Aplikacija za kolaboraciju unutar IT tima

Ćališ, Verino

Undergraduate thesis / Završni rad

2018

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Algebra University College / Visoko učilište Algebra**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:225:385111

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

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

Repository / Repozitorij:

Algebra Univerity College - Repository of Algebra Univerity College

VISOKO UČILIŠTE ALGEBRA

ZAVRŠNI RAD

Aplikacija za kolaboraciju unutar IT tima

Verino Ćališ

Zagreb, veljača 2018.

„Pod punom odgovornošću pismeno potvrđujem da je ovo moj autorski rad čiji niti jedan dio nije nastao kopiranjem ili plagiranjem tuđeg sadržaja. Prilikom izrade rada koristio sam tuđe materijale navedene u popisu literature, ali nisam kopirao niti jedan njihov dio, osim citata za koje sam naveo autora i izvor, te ih jasno označio znakovima navodnika. U slučaju da se u bilo kojem trenutku dokaže suprotno, spreman sam snositi sve posljedice uključivo i poništenje javne isprave stečene dijelom i na temelju ovoga rada“.

U Zagrebu, datum.

Ime Prezime

Predgovor

Na prvome mjestu posebno sam zahvalan roditeljima koji su mi omogućili obrazovanje, pomagali i uvijek bili spremni pružiti podršku u bilo kojem smislu te riječi.

Zahvaljujem profesorima koji su prenijeli svoje znanje i iskustvo.

Prilikom uvezivanja rada, Umjesto ove stranice ne zaboravite umetnuti original potvrde o prihvaćanju teme završnog rada kojeg ste preuzeli u studentskoj referadi

Sažetak

U ovome radu cilj je izraditi aplikaciju koja će prvenstveno doprinijeti komunikaciji unutar tima ljudi. Jednostavnost same aplikacije treba olakšati komunikaciju između istih i/ili raznih odjela. Pod komunikacijom se podrazumijeva razmjena datoteka, obavijesti članovima tima, zadaci, jasna hijerarhija, te prikaz učinkovitosti. Komunikacija između članova informacijskog tima je bitna zbog različitih timova i razina, a u većini slučajeva svi moraju biti uključeni ili obaviješteni o trenutnom stanju projekta. Kroz iskustvo u radu i komunikaciji u manjim timovima je ujedno bila i motivacija za izradu ovoga projekta iz razloga što sam pokušao uočene nedostatke pretočiti u rješenje tih nedostataka. Prilikom izrade rada posjedovao sam napredno znanje HTML, CSS, JavaScripta i osnovno znanje PHP tehnologija. Web aplikacija je rješenje jer je omogućen pristup s raznih uređaja koji imaju pristup internetu.

Ključne riječi: komunikacija, web aplikacija, jednostavnost, tim ljudi

Summary

In this project the goal is to create an application that will primarily contribute to communication within the team of people. The simplicity of the application itself should faciliate communication between the same or different departments. Communication, in this case, involves exchange of files, team members notice, assignments, hierarchy of responsibilities and the perfomrnace. Communication between IT team members is important for a variety of teams and levels, and in most cases everyone must be involved or informed about the current state of the project. Through personal experience in work and communication in smaller teams was the motive behind this project because I tried to spot the solution of these shortcomings. During the development of the project I possessed advanced knowledge of HTML, CSS, JavaScript and the basic knowledge of PHP technology.

Keywords: communication, web application, simple, team of people

Sadržaj

1. Uvod 1
2. Korištene tehnologije 2
   2.1. HTML 2
   2.2. SASS i CSS 3
   2.3. SASS 3
   2.4. PHP 4
   2.5. JavaScript i jQuery 5
   2.6. Baze podataka i SQL 5
   2.7. PhpMyAdmin 6
   2.8. XAMPP 6
3. Korisničko iskustvo 7
   3.1. Primjena korisničkog iskustva 8
   3.2. Elementi korisničkog iskustva 9
      3.2.1. Strategija 9
      3.2.2. Opseg 9
      3.2.3. Struktura 10
      3.2.4. Nacrt 10
      3.2.5. Vizualni dizajn 10
   3.3. Funkcionalnosti 11
4. Izrada aplikacije 12
   4.1. Struktura 12
   4.2. Registracija u sustav 13

4.3. Prijava u sustav ... 14
4.4. Popis zadataka ... 14
4.5. Razgovori ... 16
4.6. Datoteke ... 18
4.7. Tablice baze podataka ... 20
4.7.1. Korisnici ... 20
4.7.2. Timovi ... 20
4.7.3. Obavijesti ... 21
4.7.4. Zadaci ... 21
4.7.5. Datoteke ... 22
4.7.6. Razgovori ... 22
4.8. Unos, dohvaćanje, brisanje i uređivanje podataka ... 23
4.8.1. Unos podataka ... 23
4.8.2. Dohvaćanje podataka ... 23
4.8.3. Izmjena podataka ... 24
4.8.4. Brisanje podataka ... 24
4.9. Sesija ... 24
4.10. Sigurnosni aspekt ... 25
Zaključak ... 26
Popis kratica ... 27
Popis slika ... 28
Popis kôdova ... 29
Literatura ... 30

1. Uvod

Skupina ljudi koja radi na istome projektu ili određenome proizvodu mora međusobno komunicirati kako bi, za početak, razvila ideju, skupila misli i ideje, razvila i na kraju završila, odnosno privela kraju taj određeni proizvod. Takav proces zahtjeva aplikaciju koja bi olakšala komunikaciju, raspodjelu poslova, postavljanje rokova, praćenje napretka, razmjena datoteka poput tekstnih dokumenata, slika i slično.

Veliki naglasak se stavlja na jednostavnost jer je potrebno da svaki član zna na čemu trenutno radi, što ga čeka za napraviti, ali i povijest izrade.

Rad će se sastojati od web aplikacije koja omogućava prije navedene stvari. Za početak, potrebno je napraviti sustav koji omogućava registraciju i prijavu korisnika. Za takav sustav potrebna je baza podataka koja sadrži sve informacije korisnika, ali i ostale podatke potrebne za rad aplikacije. Korisnici koji rade na projektu imati će jedno, a administratori drugo sučelje za upravljanje. Korisnici bez dodatnih privilegija mogu pregledavati, čitati informacije, no ne mogu uređivati informacije jer je to previđeno za administratore.

Cilj je pojednostaviti proces komunikacije i upravljanje zadacima u manjem timu ljudi kako bi se u svakome trenutku moglo provjeriti kako napreduje projekt.

Struktura rada je takva da se u prvome dijelu, uvodu, obrazloži svrha i problem. U drugome dijelu postavljanje samoga zadatka, popis i objašnjenje korištenih tehnologija. Treći dio je posvećen korisničkom sučelju i funkcionalnostima. U četvrtome dijelu je detaljno objašnjenje programskog koda koji je korišten za izradu aplikacije. Peti dio rada je rezerviran za analizu izrade rada, korisnosti aplikacije, problemi prilikom izrade, te mogućnost daljnjeg razvijanja.

2. Korištene tehnologije

U ovome poglavlju biti će predstavljene korištene tehnologije. Svaka tehnologija će biti pojašnjena općenitom primjenom i konkretno u ovome radu. Uz primjere će biti jasnija potreba za određenom tehnologijom i po potrebi dodatna objašnjena.

