke, a neki od korištenijih jesu:

- `<head></head>` - element sadržan u svakom HTML dokumentu i služi za spremanje informacija o zaglavlju dokumenta, a može uključivati naslov dokumenta, skripte, stilove, meta podatke i drugo
- `<body></body>` - element sadržan u svakom dokumentu, u njemu se upisuje sadržaj dokumenta
- `<br />` - koristi se za prelazak u novi red, jer se HTML-u treba eksplicitno narediti gdje je prekid linije
- `<p></p>` - oznaka za odlomak, služi za pisanje teksta u dokumentu, a sadrži prelazak u novi red nakon zaključnog taga
- `<h1></h1>` - tag za naslov, za odabir jedne od 6 razina naslova, upisuje se brojka uz slovo h
- `<ol></ol>` - tagovi za kreiranje listi, moguće je izabrati tip liste
- `<a></a>` - koristi se za linkove.

Unutar svakog taga moguće je zapisati attribute koji će tagovima omogućiti raznolikost

⁴ Engl. World Wide Web Consortium

⁵ Engl. Web Hypertext Application Technology Working Group

opisivanja sadržaja elemenata. Atributi se zapisuju u obliku `ime_atributa="vrijednost"`, a neki atributi su *align* za poravnanje, *width* za širinu, *height* za visinu, *href* za dodavanje destinacije linku, *src* adresa na kojoj se nalazi slika i mnogi drugi.

4.3.2 Cascading Style Sheets

Cascading Style Sheets (skraćeno CSS) je jednostavan mehanizam za dodavanje stilova HTML datotekama. Elementima datoteke moguće je dodati stilove fontova, boja, visine, širine, boju pozadine, poravnanje i druge. CSS je nastao 1996. godine, a kreiran je od strane Hakon Wium Lie-a, Bert-a Bos-a i W3C konzorcija. Dolaskom u opticaj zamijenio je dotadašnji način kreiranja web stranica putem tabličnih layout-a (koristile su se tablice za formiranje stranice) i uveo novi način pisanja HTML dokumenata, pa postalo moguće odvojiti prezentaciju podataka i dizajn. Stoga je HTML kod postao manji i čitljiviji, a velika novost je bila i mogućnost upotrebom jednostavnih parametara promijeniti izgled web stranice. CSS je donio širok spektar načina za uređivanje podataka s mogućnošću kontroliranja bilo kojeg segmenta web stranice, koji nisu bili mogući s HTML-om.

Samo neke od mogućnosti CSS-a su podešavanje pozadine, postavljanje okvira na elemente, uređivanje listi, postavljanje veličine elemenata, uređivanje tablica i teksta. Moguće je dodavati stilska pravila samim elementima, a ta pravila će vrijediti za svaki element s tim tagom, ili deklarirati stilske klase koje će se naknadno moći koristiti u HTML kodu. Korištenje CSS klase u HTML kodu vrši se pisanjem imena klase unutar atributa `class` (`class="ime_atributa"`) koji je sadržan unutar taga nekog elementa (W3Schools, 2016). Primjer CSS klase može se vidjeti na primjeru koda 4.1.

```
.tekst {  
font-family: Arial;  
font-size: 12px;  
color: red;  
background-color: blue;  
}
```

Kod 4.1 Primjer CSS klase

4.3.3 PHP: Hypertext Preprocessor

PHP: Hypertext Preprocessor (skraćeno PHP) je open-source server-side skriptni programski jezik namijenjen programiranju prvenstveno dinamičkih web stranica. Nastao je 1995., kada je bio skup perl skripti, razvio ga je Rasmus Lerdorf za vlastite potrebe, a uloga mu je bila brojanje posjeta na njegovoj web stranici. S vremenom se pojavila potreba za upotrebom većeg broja funkcija, stoga je razvio novu verziju u programskom jeziku C koja je omogućavala rad s bazama podataka te na taj način omogućila programiranje jednostavnih dinamičkih web aplikacija. Rasmus se odlučio na objavljivanje PHP-a kao open-source programskog jezika. PHP je postao jedan od najkorištenijih programskih jezika za programiranje dinamičkih web aplikacija i ostao do dan danas. Svoju popularnost postigao je zbog sličnosti C-u, a do sada je izašao niz sljedećih inačica:

- 1) PHP/FI (1995.)
- 2) PHP/FI 2.0 (1997.)
- 3) PHP3 (1997.)
- 4) PHP4 (2000.)
- 5) PHP5 (2006.)
- 6) PHP7 (2015.).

PHP je bogat funkcijama za manipuliranje različitim sadržajima kao što su rad grafikom (jpg⁶, png⁷ i drugim), .NET⁸ modulima i sa XML-om. Moguće ga je kombinirati s drugim skriptnim jezicima poput JavaScript-a. PHP je moguće koristiti na većini web poslužitelja, operativnih sustava (Windows, Unix, Linux i drugi), platformi i u kombinaciji s velikim brojem sustava za upravljanje relacijskim bazama podataka (među njima i MySQL). Svaki ozbiljniji hosting servis podržava korištenje PHP-a od strane korisnika.

PHP radi na sljedeći način: kad web server primi zahtjev od klijenta on mu dostavi određenu web stranicu koja se generira uz pomoć PHP skripti, a web server pokreće PHP interpreter koji zatim procesira dane datoteke koje sadrže PHP i HTML kod. Na mjestima gdje se nalazi PHP skripta generira se HTML kod i postavlja na isto mjesto.

⁶ JPG (engl. JPEG - Joint Photographic Experts Group) je komprimirani grafički format za pohranu slika s gubicima

⁷ PNG (engl. Portable Network Graphics) je grafički format namijenjen pohrani ili streamingu slika bez gubitaka

⁸ Microsoftova tehnologija čija je osnova .NET Framework, tehnologija koja nudi gotova rješenja i funkcionalnosti kako bi olakšala i ubrzala posao programerima

Web poslužitelj potom sprema rezultat i prosljeđuje ga klijentu čiji web preglednik prikazuje sadržaj u obliku web stranice. Kod nekih PHP skripti tijekom izvođenja istih potrebna je upotreba MySQL tehnologije kako bi se podaci dobili iz relacijske baze podataka.

Pisanje PHP koda vrši se unutar „<?php“ i „?>“ oznaka, koje označavaju gdje kod počinje i završava. Jezik je vrlo intuitivan i sličan programskim jezicima poput C (kojem je najbliži po sintaksi), C++ ili Java-e. Varijable karakterizira znak \$ ispred imena, a nije im potrebno doznati tip već ih sam prepoznaje. PHP podržava sve standardne tipove podataka kao što su: float (decimalni brojevi), integer (cijeli brojevi), string (niz znakova), array (polje) i objekte. Također podržava aritmetičke operatore (+, -, *, /, %, =), operatore uspoređivanja (<, >, ==, <=, >=), operatore pridruživanja, upotrebu standardnih petlji (for, while, do..while) i upotrebu uvjetnih grananja (if, if..else). Komentare je moguće označiti s tri različita tipa sintakse, za više redova /* */ , // i # za komentar u jednom redu. PHP podržava rad s funkcijama kao i objektno orijentirano programiranje (OOP)⁹ (php.com.hr, 2008). Jednostavna PHP skripta može se vidjeti na primjeru koda 4.2, dok kod 4.3 prikazuje funkciju zbrajanja u PHP-u.

```
<html>
  <head>
    <title>Primjer</title>
  </head>
  <body>
    <?php echo '<p>Hello World</p>'; ?>
  </body>
</html>
```

Kod 4.2 Jednostavna PHP skripta unutar HTML strukture

```
<?php
function sum($x, $y) {
    $z = $x + $y;
    Return $z;
}
echo '5 + 10 = ' . sum (5, 10) . '<br>';
echo '7 + 13 = ' . sum (7, 13) . '<br>';
echo '2 + 4 = ' . sum (2, 4) . '<br>'; ?>
```

Kod 4.3 Funkcija zbrajanja u PHP-u

⁹ Pristup programiranju u kojem se projektiranje aplikacije promatra kao skup objekata koji izmjenjuju poruke između sebe. Osnovne vrijednosti su: objekti, enkapsulacija, apstrakcija, nasljeđivanje i polimorfizam

4.3.4 MySQL

MySQL je jedan od predstavnika sustava upravljanja relacijskim bazama podataka. Nastao je 1995. godine, a razvila ga je švedska kompanija MySQL AB čiji su vlasnici Allan Larsson i Michael Widenius. Trenutno je u vlasništvu Oracle-a. Prvotno je bio namijenjen za osobne potrebe tvrtke jer s tadašnjim sustavima upravljanja nisu bili zadovoljni. Napisan je u C-u i C++-u te je potpuno besplatan za korištenje (open-source softver). Njegova popularnost može se vidjeti u podacima o tvrtkama koje ga koriste, a to su među ostalim Facebook, Twitter, Flickr i Youtube.