2.1. HTML

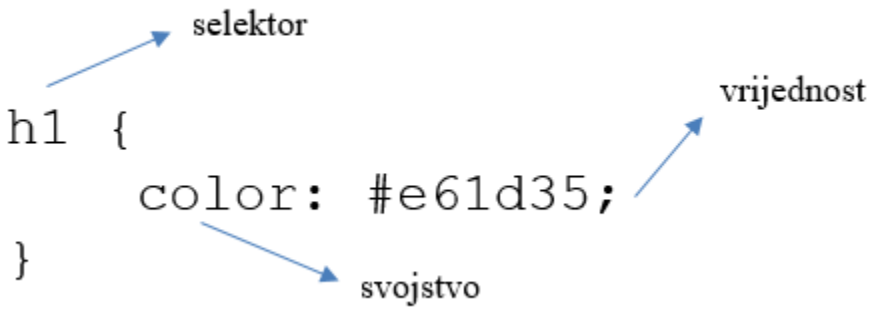
HTML (engl. *HyperText Markup Language*) je osnovni jezik za prezentaciju i strukturiranje sadržaja na web stranicama. Web preglednik na osnovu ovoga jezika oblikuje stranicu onakvom kakvom ju mi kao korisnici vidimo. Prva verzija, iako neslužbena, objavljena je 1991. godine i to je verzija Tim Berners Lee-a. Ovim jezikom se stvara hipertekstualni dokument. Hipertekstualni dokument se od običnog razlikuje u tome što sadrži hiperveze, a pomoću hiperveza je jedan dokument povezan s drugim takvim dokumentom. HTML kao jezik je vrlo popularan i raširen iz razloga što je besplatan, dostupan i početna točka svakoga tko želi krenuti sa radom na internetu u smislu prezentacije sadržaja u internet pregledniku. HTML je najpotrebniji iz razloga što služi za samu strukturu internet stranice, sljedeća vizualizacija će nam to demonstrirati:

```
<html>
  <head>
    <title>Page title</title>
  </head>
  <body>
    <h1>This is a heading</h1>
    <p>This is a paragraph.</p>
    <p>This is another paragraph.</p>
  </body>
</html>
```

Slika 2.1 Primjer HTML strukture

2.2. SASS i CSS

CSS je jezik kojim se oblikuje stil za HTML dokumente. Omogućava nam da utječemo na način prikaza teksta i svih ostalih elemenata HTML dokumenta. Ukratko, definira kako će se određeni elementi prikazivati. Uobičajeni primjeri su slova, poveznice, slike, grupe elemenata. Glavna prednost je olakšan i ubrzan pristup uređivanju internet stranica. Doprinosi općenito boljoj dosljednosti dizajna jer možemo, na primjer, definirati stil za naslove i onda koristiti taj stil za sve naslove koje koristimo.



Slika 2.2 Izgled CSS koda

2.3. SASS

SASS je dorada, proširenje samoga CSS-a. SASS nam omogućuje brojne dodatne stavke poput pisanja funkcija, pravila, varijabli. Ovdje je bitno istaknuti da postoje dvije verzije sintakse[1]. Jedna verzija je SASS u kojoj se ne koriste zagrade prilikom dodavanja svojstva i vrijednosti kao što je prikazano na slici (Slika 2.1). Druga verzija koristi naziv SCSS koja koristi zagrade. Glavna prednost SASS-a u odnosu na CSS je ugnježđivanje. To nam omogućuje da budemo precizniji prilikom odabira HTML elementa na koji želimo dodati određeni stil.

[1] Sintaksa je niz pravila koji nas upućuju kako i na koji način nešto koristiti.

Slika 2.2 prikazuje kako jedan jednostavan dio SASS koda izgleda.

```
nav {
  ul {
     margin: 0;
  }
  li {
     display: inline-block;
   }
}
```

Slika 2.3 SASS primjer ugnježđivanja

U ovome primjeru vidljivo je kako je glavni element `nav` i unutar njega ostali elementi. U cijeloj priči najbitniji je pojam interpreter. To je program koji izvršava izvorni kod napisan u nekome od programskih jezika. U ovome primjeru, sve što je napisano u SASS-u, biti će prevedeno u osnovni CSS koji svaki internet preglednik razumije i uspijeva ga izvršiti.

2.4. PHP

PHP je programski jezik koji služi kao podrška internet aplikacijama. Stvoren je kako bi se njegov kod mogao ugrađivati unutar HTML datoteka. Moguće je pisati i izvoditi samostalne PHP programe, no to nije osnovna ideja. Jednostavan primjer korištenja PHP i HTML koda:

```
<?php

$ime = 'Marko';

echo '<p>Moje ime je </p>'.$ime

?>
```

Kod 2.1 Primjer korištenja PHP-a unutar HTML koda

2.5. JavaScript i jQuery

JavaScript je skriptni jezik kojim dodajemo interaktivnost i dinamičnost na našim HTML stranicama. JavaScript je jedan on najpopularnijih skriptnih jezika na svijetu. Neke od osnovnih i najzanimljivijih primjena su: detektiranje preglednika(koji preglednik, verzija, …), radnje vezane za kolačiće(engl. *cookies*), validacija polja, padajući meni.

jQuery je biblioteka (engl. *library*) za JavaScript. Omogućuje nam da sa manje linija koda napravimo funkcionalnosti JavaScript-a. Sa jQuery možemo raditi HTML i CSS manipulacije, efekte i animacije, koristiti ga u kombinaciji sa AJAX metodom komunikacije. Primjer korištenja jQuery-a nam pokazuje Kod 2.2 Primjer korištenja jQuery-a:

```
$(selektor).akcija()
```

Kod 2.2 Primjer korištenja jQuery-a

Znakom dolara pristupamo jQuery-u. Selektorom pronalazimo HTML elemente nad kojima želimo izvršiti određenu radnju, dok je akcija nakon točke selektora ono što zapravo radimo, manipuliramo.

2.6. Baze podataka i SQL

Baze podataka su veoma bitne i njihova potreba je velika. Služi nam za spremanje podataka koji se kasnije mogu dohvatiti i dalje obrađivati. Ukratko, skup organiziranih podataka. SQL je jezik koji se koristi za komunikaciju s bazom podataka. Baza podataka se smatra računalnim programom pa iz toga razloga postoji više vrsta baza podataka. U ovome radu koristiti će se MySQL[2] baza podataka. Osnovne vrste operacija, to jest vrste upita su: SELECT, INSERT, UPDATE i DELETE.

[2] MySQL je sustav otvorenog kôda za upravljanje bazom podataka.

2.7. PhpMyAdmin

PhpMyAdmin je aplikacija napisana u programskom jeziku PHP koja služi za upravljanje MySQL bazama podataka. Aplikaciju pokrećemo tako što u adresnu traku upišemo: http://localhost/phpmyadmin.

Neke od značajka PhpMyAdmin-a su:

- Web sučelje
- Uvoz podataka u obliku CSV
- Izvoz podataka
- Kreiranje PDF grafika
- Rad sa raznim operacijskim sustavima[3]

2.8. XAMPP

XAMPP je besplatan alat otvorenog koda. Serverski paket koji služi za jednostavnu instalaciju Apache servera[4]. Namijenjen je za korištenje na lokalnoj mreži, odnosno služi da se kreira server zbog testiranja. Jednostavno rečeno, XAMPP nam omogućuje da umjesto kupovine prostora i domene simuliramo u izvanmrežnom načinu rada. Sve datoteke kojima želimo pristupiti spremaju se u korijenski direktorij. Samo pristupanje se vrši kroz internet preglednik putem sljedeće adrese: http://localhost/ i ime mape ili ime datoteke.

[3] Operacijski sustav ili operativni sustav je skup osnovnih sustavnih programa koji upravljaju sklopovljem računala radi ostvarivanja osnovnih zadaća računala.
[4] Apache server je besplatni web polsužitelj.

3. Korisničko iskustvo

Korisničko iskustvo (engl. *User experience*) je najjednostavnije rečeno iskustvo koje korisnik doživljava prilikom korištenja određenog proizvoda, u ovome slučaju web aplikacije. Kao dizajner potrebno je postaviti pitanja poput kako se korisnik osjeća, što želi pronaći, kako i na koji način želi da mu se informacije prezentiraju, kako istaknuti važnosti. U današnje vrijeme se poprilično puno pažnje pridodaje ovome segmentu prilikom planiranja i izrade proizvoda. Uvedena su istraživanja korisnika na principu postavljanja pitanja kako oni doživljaju proizvod, izrada prototipa i nekoliko verzija proizvoda i onda se provodi proces prikupljanja informacija koje vode do uspješnog korisničkog iskustva.

Slika 3.1 Pet elemenata korisničkog iskustva (planiranje)

3.1. Primjena korisničkog iskustva

Prilikom planiranja završnog rada trebala se postići jednostavnost i istražiti tržište sličnih aplikacija. Potrebno je napraviti dizajn koji će biti intuitivan i u isto vrijeme moderan. Kako je aplikacija zamišljena da korisnici brzo dolaze do potrebnih informacija nema puno odvojenih stranica. Većina informacija koje korisnik treba su na naslovnoj stranici. Prikaz kratkih i važnih obavijesti je prezentiran kroz ikonu zvona koja je poznata po uprave tome i broj koji prikazuje broj nepročitanih obavijesti.

Slika 3.2 Prikaz obavijesti unutar aplikacije

Zadaci koji su korisniku zadani prikazani su tablici koja ima dvije boje kako bi se istakao naslov određene kolone i njegovi pripadajući podaci. Brojčani podaci poput raznih statistika su prikazani u obliku grafova. Korisnik brže dolazi do informacije iz grafa koji ima jasno naznačene podatke nego da ih traži u raznim tablicama i mora ih iščitavati i u isto vrijeme uspoređivati.

Jednako je bitna i responzivnost aplikacije kako bi korisnici mogli pristupiti s raznih uređaja poput mobitela ili tableta. Kako se rezolucija ekrana smanjuje, tako se elementi raspoređuju jedni ispod drugih.

Na mobitelima su korisnici naviknuti da sadržaj stranice pomiču od gore prema dole(engl. *scrolling*). Administratorsko sučelje je podijeljeno u dvije kategorije, a to su korisnik i administrator. Potrebno je odvojiti ove dvije stvari jer administrator također u isto vrijeme može biti i korisnik i administrator. Jednostavnosti radi, odvojenost tih dviju kategorija pridodaje snalažljivosti i učinkovitosti aplikacije.

3.2. Elementi korisničkog iskustva

Korisnik koji dolazi do internet stranice ili aplikacije proživljava određeno iskustvo. Takvo iskustvo se može razumjeti kroz pet koraka. Svaki korak je sloj za sebe, ali se međusobno nadograđuju jedan na drugi. Slojevi se kreću od apstraktne do konkretne razine[5].

3.2.1. Strategija

Strategija je prvi korak. U ovome koraku potrebno je osmisliti i navesti razlog proizvoda, odnosno projekta. Postaviti pitanja poput zašto se radi na projektu ili proizvodu, za koga, zašto bi netko koristio ili ne bi koristio. Cilj je definirati korisnikove potrebe i poslovne ciljeve.

Unutar korisnikovih potreba vrši se istraživanje korisnika, izrada persona pomoću kojih stvaramo osobne koje su idealne osobe za korištenje našeg proizvoda.

Poslovni ciljevi su zapravo primarni ciljevi koje želimo ostvariti, dakle postavljanje ciljeva i kako ih ostvariti. Skica dizajna i koje metrike koristiti za mjerenje uspjeha.

3.2.2. Opseg

Nakon definiranja što mi želimo i na kraju krajeva korisnici, potrebno je definirati kako zadovoljiti strateške ciljeve. Prijelaz sa strategije na opseg dogoditi će se kada ciljeve pretvorimo u točno definirane zahtjeve kakav sadržaj i funkcionalnosti želimo. Možemo ih podijeliti na sadržajne i funkcionalne zahtjeve.

Funkcionalni zahtjevi su zahtjevi vezani za značajke proizvoda, kako međusobno funkcioniraju i slično. Sadržajni zahtjevi su informacije koje moramo moći ponuditi korisniku(slike, glazba, video i slično).

[5] Izvor: https://medium.com/omarelgabrys-blog/ux-a-quick-glance-about-the-5-elements-of-user-experience-part-2-a0da8798cd52

3.2.3. Struktura

Struktura definira kako korisnik komunicira, sudjeluje u samome proizvodu, odnosno kako se proizvod ponaša kada on vrši interaktivnost nad njime. Razvijanje konceptualne strukture. Dijeli se na dvije komponente: interaktivnost dizajna i informacijska arhitektura.

Interaktivnost je uspješno obavljena ako korisnik ostvaruje svoje ciljeve, informiranje korisnika o njegovim radnjama(npr. obavijest o spremljenim promjenama profila), sprječava korisnika da napravi pogrešku.

Informacijska arhitektura je uspješna kada su informacije unutar proizvoda organizirane, kategorizirane na temelju korisnikovih potreba. Jednostavnost prezentiranja informacija.

3.2.4. Nacrt

Nacrt ili skica je detaljnije razvijanje strukture. Zapravo je to određivanje vizualne forme na zaslonu. Strukturu činimo konkretnom. Identificira se specifičan dizajn poput navigacije i prezentacije informacija. Kako će se korisnici kretati unutar prezentiranih informacija, kako su one prezentirane. Nacrt dijelimo na tri cjeline dizajna: sučelje, navigacija i informacija.

Dizajn sučelja znači predstaviti i urediti sučelje na način da korisnik komunicira sa funkcionalnostima koje su postavljenje ispred njega.

Dizajn navigacije je strukturiranje hijerarhije, potrebno je staviti stvari ondje gdje ih korisnici očekuju.

Dizajn informacija je grupiranje i postavljanje elemenata na njihova mjesta.

Ova faza nam daje odgovore na pitanja poput kakva je vizualna forma svih stvari koje se nalaze na sučelju, kako se korisnik kreće, u ovome slučaju, unutar aplikacije.

3.2.5. Vizualni dizajn

Nakon svih faza, definiranja i istraživanja, na red dolazi kako će sve to vizualno izgledati. Potrebno je sve funkcionalnost i informacije prezentirati na način da je korisnicama vizualno odmah jasno kako što funkcionira.

3.3. Funkcionalnosti

Funkcionalnosti su podijeljene u dvije kategorije, grupe: korisnik i administrator.