MySQL je u mogućnosti istodobno posluživati veći broj korisnika koji se služe istom bazom podataka. Njegova jezgra je SQL¹⁰ jezik uz pomoć kojeg korisnik komunicira s bazom podataka. Ono što odlikuje ovaj tip relacijske baze je brzina (sustav nadmašuje konkurentne sustave u brzini izvođenja operacija), dostupnost (moguće ga je nabaviti besplatno), otvorenost sustava (moguće je mijenjati izvorni kod kako bi ga prilagodili svojim potrebama), mogućnost rada na većini operativnih sustava i kompatibilnost s programskim jezicima kao što su PHP, PERL, C, C++, Java i mnogim drugim.

Osnovni pojmovi koji su bitni prilikom korištenja MySQL-a jesu:

- Baza podataka – skup tablica s podacima
- Tablica – predstavlja matricu s podacima
- Stupac – sadrži jednu vrstu podataka, npr. imena ljudi
- Redak – povezani skup podataka, npr. podatke o jednoj osobi
- Primarni ključ – označava vrijednost koja je predstavlja redak, svaki takav ključ može se pojaviti u samo jednom retku
- Strani ključ – predstavlja vezu između dviju tablica
- Kompozitni ili složeni ključ – ključ koji je sastavljen od više stupaca, koristi se kada nije dovoljan samo jedan stupac za predstaviti redak

Početak rada u MySQL-u započinje kreiranjem baze podataka, potom se novo kreirana baza puni tablicama prilikom čega se definira struktura svake pojedine tablice kao i međuovisnosti s drugim tablicama. Tablice se popunjavaju s relevantnim podacima.

SQL naredbe za rad u sustavu koriste logične izraze poput CREATE (DATABASE ili TABLE) za kreiranje sadržaja, INSERT za ispunjavanje tablica s podacima, UPDATE za izmjenu podataka u tablicama te DELETE za brisanje podataka. Vrlo važan segment u MySQL sustavu su SELECT naredbe (upiti) kojima se dohvaćaju podaci iz

¹⁰ Engl. Structured Query Language

baze podataka prema uvjetima (sintaksa WHERE) koje mi postavimo.

Kao što je već navedeno MySQL je kompatibilan s PHP programskim jezikom, a dokaz tome je i biblioteka funkcija napisana u PHP-u koja uvelike olakšava rad s MySQL-om. Neki od primjera funkcija su `mysqli_connect()` (služi za konekciju na bazu), `mysqli_query()` (služi za izvršavanje upita na bazu), `mysqli_fetch_array()` (pretvara rezultat upita u polje), `mysqli_num_rows()` (vraća broj redaka dobiven upitom) i mnogi drugi (tutorialspoint, 2016).

Kodom 4.4 prikazan je primjer MySQL naredbe za kreiranje tablice (`tip_predavanja`), a kod 4.5 je primjer upotrebe PHP funkcija uz pomoć koji se ostvaruje interakcija s MySQL-om (prikazan je kod za kreiranje baze podataka).

```
CREATE TABLE tip_predavanja
(
  id int NOT NULL AUTO_INCREMENT,
  naziv varchar(255) NOT NULL,
  vrijednost float NOT NULL,

  PRIMARY KEY (id),
  UNIQUE (naziv)
);
```

Kod 4.4 Primjer MySQL naredbe za kreiranje tablice

```
$sql = "CREATE DATABASE mojaBP";
If(mysqli_query($conn, $sql)) {
    Echo "Database created successfully";
} else {
    Echo "Error creating database: " . mysqli_error($conn);
}
```

Kod 4.5 Primjer upotrebe PHP funkcija za interakciju s MySQL-om

4.3.5 JavaScript

JavaScript je objektno orijentirani skriptni jezik, vrlo sličan PHP-u, poznat kao jezik koji se koristi za razvoj web aplikacija. Može biti i client-side i server-side orijentiran, međutim češće se koristi kao client-side. JavaScript se koristi za dizajniranje ili programiranje ponašanja web stranica ukoliko dođe do nekog događaja što je i glavna razlika između njega i ostalih skriptnih jezika. JavaScript je prototipno baziran programski jezik koji ne podliježe uobičajenoj definiciji objektno orijentiranih jezika. On ne podržava koncept klase dok nasljeđivanje realizira kloniranjem postojećih objekata, tj. stvaranjem prototipova (Mozilla Developer Network, 2015).

Funkcije u JavaScriptu su funkcije prvog reda (engl. First-class functions), što znači da mogu biti argumenti drugih funkcija, pridruživati se varijablama ili pojavljivati kao vrijednosti drugih funkcija. Uključivanje JavaScript koda u html vrši se `<script>` html tagom. Deklaracija varijabli se vrši pisanjem ključne riječi „var“ prije imena varijable i prilikom deklaracije nije nužno napisati tip varijable.

JavaScript poput PHP-a podržava aritmetičke operatore, operatore uspoređivanja, operatore pridruživanja, logičke operatore i konkatenciju varijabli tipa string upotrebom operatora za zbrajanje. Komentari u više redova zapisuju se unutar `/* i */`, a u jednom redu nakon `//`. Za ispis sadržaja koristi se naredba `document.write()`. Postoje i pred memorirani objekti koji se mogu koristiti, a to su string objekt (manipuliranje tekстом), date objekt (rad s datumom i vremenom), array objekt (rad s nizovima), boolean objekt (pretvaranje u logičke vrijednosti), math objekt (rad s matematičkim izrazima) i RegExp objekt (rad s regularnim izrazima ¹¹). JavaScript podržava rad s funkcijama, petlje (for, while, do..while), uvjetna grananja (if, if..else, switch). Kao što je spomenuto ranije u tekstu JavaScript može prepoznati određene događaje odnosno aktivnosti, što je i glavna značajka jezika. Primjeri toga su klik mišem (onClick), pristup web stranici (onLoad), prelazak mišem preko elementa stranice (onMouseOver) ili u slučaju kad je potrebna validacija forme klikom na submit form (onSubmit) (W3Schools, 2016).

¹¹ Regularni izraz (engl. Regular expression) je objekt koji opisuje „uzorak znakova“

4.3.6 jQuery

jQuery je JavaScript biblioteka osmišljena kako bi se olakšalo client-side (klijentsko) skriptiranje html-a. jQuery je najpoznatija i najkorištenija biblioteka takvog tipa, a u prilog tome ide podatak da više od pola najprometnijih web stranica koristi baš njega. jQuery je razvijen 2006. godine, a njegov autor je John Resig. Slobodan i besplatan je za upotrebu jer se nalazi pod open-source licencom (jQuery user interface, 2016).

Sintaksa jQuery-a je dizajnirana tako da omogući jednostavniju navigaciju po dokumentu, odabir DOM¹² elemenata, kreiranje animacija, bolje rukovanje događajima i razvoj AJAX aplikacija. Korištenje ove biblioteke moguće je na svim popularnim web preglednicima (Firefox, Chrome, Safari, Opera).

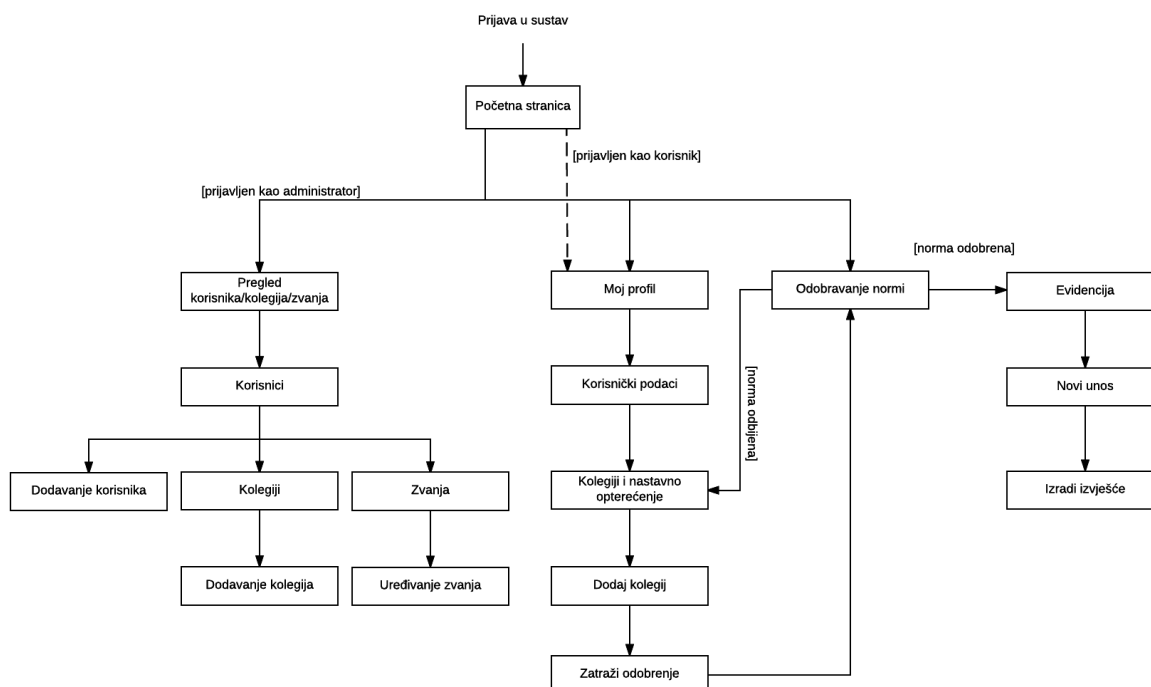
Prednosti korištenja jQuerya su:

- Odvajanje JavaScripta od HTML-a
- Kratak i jasan, omogućava ulančavanje funkcija i skraćivanje imena funkcija
- Uklanja moguće nekompatibilnosti među web preglednicima (prikaz u jednom se može razlikovati od prikaza u drugom pregledniku)
- Proširiv je – novi događaji, elementi i metode jednostavno se mogu nadodati i ponovno koristiti kao plugin.

¹² Engl. Document Object Model – struktura u obliku stabla koja predstavlja sve elemente web stranice

5. Hodogram korištenja aplikacije

Hodogram korištenja (slika 5.1) je prikaz puteva korištenja aplikacije. Početak rada s aplikacijom počinje prijavom u sustav. Korisnicima je dodijeljena jedna od dvije uloge, administrator ili korisnik, a ovisno o ulozi mijenja se mogući odabir. Razlika među njima je u tome da administrator ima omogućen pregled korisnika, kolegija i zvanja kao i njihovo dodavanje ili izmjenu, te mogućnost odobravanja normi. Ono što im je zajedničko je put do „Izrade izvješća“. Put započinje ulaskom u „Moj profil“ odnosno „Korisničke podatke“ gdje je potrebno izabrati zvanje kako bi se otvorila opcija „Kolegiji i nastavno opterećenje“. „Kolegiji i nastavno opterećenje“ nude opciju dodavanja kolegija, a nakon što su svi kolegiji dodani potrebno je zatražiti odobrenje norme. Administrator može i ne mora odobriti normu, ukoliko ne odobri korisnik/administrator se vraća na „Kolegiji i nastavno opterećenje“ gdje mora izvršiti izmjene i ponovno zatražiti odobrenje. Kada je norma odobrena korisnik/administrator dobiva pristup evidenciji gdje može vršiti nove unose u evidenciju. Nakon što je unio sve zapise u evidenciju ima opciju izrade izvješća na temelju evidencije.

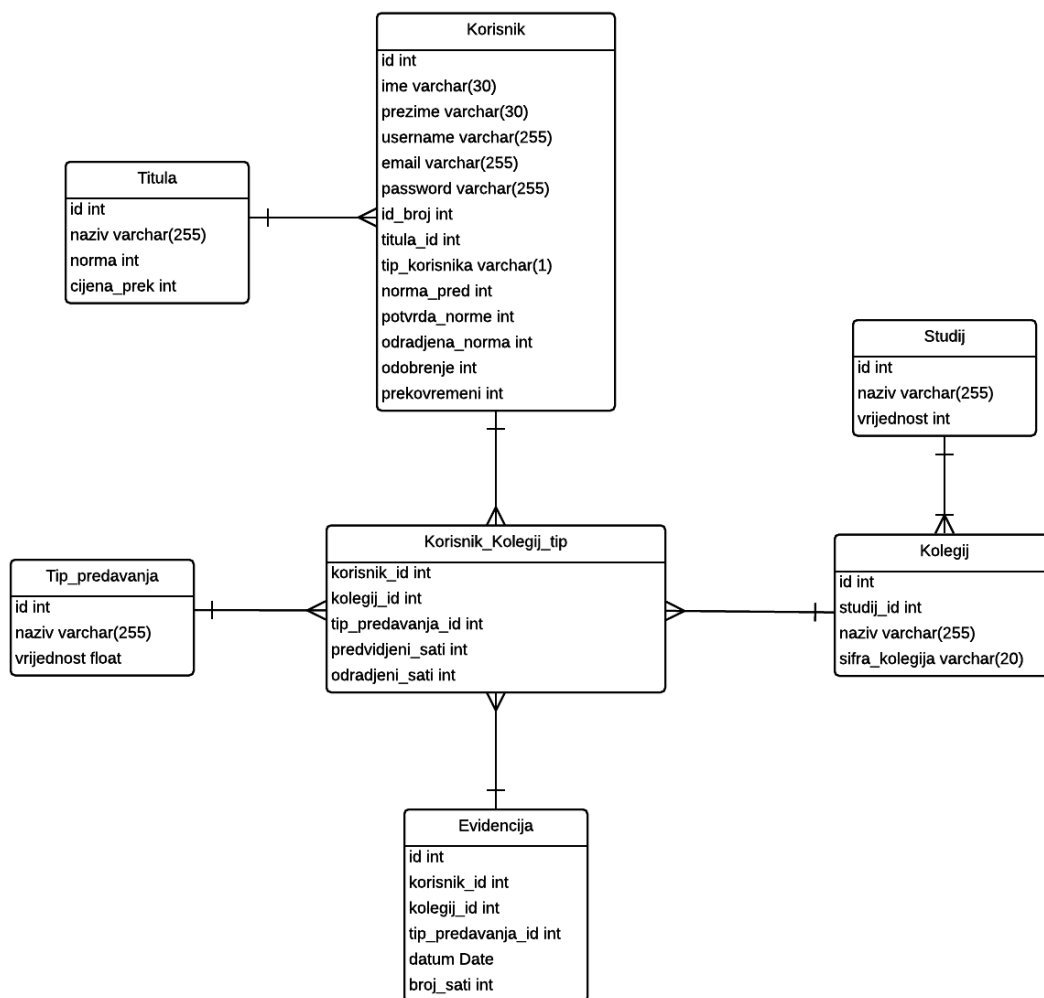


Slika 5.1 Hodogram korištenja aplikacije

6. Programsko rješenje

6.1 Baza podataka

Kako bi se omogućio normalan rad aplikacije potrebno je imati bazu podataka iz koje će aplikacija dohvaćati i spremati podatke. Na slici 6.1 vidi se dijagram entiteti-veze (eng. Entity-Relationship Diagram, ERD) koji prikazuje model baze podataka korištene u aplikaciji. Baza podataka sastavljena je od sedam tablica koje su međusobno povezane. Svaka tablica, osim tablice Korisnik_kolegij_tip, posjeduje atribut id koji je primarni ključ. Prilikom stvaranja novog retka baza automatski generira jedinstveni id (u toj tablici) za taj redak preko kojeg ga je kasnije lakše identificirati. Detaljan opis tablica slijedi u nastavku teksta.



Slika 6.1 ERD model

6.1.1 Tablica Titula

Tablica Titula sadrži informacije o zvanjima nastavnika. Sastavljena je od četiri atributa: id, naziv, norma i cijena_prek. Tipovi atributa su: id integer, naziv varchar (do 255 znakova), norma integer, cijena_prek integer.

Naziv je ujedno i jedinstveni ključ¹³, a sadrži naziv zvanja nastavnika. Norma označava vrijednost nastavnog opterećenja za određeno zvanje. Cijena_prek sadržava informaciju o cijeni prekovremenog sata tog zvanja.