Korisnik na početku ima odabir prijave u sustav ili registracija. Registracijom korisnik upisuje osnovne podatke(korisničko ime, lozinku, email adresu i bira kojem timu pripada) i šalje zahtjev za pristup aplikaciji. Prijava omogućuje korisniku da pristupi svome profilu. Na naslovnoj stranici korisnik može pregledati pročitane i nepročitane brze obavijesti koje administrator šalje na razini korisnika ili na grupi korisnika. Tablica sa zadacima daje uvid u opis zadatka, datum izvršavanja(engl. *deadline*), status i mijenja statusa. Kroz grafove korisnik vidi svoj učinak, omjer korisnika i administratora, te opće informacije o stanju projekata. Na posebnoj stranici, kojoj se pristupa iz glavne navigacije, nalaze se razgovori(engl. *chat*) gdje korisnici mogu međusobno komunicirati. Također, u glavnoj navigaciji nalazi se stranica za razmjenu datoteka(engl. *File-sharing*) gdje korisnici mogu učitati(engl. *upload*), preuzeti(engl. *download*) i pregledati sve datoteke.

Administrator ima odvojenu stranicu za upravljanje sustavom i korisnicima. Omogućen mu je uvid u popis korisnika i mijenjanje podataka koje je korisnik unio prilikom registracije, mijenja grupe korisnika, dodaje i uređuje zadatke i obavijesti. Sučelje sadrži tablicu u kojoj administrator vidi popis registriranih korisnika kojima nije dopušten pristup te ima odabir hoće li dozvoliti ili ne. Na vrhu stranici se nalazi se sekcija predviđena za administratorove obaveze. Kroz tu sekciju administratoru se prikazuju njegove obveze.

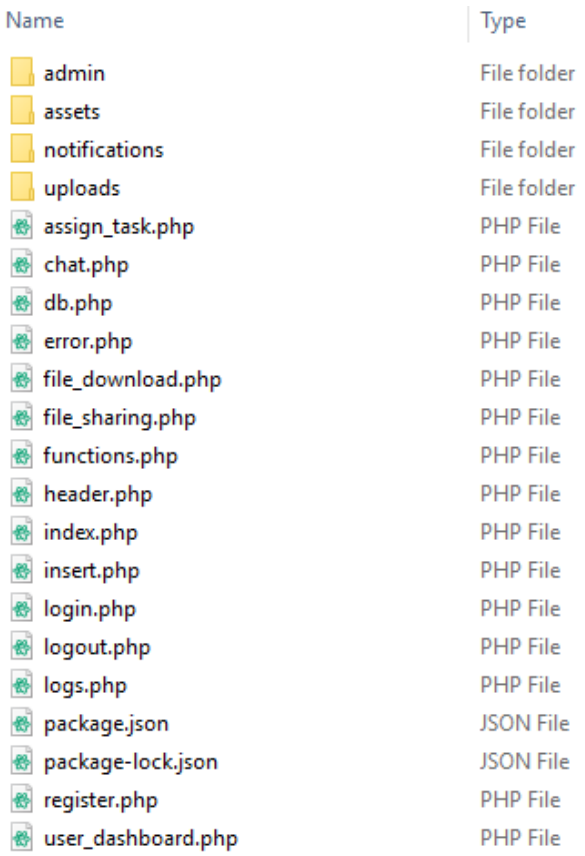
Korisnici i administratori mogu međusobno komunicirati kroz razgovore.

4. Izrada aplikacije

U ovome dijelu biti će objašnjen postupak izrade aplikacije iz aspekta programiranja. Objelodaniti će se kako aplikacija funkcionira i kako su određeni dijelovi odrađeni.

4.1. Struktura

Struktura direktorija ovog projekta vidi se na Slika 4.1 Struktura direktorija.



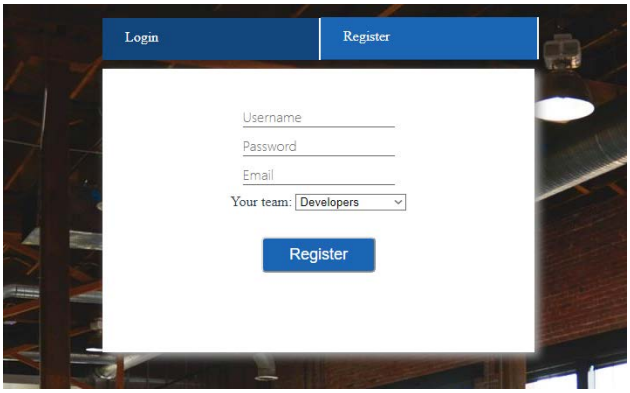
Slika 4.1 Struktura direktorija

Prilikom kreiranja samoga projekta bitno je napraviti skicu i plan kako će se datoteke i mape spremati. Bitno je nazivati datoteke i mape kako bi se iz samog naziva moglo zaključiti o čemu se radi.

Mapa nazvana „admin“ sadrži datoteke koje koriste za administratorski dio aplikacije. Mapa „assets“ se može gledati na način da su to sve dodaci vitalnim dijelovima, poput stilova, raznih kodova koji pridonose funkcionalnosti, slike i razvojno bazirane datoteke. Također, kako bi se olakšalo snalaženje i održavanje određeni dijelovi aplikacije su odvojeni u posebne mape poput „notifications“. Glavne datoteke projekta su u korijenskoj mapi projekta.

4.2. Registracija u sustav

Kako bi se korisnik registrirao potrebno je ispuniti prijavnicu.

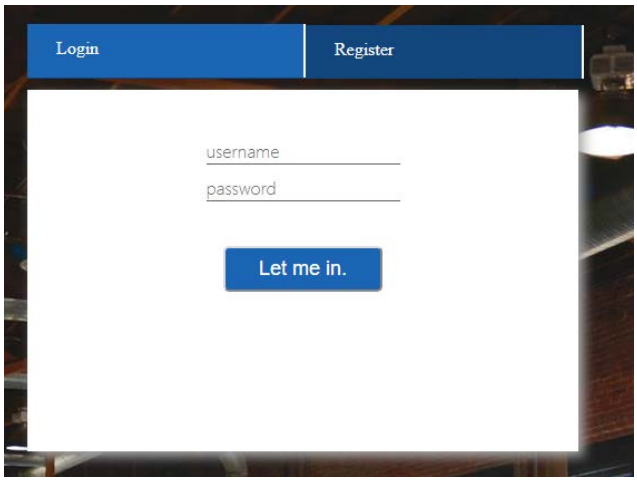


Slika 4.2 Registracija

Korisnik unosi željeno korisničko ime, lozinku, email i odabire kojem timu pripada. Osim podataka koje korisnik unosi, u pozadini se spremaju sljedeći podaci: datum registracije, automatski generirana kombinacija brojeva i slova, te tip korisnika. Korisnički tip se automatski postavlja na običnog korisnika(nema nikakva dodatna prava). Slijedi provjera ukoliko korisničko ime već postoji. Ukoliko postoji, korisnika se preusmjerava na stranicu predviđenu za pogreške, ispisuje se pogreška. Stranica za pogreške prilikom registracije će se također prikazati korisniku ako je unio pogrešnu lozinku ili email adresu sa prikladnom porukom i mogućnošću vraćanja na početnu stranicu. Nakon provjera, podaci se šalju u bazu podataka gdje se spremaju.

4.3. Prijava u sustav

Nakon uspješno odrađene registracije korisnik je zatražio pristup korištenja aplikacije i čeka administratorovo odobrenje. Kada administrator odobri prijavu može početi koristiti aplikaciju.