6.1.2 Tablica Korisnik

Tablica Korisnik sadrži informacije o korisniku aplikacije. Sastavljena je od petnaest atributa: id, ime, prezime, username, email, password, id_broj, titula_id, tip_korisnika, norma_predvi, potvrda_norme, odradjena_norma, odobrenje, prekovremeni i vanjski. Tipovi atributa su: id integer, ime varchar (do 30 znakova), prezime varchar (do 30 znakova), username varchar (do 255 znakova), email varchar (do 255 znakova), password varchar (do 255 znakova), id_broj integer, titula_id integer, tip_korisnika varchar (jedan znak), norma_predvi integer, potvrda_norme integer, odradjena norma integer, odobrenje integer, prekovremeni integer i vanjski integer.

Atributi ime i prezime označavaju ime i prezime korisnika. Username (jedinstveni ključ) je korisničko ime korisnika, email je adresa e-pošte korisnika, a password označava korisničku lozinku. Id_broj je identifikacijski broj korisnika kako bi se moglo raspoznati korisnike s istim imenom. Titula_id je vanjski ključ¹⁴ tablice Titula označava zvanje korisnika. Tip_korisnika predstavlja zapis o kojem tipu korisnika je riječ (administrator ili korisnik). Norma_predvi sadrži predviđen broj norma sati, dok odradjena_norma sadrži broj odrađenih norma sati. Potvrda_norme sadrži informaciju je li odobrena norma od strane administratora, dok odobrenje sadrži je li zatraženo odobrenje norme od strane korisnika. Prekovremeni sadrži zbroj prekovremenih sati korisnika, a vanjski može sadržavati brojeve 0 ili 1 ovisno o tome radi li se o vanjskom korisniku ili ne.

6.1.3 Tablica Tip_predavanja

Tablica Tip_predavanja sadrži informacije o tipovima nastave. Sastavljena je od tri atributa: id, naziv i vrijednost. Tipovi atributa su: id integer, naziv varchar (do 255

¹³ Osobina jedinstvenog ključa je da u istoj tablici za taj atribut ne smije postojati ista vrijednost u više redaka.

¹⁴ Vanjski ključ je ključ naslijeđen od tablice roditelja i služi kao veza s njom.

znakova) i vrijednost float.

Naziv je jedinstveni ključ i sadrži naziv tipa nastave, atribut vrijednost sadrži informaciju o vrijednosti tipa nastave u norma satima.

6.1.4 Tablica Studij

Tablica Tip_predavanja sadrži informacije o tipovima nastave. Sastavljena je od tri atributa: id, naziv i vrijednost. Tipovi atributa su: id integer, naziv varchar (do 255 znakova) i vrijednost float.

Naziv je jedinstveni ključ i sadrži naziv studija, atribut vrijednost sadrži informaciju o vrijednosti predavanja po studiju u norma satima.

6.1.5 Tablica Kolegij

Tablica Kolegij sadrži informacije o kolegijima. Sastavljena je od četiri atributa: id, studij_id, naziv i sifra_kolegija. Tipovi atributa su: id integer, studij_id integer, naziv varchar (do 255 znakova) i sifra_kolegija varchar (do 20 znakova).

Studij_id je vanjski ključ tablice Studij i označava studij na kojem se izvodi kolegij. Naziv se odnosi se na naziv kolegija, a sifra_kolegija (jedinstveni ključ) je šifra kolegija koja je jedinstvena iz razloga što naziv kolegija može biti ista stoga mora postojati način raspoznavanja.

6.1.6 Tablica Korisnik_kolegij_tip

Tablica Korisnik_kolegij_tip je specifična tablica po tome što ima tri vanjska ključa koji čine primarni (kompozitni) ključ. Sastavljena je od pet atributa: korisnik_id, kolegij_id, tip_predavanja_id, predvidjeni_sati i odradjeni_sati. Tipovi atributa su: korisnik_id integer, kolegij_id integer, tip_predavanja_id integer, predvidjeni_sati integer i odradjeni_sati integer.

Korisnik_id, kolegij_id i tip_predavanja_id su vanjski ključevi tablica Korisnik (informacije o korisniku), Kolegij (informacije o kolegiju) i Tip_predavanja (informacije o tipu nastave). Ujedno tvore kompozitni primarni ključ što znači da je njihova kombinacija jedinstvena unutar tablice. Predvidjeni_sati sadrži broj predviđenih sati nastave, dok odradjeni_sati sadrži podatak o odrađenim satima nastave.

6.1.7 Tablica Evidencija

Tablica Evidencija sadrži zapise u evidenciju. Sastavljena je od šest atributa: id, korisnik_id, kolegij_id, tip_predavanja_id, datum i broj_sati. Tipovi atributa su: id integer, korisnik_id integer, kolegij_id integer, tip_predavanja_id integer, datum Date i broj_sati integer.

Atributi korisnik_id, kolegij_id i tip_predavanja_id su vanjski ključ tablice Kolegij_korisnik_tip, njihov zapis je jedinstven za određeni datum. Datum sadrži datum unosa zapisa u evidenciju, dok broj_sati sadrži broj sati odrađene nastave.

6.2 Rješenja specifičnih problema

6.2.1 Dodjeljivanje identifikacijskog broja korisniku

Aplikacija podržava rad velikog broja korisnika, pa se može dogoditi da se pojave korisnici s istim imenom i prezimenom. Iz tog razloga potrebno je uvesti identifikacijski broj korisnika u svrhu njihovog razlikovanja. Kako bi se administratoru olakšao posao prilikom dodavanja novog korisnika, aplikacija sama generira peteroznamenasti identifikacijski broj uz pomoć funkcije rand().

Kod 6.1 prikazuje rješenje problema generiranja broja. U varijablu se sprema slučajan peteroznamenasti broj, potom se u bazi podataka provjerava postoji li isti identifikacijski broj, a ukoliko postoji dodjeljuje se novi sve dok se ne generira broj koji već nije evidentiran.

```
$id_broj = rand(10000, 99999);  
$idbroj_sql = "SELECT id_broj FROM korisnik WHERE id_broj = '$id_broj'";  
$idbroj_rez = mysqli_query($conn, $idbroj_sql);  
while (mysqli_num_rows($idbroj_rez) == 1) {  
    $id_broj = rand(10000, 99999);  
    $idbroj_rez = mysqli_query($conn, $idbroj_sql);  
}
```

Kod 6.1 Rješenje problema identifikacijskog broja

6.2.2 Generiranje lozinke

Aplikacija podržava rad velikog broja korisnika, pa se može dogoditi da se pojave korisnici s istim imenom i prezimenom. Iz tog razloga potrebno je uvesti identifikacijski broj korisnika u svrhu njihovog razlikovanja. Kako bi se administratoru olakšao posao prilikom dodavanja novog korisnika, aplikacija sama generira peteroznamenasti identifikacijski broj uz pomoć funkcije rand().

Kod 6.2 prikazuje rješenje problema generiranja broja. U varijablu se sprema slučajan peteroznamenasti broj, potom se u bazi podataka provjerava postoji li isti identifikacijski broj, a ukoliko postoji dodjeljuje se novi sve dok se ne generira broj koji već nije evidentiran.

```
$alphabet =  
'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890';  
$pass = array();  
$alphaLength = strlen($alphabet) - 1;  
for ($i = 0; $i < 8; $i++) {  
    $n = rand(0, $alphaLength);  
    $pass[] = $alphabet[$n];  
}  
$password = implode($pass);
```

Kod 6.2 Rješenje problema generiranja lozinke

6.2.3 Automatizirano slanje e-mail-a

Nakon što je novi korisnik dodan u sustav, potrebno ga je obavijestiti o njegovim korisničkim podacima. Rješenje tog problema je slanje automatiziranog e-mail-a koji sadrži korisničko ime i lozinku korisnika, izravno na e-mail adresu korisnika. To je postignuto korištenjem mail() funkcije koja je dio standardne PHP biblioteke.

Funkcija mail() u ovom slučaju ima četiri argumenta, \$to varijabla u koju je spremljena e-mail adresa primatelja, \$subject varijabla gdje je spremljen naslov e-mail-a, \$message varijabla koja sadrži HTML kod s porukom i \$headers varijabla koja sadrži dodatne informacije o pošiljatelju e-mail-a. Kod 6.3 prikazuje korištenje funkcije mail().

```
$to = '$email';  
$subject = 'Korisnicki podaci sustava za evidenciju nastave';  
$message = '  
<html>  
  <head>  
    <title>Korisnicki podaci</title>  
  </head>  
  <body>  
    <p>Postovani/a, kreiran je Vas korisnicki racun za pristup sustavu  
evidencije nastave.</p>  
    <a href="http://aplikacija.16mb.com">Sustav za evidenciju nastave</a>  
    <p>Korisnicko ime: $username</p>  
    <p>Lozinka: $password</p>  
  </body>  
</html>  
';  
$headers = 'MIME-Version: 1.0' . "\r\n";  
$headers .= 'Content-type: text/html; charset=UTF-8' . "\r\n";  
$headers .= 'From: sen@aplikacija.16mb.com' . "\r\n" .  
  'Reply-To: sen@aplikacija.16mb.com';  
mail($to, $subject, $message, $headers);
```

Kod 6.3 Automatizirano slanje e-mail-a

6.2.4 Prikaz listi

Prikaz listi može predstavljati veliki problem pregledavanja kad se radi o velikim listama koje je potrebno grupirati po nekom kriteriju. Primjer je toga lista kolegija koju je potrebno grupirati po vrsti studija (preddiplomski, diplomski i poslijediplomski), a jedan od mehanizama za to postići je korištenje jQuery biblioteke, konkretno accordion-a. Stvaraju se kartice koje je moguće zatvoriti ili otvoriti, po potrebi, unutar kojih se nalaze podaci (jQuery user interface, 2016).

Kod 6.4 prikazuje način uključivanja skripte u datoteku i postavljanje željenog ponašanja (poput mogućnosti otvaranja i zatvaranja ili visine). U nastavku je vidljivo kako se skripta koristi unutar HTML koda, dodjeljivanjem atributa elementu.

```
<script src="https://code.jquery.com/jquery-1.12.4.js"></script>
<script src="https://code.jquery.com/ui/1.12.0/jquery-ui.js"></script>
<script>
    $( function() {
        $( "#accordion" ).accordion({
            collapsible: true,
            heightStyle: "content",
            active: false
        });
    });
</script>

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

Kod 6.4 Korištenje accordion-a

6.2.5 Provjera popunjenosti polja u obrascima

Provjera popunjenosti obrazaca je bitan faktor za održavanje podataka u bazi podataka točnima. Kako bi izbjegli unos praznih ili nepravilno ispunjenih polja u obrascu potrebno je uvesti mehanizam koji to može provjeriti. Jedan od načina je provjera putem JavaScript RegExp objekta (regularni izrazi).

Unutar JavaScripta deklariraju se varijable u koje su spremljeni znakovi koje je dozvoljeno koristiti prilikom unosa, kod 6.5 prikazuje primjer s kolegijima gdje je unutar varijable ck_naziv definirano da se smiju koristiti slova (velika i mala), brojevi od 0 do 9 i posebna slova koja su specifična za hrvatsku abecedu. Potom se poziva funkcija validate() s argumentom koji je naziv obrasca, te se novim varijablama daju vrijednosti varijabla iz obrasca koje su preuzete s atributa name unutar <input> tagova. Vrš se provjera i ovisno o rezultatu provjere obavještava korisnika o nepravilnostima.

```

<script type="text/javascript">
var ck_naziv = /^[A-Za-z0-9 \u010D\u0107\u0161\u017E\u0111\u010C
                                     \u0106\u0160\u017D\u0110]{3,40}$/;
var ck_sifra = /^[A-Za-z0-9_ ]{2,5}$/;

function validate(form){
    var studij = form.studij.value;
    var naziv = form.naziv.value;
    var sifra = form.sifra.value;
    var errors = [];

    if (studij==0) {
        errors[errors.length] = "Izaberite vrstu studija.";
    }
    if (!ck_naziv.test(naziv)) {
        errors[errors.length] = "Unesite točan naziv kolegija.(max.
                                   40 slova ili brojeva)";
    }
    if (!ck_sifra.test(sifra)) {
        errors[errors.length] = "Unesite šifru kolegija.(max. 5
                                   slova ili brojeva)";
    }
    if (errors.length > 0) {
        reportErrors(errors);
        return false;
    }
    return true;
}
function reportErrors(errors){
    var msg = "Molimo Vas unesite ispravne podatke...\n";
    for (var i = 0; i<errors.length; i++) {
        var numError = i + 1;
        msg += "\n" + numError + ". " + errors[i];
    }
    alert(msg);
}
</script>

```

Kod 6.5 Regularni izrazi

Kod 6.6 prikazuje način na koji se poziva JavaScript, a to je putem atributa onSubmit unutar taga <form> i zadavanje imena (atribut name) formi preko kojeg je JavaScript prepoznao što treba provjeriti.

```

<form method="POST" action="pregled_kolegija.php" onSubmit="return
validate(this);" name="form">

```

Kod 6.6 Pozivanje JavaScripta putem događaja onSubmit

6.2.6 Dinamički kalendar

Bitan segment unosa zapisa u evidenciju je postavljanje točnog datuma kojeg je nastava održana i vođenje brige da je podatak unesen pravovremeno. Bitno je isto tako korisniku ponuditi suvremen način odabira datuma iz dinamičkog kalendara. Za realizaciju takvog kalendara moguće je isto koristiti jQuery biblioteku JavaScripta. Koristi se dodatak naziva datepicker (na hrvatskom birač datuma). Nudi širok spektar mogućnosti pa i onu koja je najbitnija a to je ograničavanje mogućih izbora datuma (jQuery user interface, 2016). Kod 6.7 prikazuje korištenje datepicker-a, i postavljanje parametara njegovog prikaza što uključuje format datuma, ograničenje u biranju datuma od tri tjedna unatrag do trenutnog datuma, imena mjeseci, skraćena imena mjeseci i dan od kojeg počinje tjedan (standardno počinje od nedjelje).

```
<script src="http://code.jquery.com/jquery-1.10.2.js"></script>
<script src="http://code.jquery.com/ui/1.10.4/jquery-ui.js"></script>

<script>
$(function() {
$( "#datepicker-1" ).datepicker({
dateFormat:"dd/mm/yy",
maxDate: "0",
minDate: "-3w",
monthNames: [ "Siječanj", "Veljača", "Ožujak", "Travanj", "Svibanj",
               "Lipanj", "Srpanj", "Kolovoz", "Rujan", "Listopad",
               "Studenj", "Prosinac" ],
dayNamesMin: [ "Ned", "Pon", "Uto", "Sri", "Čet", "Pet", "Sub" ],
firstDay: 1
});
});
</script>
```

Kod 6.7 Korištenje datepicker-a

6.2.7 Izračun prekovremenih sati

Svaki korisnik ima svoju predviđenu normu za akademsku godinu koja se određuje prema njegovom zvanju, međutim kada se ta norma prekorači, svi sljedeći održani sati nastave se pribrajaju u prekovremeni rad. Kako bi sustav registrirao kad je dostignuta predviđena norma potrebno je imati mehanizam provjere.

Kod 6.8 prikazuje jedno od mogućih rješenja, a to je sljedeće: vrši se provjera je li odrađena norma manja od predviđene, ukoliko je manja, ali s novim normom satima prelazi predviđeno. Zatim se izračunava višak norme sati koji se potom vraćaju u oblik običnih sati i pribrajaju se prekovremenim satima. U slučaju da je odrađena norma sati

već prije prešla odrađenu tada se svi sati pišu u prekovremene.