Slika 4.3 Prijava u sustav

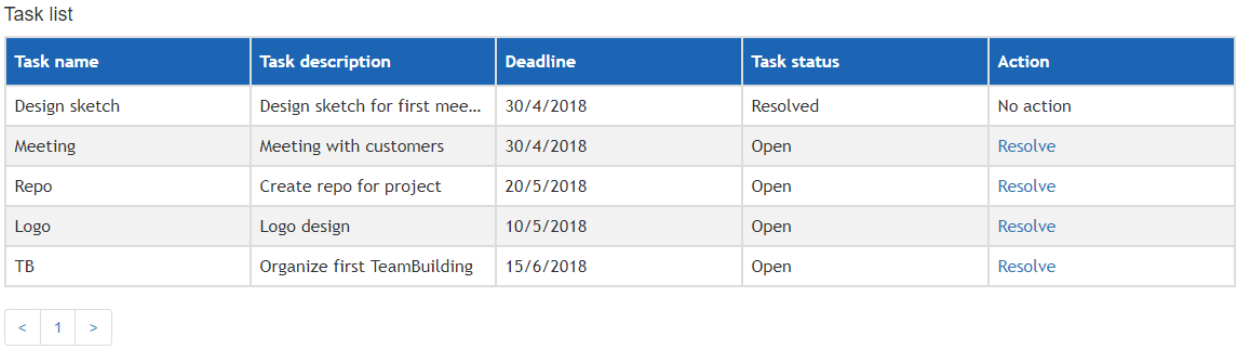
Unošenjem ispravnog korisničkog imena i lozinke pristupa se početnoj stranici. U slučaju neispravnih podataka prikazuje se ista stranica za pogreške kao i kod registracije sa odgovarajućim porukama koje sugeriraju koji su točno podaci neispravni. To može biti pogrešna lozinka i/ili korisničko ime. Ovdje je bitno istaknuti da se prilikom prijave određuje koje sučelje će se prikazati, ono korisničko ili administratorovo.

4.4. Popis zadataka

Popis zadataka je jedna od ključnih stavki u aplikacijama poput ove. Potrebno je da svaki član u svakome trenutku zna što i do kada radi. Kada administrator, sam ili u suradnji s voditeljem projekta, doda zadatak, on će se kao takav prikazati članu tima.

Prikaz zadatak je ostvaren kroz tablicu koja je jednostavna i na jednostavan način prezentira sve potrebne podatke i obavijesti vezane za određeni zadatak.

Naslov zadataka, kratak opis, datum koji govori do kada zadatak treba biti izvršen, status i interaktivni gumbi su dio takve tablice, kao što je i prikazano na Slika 4.4 Tablični prikaz zadataka.


Task list

Task name	Task description	Deadline	Task status	Action
Design sketch	Design sketch for first mee...	30/4/2018	Resolved	No action
Meeting	Meeting with customers	30/4/2018	Open	Resolve
Repo	Create repo for project	20/5/2018	Open	Resolve
Logo	Logo design	10/5/2018	Open	Resolve
TB	Organize first TeamBuilding	15/6/2018	Open	Resolve

< 1 >

Slika 4.4 Tablični prikaz zadataka

Član tima ima mogućnost da, kada smatra da je zadatak izvršen, označi zadatak gotovim što se smatra da je završen i spreman za pregled nadređenih osoba.

Prikaz zadataka je ostvaren uz sljedeći kod:

```
$sql = "SELECT * FROM tasks";
$result = mysqli_query($mysqli, $sql);
```

Kod 4.1 Dohvat zadataka iz baze podataka

U prvoj liniji vidljivo je da se dohvaćaju podaci SQL upitom. U drugoj liniji se nalazi varijabla[6] u koju se spremaju rezultati upita kao takvog. Nakon toga se izvodi petlja[7] koja prolazi kroz dobivene rezultate i ispisuje informacije, u ovome slučaju zadatke. Petlja ima novu varijablu u koju se sprema prva iteracija kroz dobivene rezultate. Varijabla sadrži sve potrebne podatke za ispis. Nakon što završni prva iteracija, započinje druga i tako dalje sve dok postoje rezultati. Također, unutar tablice u koju se ispisuju podaci nalazi se i dio koda koji služi da označavanje pojedinog zadataka kao riješenim, odnosno odrađenim. Kako bi se znalo koji zadatak je označen kao odrađen, svaki zadatak posjeduje svoj jedinstveni, identifikaciji broj. U trenutku kada se odabere tekst predviđen za takvu akciju, poziva se funkcija koja sprema vrijednost u bazu podataka točno za taj zadatak. Slijedi dio koda koji to demonstrira.

[6] Pojam koji se koristi za označavanje promjenjive vrijednosti, odnosno ime kojemu je pridružena vrijednost.
[7] Petlja u vidu programiranju omogućuje da se dijelovi programa izvrše više puta.

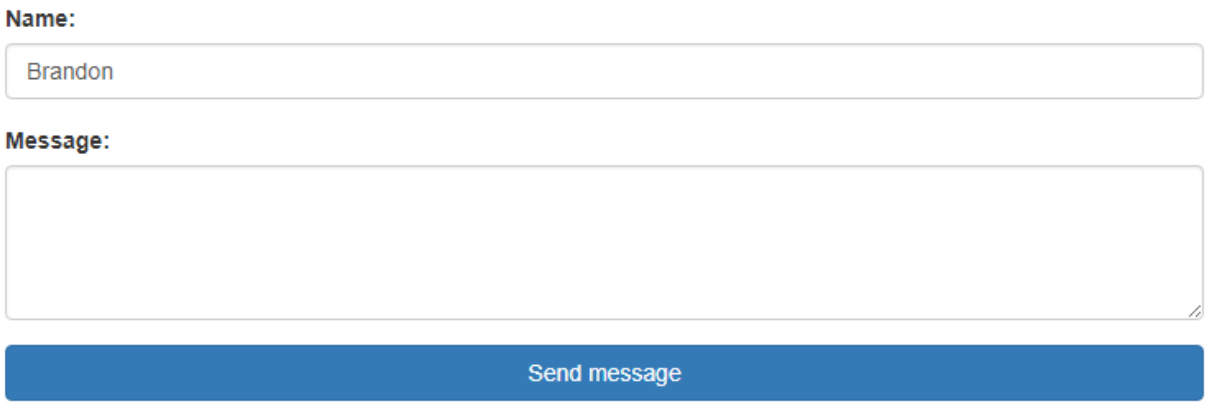
```
<?php
        while ($task = mysqli_fetch_array($result)) {
        ?>
            <tr>
                <td><?= $task['task_name'] ?></td>
                <td><?= $task['task_text'] ?></td>
                <td><?= $task['deadline'] ?></td>
                <?php if($task['status']==="0") { ?>
                    <td>Open</td>
                <?php } else {?>
                    <td>Resolved</td>
                <?php }?>
                <?php if($task['status']==="0") { ?>
                    <td><a href="?taskResolveId=<?php echo
$task['task_id']?>">Resolve</a></td>
                <?php } else {?>
                    <td>No action</td>
                <?php }?>
            </tr>
        <?php
        }
```

Kod 4.2 Petlja za ispis zadataka

4.5. Razgovori

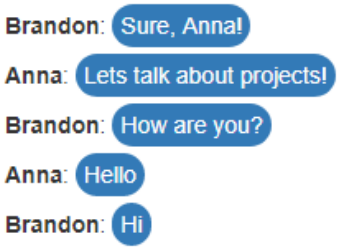
Razgovori su oblik komunikacije kroz samu aplikaciju. Namijenjen je svim registriranim i potvrđenim korisnicima. Svi korisnici i administratori ovdje mogu pregledati i pisati poruke. Prozor se sastoji od dva dijela. Prvi dio je unos poruke, a drugi je pregled poruka.