```
$prek_sql = "SELECT k.odradjena_norma, k.prekovremeni, t.norma FROM
korisnik k, titula t WHERE k.id = $kor_id AND k.titula_id = t.id";
$prek_rez = mysqli_query ($conn, $prek_sql);
$prek_niz = mysqli_fetch_array($prek_rez);
$topt = $prek_niz['norma'];
$opt = $prek_niz['odradjena_norma'];
$prek = $prek_niz['prekovremeni'];
if($opt < $topt){
    $razlika = 0;
    $razlika = $opt - $topt;
    $visak = $razlika + $pnorma;
    if($visak > 0)
        $prek = $prek + ($visak / ($z==1 ? $ts : $tv));
}
else{
    $prek = $prek + $brojsati;
}
```

Kod 6.8 Izračun prekovremenih

6.2.8 Generiranje PDF izvješća

Generiranje izvješća iz evidencije najprikladnije je u PDF formatu. Razlog tome je mogućnost preuzimanja i lakog pregledavanja dokumenta. Tu mogućnost pruža korištenje posebno napisana PHP klasa naziva mPDF koja generira PDF dokumente iz HTML datoteka, što uvelike olakšava oblikovanje izgleda PDF dokumenta (Back, 2015).

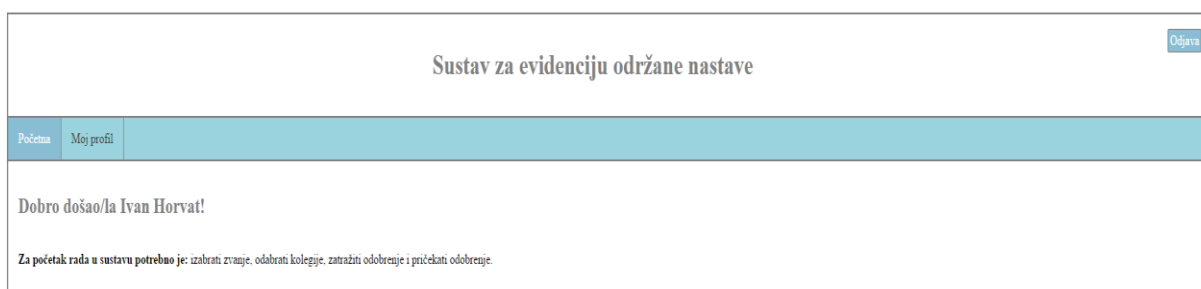
6.3 Korištenje web aplikacije

Početak korištenja aplikacije započinje dolaskom na stranicu za prijavu u sustav na kojoj se nalazi obrazac za prijavu. U sustav je moguće prijaviti se kao administrator ili korisnik. Svaka stranica unutar sustava ima omogućenu opciju odjave iz sustava. Slika 6.2 prikazuje obrazac za prijavu.



Slika 6.2 Obrazac za prijavu

Nakon uspješne prijave kao korisnik, otvara se početna stranica na kojoj se nalazi poruka dobrodošlice i određena poruka sa zahtjevima koje korisnik treba ispuniti za početak rada u sustavu. Konkretno, potrebno je izabrati zvanje, dodati kolegije za koje će korisnik održavati nastavu, zatražiti i pričekati odobrenje administratora. S početne stranice moguć je ulazak na stranicu „Moj Profil“. Izgled početne stranice vidljiv je na slici 6.3.



Slika 6.3 Izgled početne stranice

Sljedeći korak je postavljanje zvanja korisnika, klikom na „Moj profil“ otvara se nova

podstranica naziva „Korisnički podaci“ s formom u kojoj su podaci o korisniku. Prikazani podaci su: ime, prezime, korisničko ime, identifikacijski broj i zvanje, dok je nova lozinka obrazac za promjenu stare lozinke. Za promjenu zvanja potrebno je izabrati jedno od ponuđenih zvanja iz padajućeg izbornika. Ukoliko postoji potreba za izmjenom određenih podataka, moguće je promijeniti ime, prezime i zvanje. Na slici 6.4 vidi se izgled stranice Korisnički podaci.

Sustav za e

Početna Moj profil

Korisnički podaci

Korisnički podaci

Ime: Ivan

Prezime: Horvat

Korisničko ime: ihorvat

Nova lozinka:

Identifikacijski broj: 76817

Zvanje: -- izaberi zvanje --

Izmijeni

- izaberi zvanje --
- Proroktor
- Dekan
- Prodekan
- Pročelnik
- Zamjenik pročelnika
- Docent
- Izvanredni profesor
- Redoviti profesor
- Redoviti profesor u trajnom zvanju
- Predavač
- Viši predavač

Slika 6.4 Korisnički podaci

S postavljenim zvanjem mijenja se izgled stranice i pojavljuje se mogućnost odlaska na podstranicu „Kolegiji i nastavno opterećenje“. Sljedeći korak je odlazak na stranicu „Kolegiji i nastavno opterećenje“. Na njoj je moguće vidjeti predviđeno nastavno opterećenje po zvanju, predviđeno nastavno opterećenje, zbroj odrađenih sati i zbroj prekovremenih sati. Ukoliko ne postoji niti jedan odabrani kolegij od strane korisnika pojavljuje se poruka o nepostojanju kolegija, što je vidljivo na slici 6.5. Sljedeći korak je dodavanje kolegija i pripadajućeg broja predviđenih sati klikom na link „Dodaj kolegij“. Pritom se otvara stranica na kojoj se nalazi obrazac za unos kolegija za koji će korisnik održavati nastavu, a polja koje je potrebno unijeti su sljedeća: kolegij i tip nastave koji se biraju iz padajućih izbornika, te predviđeni broj sati nastave. Postoji mogućnost povratka klikom na link „Povratak“ koji vodi na prethodnu stranicu ukoliko korisnik odustane od unosa. Izgled stranice s obrascem prikazan je na slici 6.6.

Početna	Moj profil	
Korisnički podaci		Kolegiji i nastavno opterećenje Nastavno opterećenje prema tituli: 60 norma sati. Predviđeno nastavno opterećenje: 0 norma sati. Zbroj odrađenih norma sati: 0. Zbroj prekovremenih sati: 0. Ne postoji niti jedan kolegij! Dodaj kolegij
Kolegiji i nastavno opterećenje		

Slika 6.5 Kolegiji i nastavno opterećenje bez odabranih kolegija

Povratak	
Dodaj novi kolegij i predviđene sate! Kolegij: -- izaberi kolegij -- Tip nastave: -- izaberi tip nastave -- Predviđeni broj sati: Dodaj	

Slika 6.6 Dodavanje novog kolegija s predviđenim satima

Korisnik mora dodati sve kolegije na kojima održava nastavu, a postoji i mogućnost brisanja zapisa ukoliko je nešto pogrešno uneseno. Nakon što je to učinio ima mogućnost pritiska na link „Zatraži odobrenje“ kojim šalje obavijest administratoru da provjeri unesene podatke i ako su točni odobri korisnikovo nastavno opterećenje. U slučaju da uneseni podaci nisu točni administrator ima opciju odbijanja norme, a korisnik mora provjeriti u čemu je problem, ispraviti grešku te ponovno zatražiti odobrenje. Ukoliko je zatraženo odobrenje dodavanje novih kolegija i brisanje postojećih je onemogućeno. Izgled stranice „Kolegiji i nastavno opterećenje“ nakon unosa kolegija moguće je vidjeti na slici 6.7.

Kolegiji i nastavno opterećenje

Nastavno opterećenje prema tituli: 150 norma sati.
 Predviđeno nastavno opterećenje: 160 norma sati.
 Zbroj odrađenih norma sati: 0.
 Zbroj prekovremenih sati: 0.

[Dodaj kolegij](#)
[Zatraži odobrenje](#)

Naziv	Šifra	Tip	Br. predviđenih sati	Br. odrađenih sati	
Programiranje	PROG1	Predavanje	40	0	Obrisi
Dinamicke web aplikacije	DWA	Predavanje	40	0	Obrisi

Slika 6.7 Kolegiji i nastavno opterećenje s odabranim kolegijima

Onog trenutka kada administrator odobri korisnikovo nastavno opterećenje otvara se mogućnost odlaska na stranicu „Evidencija“ na kojoj se unose zapisi u evidenciju. Ukoliko nema unosa u evidenciju, prikazat će se poruka o nepostojanju zapisa (slika 6.8).

Početna

Moj profil

Evidencija

Evidencija održane nastave

[Novi zapis](#)

Nema zapisa u evidenciji!

Slika 6.8 Evidencija bez zapisa

Klikom na link „Novi zapis“ otvara se nova stranica na kojoj je ispunjavanjem obrasca moguće unijeti novi zapis u evidenciju (slika 6.9). Potrebno je izabrati kolegij i tip nastave iz padajućeg izbornika, datum kada je nastava održana (unazad 3 tjedna) i broj održanih sati nastave.

Povratak

Unos u evidenciju nastave

Kolegij i tip: -- izaberi kolegij i tip --

Datum:

Broj sati:

Pošalji

Slika 6.9 Obrazac za unos evidencije

Nakon prvog zapisa u evidenciju na stranici „Evidencija“ pojavljuje se tablica s prikazom zapisa u evidenciju kao što je vidljivo na slici 6.10.

Početna

Moj profil

Evidencija