Poruka se unosi tako da se upiše ime(automatski se unosi u polje predviđeno za to) i sama poruka u prozor predviđen za to.



Slika 4.5 Unos poruke

Pregled poruka se vrši kroz jednostavno sučelje gdje je navedeno ime pošiljaoca i poruka.



Slika 4.6 Pregled poruka

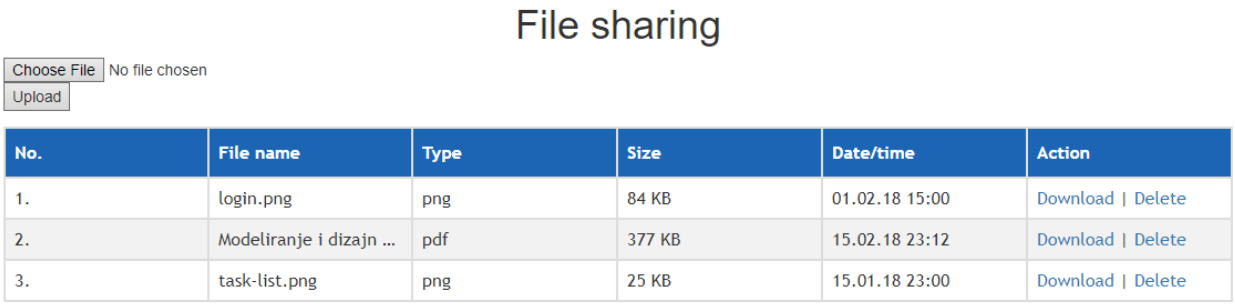
Programski kod za ovaj dio aplikacije nalazi se u ovim datotekama:

1. chat.php
2. insert.php
3. logs.php

Navedene skripte su povezane na način da se u prvoj nalazi programski kod za unos poruke i osvježavanje preglednika koje služi kao provjera postoje li nove poruke. Kada se poruka unese poziva se druga skripta(insert.php) koja zapisuje unesene podatke u bazu podataka. Treća skripta(logs.php) služi za dohvat poruka koje su sortirane od najnovije prema onoj najstarijoj.

4.6. Datoteke

Razmjena datoteka bitna je na svim razinama i u svim timovima. Kako je aplikacija predviđena za manji tim ljudi, svi korisnici imaju pristup svim datoteka.



Slika 4.7 Razmjena datoteka

Spremanje datoteka se vrši na način da se prilikom učitavanja prikuplja nekoliko vrsta podataka poput imena, vrste, veličine. U trenutku pritiska na gumb koji označava dodavanje datoteka, izvršava se sljedeći kod:

```
$name  = $_FILES["file"]["name"];
$type  = $_FILES["file"]["type"];
$error = $_FILES["file"]["error"];
$size  = $_FILES["file"]["size"];
$tmp_name = $_FILES["file"]["tmp_name"];
$new_name = "upload_".time()."." . $ext;
```

Kod 4.3 Prikupljanje podataka o učitanoj datoteci

`$_FILES` je polje(engl. *array*) koje skuplja, to jest sprema informacije o učitanoj datoteci. Zadnje dvije linije koda daju učitanoj datoteci privremeno ime i novo ime koje se sastoji od riječi „upload_“ i vrijeme učitavanja, te na kraju ime ekstenzije, odnosno vrsta datoteke.

Bitno je napomenuti nekoliko stavki kod učitavanja datoteka. Do vrste datoteke dolazi se tako što se ime razdvaja nakon točke, budući da je poznato da se nakon točke nalazi ime nastavka, ekstenzije.

```
$ime = explode(".", $ime_datoteke);
```

Kod 4.4 Razdvajanje imena datoteke od ekstenzije

Informacije se spremaju sljedećim upitom na bazu podataka:

```
$query = "INSERT INTO uploads (path, name_original, name_server, type, ext, size, upload_time) VALUES ('".PATH."', '$name', '$new_name', '$type', '$ext', '$size', NOW())";
```

Kod 4.5 Spremanje podataka o datoteci

Nakon unosa, u tablicu se zapisuju informacije iz Kod 4.3 Unos podataka u tablicu baze podataka i prikazuju.

Brisanje datoteke vrši se na način da svaki redak tablice, odnosno svaka datoteka, ima svoj identifikacijski broj i po tome se točno zna koja datoteka se briše. Kada se izabere akcija za brisanje, vrši se upit na bazu kojim se zapis briše.

```
$query = "DELETE FROM uploads WHERE id = '$id' LIMIT 1";
$result = mysqli_query($mysqli, $query);
```

Kod 4.6 Brisanje datoteke

Preuzimanje datoteka je sličan načinu brisanja. Prema identifikacijskom broju se prepoznaje datoteka koja će biti obrisana. Prvo se provjerava da li je poslan zahtjev za brisanje na sljedeći način:

```
if(isset($_GET["action"]) && $_GET["action"] == "del")
```

Kod 4.7 Provjera zahtjeva za brisanje

Provjerava se ukoliko je zatražena akcija putem gumba i da li je identifikacijski tekst gumba isti onome koji smo postavili za brisanje. Zatim se vrše dva upita, prvi za provjeru postoji li takav zapis i drugi koji vrši upit brisanja.

```
$id = (int)$_GET["ID"];
```

Kod 4.8 Varijabla za spremanje identifikacijskog broj

`$_GET` je super globalna varijabla koja je polje i unutar nje se nalaze vrijednosti elemenata forme(Božajić, 2014.). U ovome slučaju riječ je o identifikacijskom broju datoteke.

```
if($id > 0) {
      $query = "SELECT * FROM uploads WHERE id = '$id'";
      $result = mysqli_query($mysqli, $query);
      if($result) {
```

Kod 4.9 Prikaz svih učitanih datoteka

4.7. Tablice baze podataka

Spremanje podataka je zapravo prebacivanje svih podataka koje korisnik unese u aplikaciji kako bi se kasnije moglo istima pristupiti, upravljati, uređivati i prikazivati. Potrebne tablice su sljedeće:

- Korisnici
- Timovi
- Obavijesti
- Zadaci
- Datoteke
- Razgovori

4.7.1. Korisnici

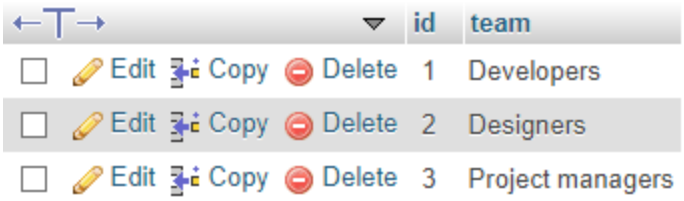
Tablica korisnika sadrži sve podatke koje korisnik unio prilikom registracije i prije navedene dodatne podatke. Kako jedna tablica sa podacima izgleda prikazuje sljedeća slika:

id	username	password	user_type	email	registration_date ▾ 1	hash	confirmed	team
40	Alex	$2y$10$IsQdpJae9uauQnxX8zol9ukH6bMKvheWXIaBez.Nirp...	user	john@domain.com	17-11-2017	a97da629b098b75c294dffdc3e463904	0	Project managers
38	John	$2y$10$eJs2Ue9DkFDCRUms9un52u8mS83wdO6e40ActrTOY4w...	user	joe@domain.com	14-02-2018	217eedd1ba8c592db97d0dbe54c7adfc	1	Developers
41	Anna	$2y$10$o.K8j7sYMLCy46bH7g.a7u88dy0kNfEACo8j0fCAi/W...	user	anna@domain.com	12-12-2017	0ff39bbbf981ac0151d340c9aa40e63e	1	Project managers
39	Joe	$2y$10$L3ArmBDsfvRxvwGi.jwN3eaJmsZT9fh1V.04GBpmBGy...	admin	admin@admin.com	10-02-2018	f2201f5191c4e92cc5af043eebfd0946	1	Project managers