Evidencija održane nastave

Novi zapis

Izradi izvješće

Naziv kolegija	Šifra	Tip nastave	Datum	Br. sati
Programiranje	PROG1	Predavanje	26/08/2016	5
Dinamičke web aplikacije	DWA	Predavanje	27/08/2016	6

Slika 6.10 Prikaz zapisa u evidenciji

Kada korisnik unese sve zapise o održanoj nastavi ima mogućnost pritiska na link „Izradi izvješće“ koji u novom tabu generira PDF izvješće o održanoj nastavi s podacima korisnika. Primjer PDF izvješća prikazan je na slici 6.11.



Sveučilište Jurja Dobrile u Puli
Zagrebačka 30, Pula
Centrala: +385 (52) 377 000
E-mail: ured@unipu.hr

Izvješće o održanoj nastavi

Izvješće o održanoj nastavi
za zaposlenika:
Ivan Horvat
Zvanje: Pročelnik
U razdoblju: 26.08.2016 - 27.08.2016

Datum: 14.09.2016

ŠIFRA	NAZIV	TIP PREDAVANJA	PREDVIDENIH SATI	ODRAĐENIH SATI
PROG1	Programiranje	Predavanje	40	105
DWA	Dinamičke web aplikacije	Predavanje	40	6
UKUPNO:			80	111
Norma prema tituli:				150
Norma predviđena:				160
Norma odrađena:				222
Broj prekovremenih sati:				36
Cijena prekovremenog sata:				180 KN
UKUPNO ZA ISPLATITI:				6480 KN

Potpis zaposlenika

Potpis pročelnika

Slika 6.11 Generirano izvješće o održanoj nastavi

6.3.1 Korištenje web aplikacije u svojstvu administratora

Ukoliko je korisnik prijavljen u svojstvu administratora način korištenja osnovnih funkcionalnosti je potpuno jednak kao kod korisnika, ali su mu omogućene i neke dodatne funkcionalnosti, a to su: pregledavanje i dodavanje novih korisnika i kolegija te uređivanje zvanja.

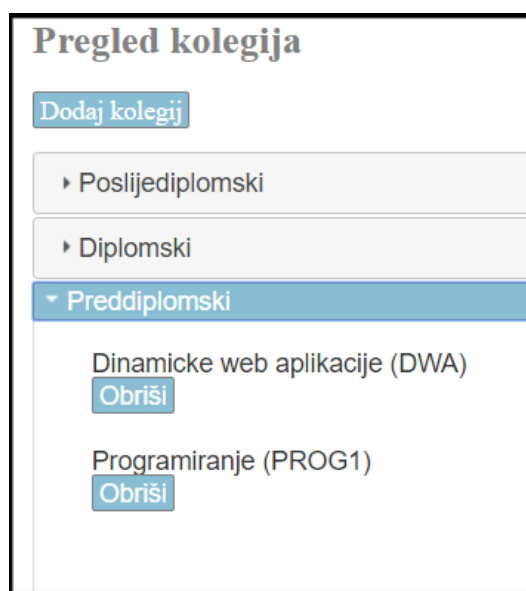
Klikom na stranicu „Pregled korisnika/kolegija/zvanja“ otvara se stranica pregleda korisnika. Na stranici „Korisnici“ mogu se vidjeti korisnici koji su kategorizirani u izbornicima po tipu korisnika (administrator ili korisnik). Ukoliko ne postoje podaci za prikaz unutar izbornika pojaviti će se poruka o nepostojanju podataka za prikaz. Za svakog korisnika moguće je pregledati evidenciju, ako ona postoji, obrisati korisnika, otključati mogućnost da korisnik uređuje odabrane kolegije i poslati obavijest e-mailom,

ako korisnik nije još zatražio odobrenje nastavnog opterećenja. Na slici 6.12 vidi se prikaz korisnika. Klikom na opciju „Dodaj korisnika“ odlazi se na novu stranicu koja sadrži obrazac za unos podataka, imena, prezimena, korisničkog imena, e-mail adrese i odabira je li korisnik vanjski suradnik. Po formi je sličan obrascu za unos evidencije.



Slika 6.12 Pregled korisnika

Sa stranice korisnika moguće je ići na stranice „Kolegiji“ i „Zvanja“. Stranica „Kolegiji“ vrlo je slična stranici „Korisnici“ po funkcionalnostima osim što prikazuje podatke o kolegijima koji su uneseni u sustav, dopušta unos novih kolegija i brisanje postojećih. Izgled stranice prikazan je na slici 6.13.



Slika 6.13 Pregled kolegija

Stranica „Zvanja“ prikazuje odabir mogućih zvanja nastavnika i njihovo uređivanje. Pritiskom na link „Uredi“ prelazi se na stranicu uređivanja koja sadrži obrazac s postojećim podacima, koje je moguće izmijeniti ukoliko dođe do promjene vrijednosti nastavnog opterećenja ili cijene prekovremenog sata. Izgled stranice „Zvanja“ prikazan je na slici 6.14, a izgled obrasca za izmjenu podataka o zvanju na slici 6.15.

Početna	Moj profil	Pregled korisnika/kolegija/z
Korisnici	Pregled zvanja Naziv: Prorektor Norma: 60 Cijena prekovremenog: 200 KN sat Uredi Naziv: Dekan Norma: 60 Cijena prekovremenog: 200 KN sat Uredi Naziv: Prodekan Norma: 150 Cijena prekovremenog: 180 KN sat Uredi	
Kolegiji		
Zvanja		

Slika 6.14 Pregled zvanja

Povratak	
Izmijeni zvanje Naziv: Prorektor Norma: 60 Cijena: 200 Izmijeni	

Slika 6.15 Obrazac za izmjenu podataka o zvanju

Stranica „Odobranje normi“ prikazuje listu korisnika koji su zatražili odobrenje nastavnog opterećenja, a moguće su dvije opcije: „Odobri normu“ kojom se odobrava norma korisniku i „Odbij normu“ kojom se odbija uneseno nastavničko opterećenje korisnika. Izgled stranice prikazan je na slici 6.16.

Početna	Moj profil	Pregled korisnika/kolegija/zvanja	Odobranje normi
Lista korisnika i normi za odobriti Bastijanić Luka Nastavno opterećenje prema zvanju: 150 norma sati. Predviđeno nastavno opterećenje: 170 norma sati. - Programiranje - PROG1 - Preddiplomski - Predavanje - Broj sati: 50 - Dinamicke web aplikacije - DWA - Preddiplomski - Predavanje - Broj sati: 35 Odobri normu Odbij normu			

Slika 6.16 Odobranje normi

7. Potencijalne buduće nadogradnje i poboljšanja sustava

7.1 Asynchronous JavaScript and XML

Jedna od potencijalnih nadogradnji ili poboljšanja sustava je korištenje Asynchronous JavaScript and XML (AJAX) tehnika. One omogućavaju izmjenu zatraženih podataka na web stranici bez da se cijela web stranica osvježi već samo dio za koji je to potrebno (Nixon, 2015). Korištenje AJAX tehnike, učinilo bi rad u aplikaciji puno bržim i efikasnijim.

7.2 Generiranje više vrsta izvještaja

Aplikacija zasad nudi generiranje samo jednog tipa izvještaja, onog o održanoj nastavi korisnika. Uspoređujući druge aplikacije može se vidjeti da one sadržavaju i druge tipove izvještaja koji su potencijalno korisni za obrazovnu ustanovu. Ponajprije izvještaj o održanoj nastavi za pojedini kolegij koji bi pružao uvid u količinu odrađenih sati nastave svih nastavnika.

7.3 Integracija s bazom podataka rasporeda sati

Kod usporedbe drugih aplikacija vidljivo je da su one uglavnom cjelovita rješenja za planiranje nastave, a neke imaju i dodatne funkcionalnosti. No, sve uglavnom podržavaju generiranje rasporeda i zajedničku bazu podataka iz koje se dobivaju podaci. Kako bi se olakšalo korisnicima, unos zapisa u evidenciju može se izvršiti integracijom baze podataka s bazom gdje se sprema raspored sati. Na taj način korisnik bi imao ponuđene termine u kojima je predviđena nastava za kolegije na kojima on predaje i bilo bi potrebno upisati samo broj održanih sati nastave. S tim poboljšanjem izbjegle bi se potencijalne slučajne ili namjerne korisničke greške poput unosa krivog datuma.

8. ZAKLJUČAK

Ovaj završni rad sastoji se od teorijskog i praktičnog djela razvoja aplikacije za evidenciju održane nastave. Evidencija nastave je bitan segment obrazovnog sustava kako bi se znalo u kojoj je mjeri nastava održana. Aplikacija uvelike pomaže administraciji ustanova jer smanjuje količinu papirologije i sprječava gubitak podataka ili lošu interpretaciju. Kako se radi o aplikaciji koja uvijek mora biti dostupna, odabrano je web okruženje. Na taj način omogućeno je korisnicima (zaposlenicima obrazovnih ustanova) da pristupaju aplikaciji u bilo kojem trenutku i s bilo kojeg mjesta što djeluje stimulativno. Tehnologije za razvoj same aplikacije odabrane su prema toj činjenici, a korištene su one poput skriptnih jezika PHP i JavaScript, dok je za upravljanje bazom podataka korišten MySQL, koje su kompatibilne s većinom platformi, web preglednika i web poslužitelja. Usporedbom postojećih i razvijene aplikacije utvrđeno je da dijele veći dio istih funkcionalnosti, iako razvijena pruža dodatnu pogodnost u izračunu iznosa za prekovremene sate, dok su postojeće aplikacije uglavnom cjelovita rješenja odnosno dijelovi većih sustava.