Slika 4.8.1 Tablica korisnika u bazi podataka

4.7.2. Timovi

Popis timova koristi se vrlo često u aplikaciji i ovoj tablici ovisi dosta podataka. Na primjer, obavijesti za grupu korisnika koji su dizajneri neće vidjeti voditelji projekta i obrnuto. Ona se sastoji od ID-a i od samog imena.



	id	team
Edit Copy Delete	1	Developers
Edit Copy Delete	2	Designers
Edit Copy Delete	3	Project managers

Slika 4.9 Tablica timova u bazi podataka

4.7.3. Obavijesti

Brze i kratke obavijest(engl. *notifications*) čine tablicu koja se sastoji od:

- ID
- Naslov obavijesti
- Tekst obavijesti
- Status
- Grupa korisnika

Status nam govori da li je korisnik pročitao obavijest ili nije. Grupa korisnika predstavlja kojoj grupi je namijenjenja obavijesti i ograničava vidljivost same obavijesti.

	comment_id	comment_subject	comment_text	comment_status	team
Edit Copy Delete	1	Notification #1	First meeting, room01	1	Designers
Edit Copy Delete	32	Notification #2	Designers meet #1	0	Project managers

Slika 4.10 Tablica timova u bazi podataka

4.7.4. Zadaci

Ključna tablica za zadatke u koju se spremaju podaci:

- ID
- Naslov zadatka
- Tekst zadatka
- Tko je zadužen za određeni zadatak
- Krajnji datum završetka zadataka
- Status

	task_id	task_name	task_text	assignee	deadline	status
Edit Copy Delete	81	Design sketch	Design sketch for first meeting	Wiiliam	30/4/2018	1
Edit Copy Delete	83	Meeting	Meeting with customers	Anna	30/4/2018	0
Edit Copy Delete	84	Repo	Create repo for project	John	20/5/2018	0
Edit Copy Delete	85	Logo	Logo design	Sharon	10/5/2018	0
Edit Copy Delete	86	TB	Organize first TeamBuilding	Brandon	15/6/2018	0

Slika 4.11 Tablica zadataka u bazi podataka

4.7.5. Datoteke

Jedna od najkompleksnijih tablica iz razloga što sadrži veću količinu zapisa. Oni su redom:

- ID
- Putanja
- Originalno ime
- Privremeno ime
- Vrsta datoteke
- Ekstenzija datoteke
- Veličina datoteke
- Vrijeme unosa

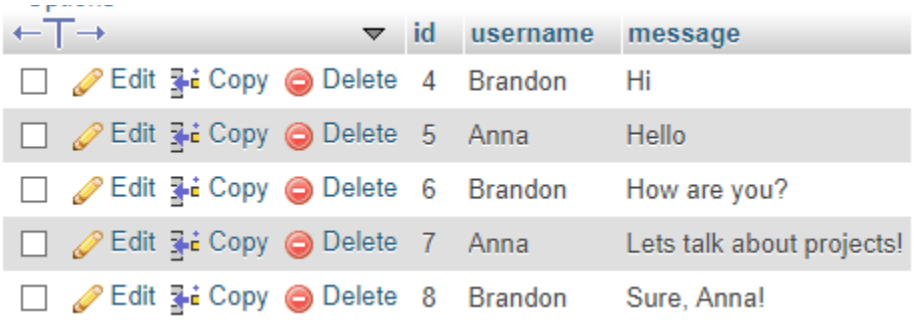
Svi ovi podaci su potrebni kako bi korisnici mogli preuzimati datoteke i snalaziti se između raznih vrsta datoteka. Iz ovih podataka moguće je doznati sve potrebne informacije.

	id	path	name_original	name_server	type	ext	size	upload_time
Edit Copy Delete	25	uploads/	login.png	upload_1518732676.png	image/png	png	85667	2018-02-01 15:00:00
Edit Copy Delete	26	uploads/	task-list.png	upload_1518732686.png	image/png	png	24660	2018-01-15 23:00:00
Edit Copy Delete	28	uploads/	Modeliranje i dizajn baze podataka.pdf	upload_1518732738.pdf	applicatio	pdf	385704	2018-02-15 23:12:18

Slika 4.12 Prikaz datoteka u bazi podataka

4.7.6. Razgovori

Tablica se sastoji od dvije vrste podataka, a to su ime korisnika koji je poslao poruku i njegova poruka. Osim za spremanje poruka, služi i za ispis povijesti razgovora.



	id	username	message
Edit Copy Delete	4	Brandon	Hi
Edit Copy Delete	5	Anna	Hello
Edit Copy Delete	6	Brandon	How are you?
Edit Copy Delete	7	Anna	Lets talk about projects!
Edit Copy Delete	8	Brandon	Sure, Anna!

Slika 4.13 Prikaz razgovora u bazi podataka

4.8. Unos, dohvaćanje, brisanje i uređivanje podataka

Radu s podacima treba pristupiti s oprezom jer zbog jednog krivog klika pokazivačem može doći do velikog gubitka podataka. Ovaj odlomak je posvećen spremanju, prikazivanju, uređivanju i brisanja podataka koristeći se PHP programskim jezikom. MySQL naredbe unutar PHP programskog jezika nam omogućuju sve radnje navedene u naslovu.

4.8.1. Unos podataka

Unos ili spremanje podatka znači pronaći odgovarajuću tablicu i ispuniti ju potrebnim podacima.

```
"INSERT INTO users(username,password,email,registration_date,hash,
user_type,team)"."VALUES('$username','$password','$email',
'$registration_date','$hash','$user_type','$team')";
```

Kod 4.10 Unos podataka u tablicu baze podataka

Iz gore navedenog primjera možemo zaključiti da se podaci spremaju u tablicu imena „users“, zatim se navode imena retka, te njihove pripadajuće vrijednosti.

4.8.2. Dohvaćanje podataka

Za dohvaćanje podataka iz tablica koristimo SELECT upit. Svaki upit na bazu podataka može se ograničiti raznim uvjetima. Primjer dohvaćanja podataka iz tablice „users“

```
"SELECT * FROM users WHERE id=1";
```

Kod 4.11 Dohvaćanje podataka iz tablice baze podataka

Navedeni primjer nam omogućuje da dohvatimo korisnika koji ima ID 1.

4.8.3. Izmjena podataka

Kako bismo izmijenili redak tablice koristimo naredbu UPDATE.

```
"UPDATE users SET user_name='Marko' WHERE id=1";
```

Kod 4.12 Izmjena retka u tablici podataka

4.8.4. Brisanje podataka

Brisanje retka vrši se s naredbe DELETE. Princip je isti kao i kod dohvaćanja, osim naredbe.