Aplikacija pruža korisnicima mogućnost unosa predviđenog nastavnog opterećenja za akademsku godinu, izmjenu osobnih podataka, unos zapisa u evidenciju održane nastave i izradu izvještaja na kraju akademske godine. Izvješće o održanoj nastavi bitan je segment aplikacije jer je to dokaz svakog nastavnika da je svoju nastavu održao.

Kako se radi o prvoj verziji aplikacije, u budućnosti su potrebne nadogradnje i poboljšanja sustava. Trenutna verzija sadržava osnovne funkcionalnosti no postoje i one koje bi se mogle implementirati, a pokazale bi se korisne. Primjer dodatne funkcionalnosti je generiranje više vrsta izvještaja koji bi bili namijenjeni ne samo korisniku već i obrazovnoj ustanovi. Nadogradnja sustava mogla bi se postići implementiranjem AJAX tehnika kako bi dobili bržu i funkcionalniju web stranicu. Postojeće aplikacije rješenja su koja nude kompletno planiranje nastave, pa bi dobra nadogradnja razvijene aplikacije bila integracija sa sustavom za izradu rasporeda sati te uvođenje novih funkcionalnosti koje bi omogućile planiranje nastave.

LITERATURA

Back , I. (2015.) *mPDF Manual* [online]. s.l., Dostupno na: <https://mpdf.github.io/> [13.09.2016]

Fakultet elektrotehnike i računarstva (2016.) *Quilt CMS: Ukratko o sustavu* [online]. Zagreb. Dostupno na: <https://www.fer.unizg.hr/cms> [11.09.2016]

Grd, D. (2016). *Evidencija nastave i raspored* [online]. Varaždin. Dostupno na: <http://www.dario-grd.iz.hr/detailed/hr/nastava> [11.09.2016]

jQuery user interface (2016.) *jQuery UI* [online]. s.l., Dostupno na: <https://jqueryui.com/> [13.09.2016]

Košak, B.; Tomiša, M.; Čačić, M. (2015.) Statičko i dinamičko upravljanje web sadržajem, *Tehnički glasnik*, 9, str. 77-83

Lama Informacijske Tehnologije (2016.) *EduPlan* [online]. Split. Dostupno na: <http://www.lama.hr/rjesenja/aplikativna-rjesenja/eduplan/> [11.09.2016]

Mozilla Developer Network (2015.) *About JavaScript* [online]. s.l., Dostupno na: https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript [12.09.2016]

Nixon, R. (2015) *Learning PHP, MySQL & JavaScript with jQuery, CSS & HTML5*. 4. izd. Sebastopol: O'Reilly

php.com.hr (2008.) *Uvod u PHP* [online]. s.l., Dostupno na: <http://php.com.hr/77> [12.09.2016]

Radovan, M. (2009.) *Dinamičke web aplikacije* [digitalna skripta], s.n.

tutorialspoint (2016.) *Learn MySQL* [online]. s.l., Dostupno na: <https://www.tutorialspoint.com/mysql/> [12.09.2016]

Ullman, L. (2012.) *PHP and MySQL for Dynamic Web Sites, Fourth Edition*, 4. izd.
Berkeley: Peachpit Press

W3Schools (2016.) *CSS tutorial* [online]. s.l., Dostupno na:
<http://www.w3schools.com/css/> [12.09.2016]

W3Schools (2016.) *HTML5 tutorial* [online]. s.l., Dostupno na:
<http://www.w3schools.com/html/> [12.09.2016]

W3Schools (2016.) *JavaScript tutorial* [online]. s.l., Dostupno na:
<http://www.w3schools.com/js/> [13.09.2016]

W3Schools (2016.) *SQL tutorial* [online]. s.l., Dostupno na:
<http://www.w3schools.com/SQL/> [12.09.2016]

POPIS SLIKA

Slika 3.1 Pregled korisnika i njihovih normi sati za određenu akademsku godinu	7
Slika 4.1 Komunikacija između klijenta i servera kod statičkih web stranica	12
Slika 4.2 Komunikacija između klijenta i servera kod dinamičkih web stranica.....	13
Slika 5.1 Hodogram korištenja aplikacije	23
Slika 6.1 ERD model	24
Slika 6.2 Obrazac za prijavu	33
Slika 6.3 Izgled početne stranice	33
Slika 6.4 Korisnički podaci	34
Slika 6.5 Kolegiji i nastavno opterećenje bez odabranih kolegija.....	35
Slika 6.6 Dodavanje novog kolegija s predviđenim satima	35
Slika 6.7 Kolegiji i nastavno opterećenje s odabranim kolegijima.....	36
Slika 6.8 Evidencija bez zapisa	36
Slika 6.9 Obrazac za unos evidencije.....	37
Slika 6.10 Prikaz zapisa u evidenciji	37
Slika 6.11 Generirano izvješće o održanoj nastavi	38
Slika 6.12 Pregled korisnika	39
Slika 6.13 Pregled kolegija	39
Slika 6.14 Pregled zvanja	40
Slika 6.15 Obrazac za izmjenu podataka o zvanju	40
Slika 6.16 Odobravanje normi	41

POPIS KODA

Kod 4.1 Primjer CSS klase	16
Kod 4.2 Jednostavna PHP skripta unutar HTML strukture	18
Kod 4.3 Funkcija zbrajanja u PHP-u	18
Kod 4.4 Primjer MySQL naredbe za kreiranje tablice	20
Kod 4.5 Primjer upotrebe PHP funkcija za interakciju s MySQL-om.....	20
Kod 6.1 Rješenje problema identifikacijskog broja	27
Kod 6.2 Rješenje problema generiranja lozinke	28
Kod 6.3 Automatizirano slanje e-mail-a	28
Kod 6.4 Korištenje accordion-a.....	29
Kod 6.5 Regularni izrazi	30
Kod 6.6 Pozivanje JavaScripta putem događaja onSubmit	30
Kod 6.7 Korištenje datepicker-a.....	31
Kod 6.8 Izračun prekovremenih.....	32

SAŽETAK

Ovaj rad se temelji na teorijskom i praktičnom djelu razvoja web aplikacije za evidenciju održane nastave. Kroz teorijski dio obrađeni su pojmovi evidencije i nastave, osnovni zahtjevi koje aplikacija mora zadovoljavati, postojeće aplikacije, vrste web stranica, definicija web aplikacije i korištene tehnologije (HTML, CSS, PHP, MySQL, JavaScript i jQuery). Praktični dio sadrži hodogram korištenja aplikacije, bazu podataka i opise tablica, rješenja specifičnih problema i opis načina korištenja web aplikacije uz koji su priložene slike izgleda sustava radi lakšeg snalaženja pri korištenju.

Ključne riječi: evidencija održane nastave, web aplikacija, web stranica, evidencija, nastava, HTML, CSS, PHP, MySQL, JavaScript

ABSTRACT

This thesis is composed of theoretical and practical part of web application development. Application's purpose is keeping record of held classes. Theoretical part deals with terms such as record keeping and teaching, application's fundamental requirements which should be met, existing software solutions, website types, definition of web application and technologies used for implementation of program solution (HTML, CSS, PHP, MySQL, JavaScript i jQuery). Practical part contains workflow of using the application, database and tables description, solutions for specific problems and description of web application usage with pictures for its convenient using.

Keywords: record keeping of held classes, web application, record keeping, website, teaching, HTML, CSS, PHP, MySQL, JavaScript