4.9. Sesija

Sesija(engl. *session*) je niz HTTP zahtjeva[8] i odgovora između njih. Svaka sesija ima svoj jedinstveni identifikator i na temelju toga se točno saznaje da pojedini zahtjevi pripadaju točno određenoj sesiji. Konkretno u ovome radu sesije su korištene za čuvanje podataka korisnika koji je trenutno prijavljen u sustav, a to je na primjer ime, tim, da li je korisnik prijavljen ili mu je istekla prijava i slično[9]. U programskome jeziku PHP sesije se definiraju na sljedeći način:

```
session_start()

session_destroy()
```

Kod 4.13 Primjer definiranja sesije i poništavanje iste

Kako bismo sesiji dodijelili vrijednost potrebno je napraviti varijablu u koju spremamo željenu vrijednost.

```
$_SESSION['imeSesije'] = vrijednost;
```

Kod 4.14 Dodijeljivanje vrijednosti sesiji

Ovdje je potrebno pripaziti na vrijednosti jer su one dostupne za korištenje samo dok je korisnik prijavljen, odnosno podaci su vidljivi dok sesija traje, onoga trenutka kada se sesija zatvori, uništi, podaci kao takvi nisu dostupni. Varijable sa podacima su različite za svakog korisnika.

[8] HTTP zahtjev je protokol za komunikaciju između poslužitelja i klijenta.
[9] Izvor: http://www.srce.unizg.hr/files/srce/docs/edu/osnovni-tecajevi/d350_polaznik.pdf

4.10. Sigurnosni aspekt

Sigurnosni aspekt konkretno u ovome slučaju je zapravo autentikacija korisnika, provjera identiteta, potvrđivanje svojeg identiteta. U aplikaciji se korisnik identificira korisničkim imenom i lozinkom. Kako bi se sačuvala privatnost i sigurnost podataka, to jest lozinke, ona se ne zapisuje u obliku kako je i unesena. Kada korisnik unese svoju lozinka ona se zapravo sprema u kombinaciji slova, brojeva i znakova. U radu je korištena PHP funckija password_hash().

```
$password = $mysqli->real_escape_string(password_hash($_POST['password'], PASSWORD_BCRYPT));
```

Slika 4.14 Spremanje lozinke u bazu podataka

`$_POST` je super globalna varijabla koja je polje i unutar nje se nalaze sve vrijednosti elemenata forme koje su poslane metodom POST(Božajić, 2014.). Vrijednosti se šalju na način da ih korisnik ne može vidjeti u tome trenutku, što također doprinosi sigurnosti. U navedenome primjeru radi se o zaporci(engl. *password*).

`PASSWORD_BYCRIPT` govori koji algoritam[10] se koristi. U ovome slučaju koristiti će se CRYPT_BLOWFISH što znači da će zaporka biti spremljena koristeći "$2y$" znakove[11].

Drugi slučaj kada je u pitanju sigurnost aplikacije je svakako SQL ubrizgavanje(engl. *injections*). SQL ubrizgavanje koda se može dogoditi na onim mjestima gdje se nalazi unos podataka, na primjer unos imena ili email adrese. Ukoliko se u polje za unos podataka unese SQL komanda može se ugroziti sigurnost baze podataka i općenito podataka. Ovaj slučaj se sprječava tako što koristimo sljedeću PHP funkciju:

```
mysqli_real_escape_string()
```

Kod 4.15 PHP funkcija za izbjegavanje SQL ubrizgavanja

Navedenom funkcijom izbjegavamo posebne znakove za upotrebu u SQL izrazu i neće se dopustiti izvršavanje opasnih komandi.

[10] Algoritam u programiranju je uputa kako riješiti određeni problem ili zadatak.
[11] Izvor: http://php.net/manual/en/function.password-hash.php

Zaključak

Tijekom rada na projektu u kojemu sudjeluje više od jedne osobe potreban je sustav za komunikaciju, praćenje i kontroliranje iste. Kako bi se komunikacija uspješno i pravovremeno odvijala potreban je određeni sustav.

U ovome radu postiglo se da se na jednostavan način može komunicirati sa svim osobama koje su uključene projekt, pratiti zadatke, obavijesti i razmjenu datoteka. Aplikacija je namijenjena malom timu ljudi.

Za rad na projektu u koji je uključeno više od deset osoba aplikacija bi postala nepregledna i teško bi se administrirala. Ukoliko je riječ upravo o većem broju ljudi, potrebno je potražiti rješenje koje omogućuje naprednije razdvajanje raznih timova, razne kategorije datoteka i prilagođeno administratorsko sučelje. Pod prilagođenim se smatra da postoji pretraga korisnika, zadataka i obavijesti, pregled svih obavijesti i uređivanje obavijesti kao takvih, naprednija kontrola nad datoteka i razgovorima.

Aplikaciju je u budućnosti moguće unaprijediti jer struktura sustava to omogućava. Problem koji se ovdje pojavljuje je da li je zaista potrebno krenuti putem unaprjeđivanja aplikacije čije performanse dopuštaju rad manjem timu. Nastavak na mogući problem je da ako se aplikacija unaprijedi u nekoliko segmenta, ona može postati nepotrebno komplicirana za manji tim ljudi.

Aplikacija kakva je sada ima prostora za napredovanje i da se ne kreće u smjeru unaprjeđivanja. Prostor za usavršavanje leži u nekoliko segmenata, a oni su redom: prikaz zadataka, razgovori i administratorsko sučelje. Prikaz zadataka, preciznije tekst zadataka, je problematičan kada je tekst zadataka, odnosno opis, duži i neće se vidjeti cijeli tekst. Ovdje je moguće ugraditi opciju prikazivanja cijelog teksta u zasebnome prozoru. Razgovori bi se mogli proširiti na privatne i opće, gdje bi korisnici mogli međusobno komunicirati umjesto da se poruke šalju svima. Administratorsko sučelje trebalo bi unaprijediti tako da postoji prikaz svih informacija i uređivanje istih.

Tijekom izrade rada naučio sam da je planiranje pola puta do ostvarivanja cilja jer ukoliko se pogriješi prilikom planiranja može značiti da smo planirali odmah neuspjeh.

Popis kratica

HTML	*HyperText Markup Language*	Vrsta programskog jezika
CSS	*Cascading Style Sheets*	Vrsta programskog jezika
SASS	*Syntactically Awsome Style Sheets*	Vrsta programskog jezika
SCSS	*Sassy Cascading Style Sheets*	Vrsta programskog jezika
AJAX	*Asynchronous JavaScript And XMLHttpRequest*	Asinkroni JavaScript i XMLHttp zahtjev
CSV	*Comma Separated Values*	Zarezom odvojene vrijednosti
PDF	*Portable Document Format*	Prijenosni format dokumenta

Popis slika

Slika 2.1 Primjer HTML strukture 2

Slika 2.2 Izgled CSS koda 3

Slika 2.3 SASS primjer ugnježđivanja 4

Slika 3.1 Pet elemenata korisničkog iskustva (planiranje) 7

Slika 3.2 Prikaz obavijesti unutar aplikacije 8

Slika 4.1 Struktura direktorija 12

Slika 4.2 Registracija 13

Slika 4.3 Prijava u sustav 14

Slika 4.4 Tablični prikaz zadataka 15

Slika 4.5 Unos poruke 17

Slika 4.6 Pregled poruka 17

Slika 4.7 Razmjena datoteka 18

Slika 4.8.1 Tablica korisnika u bazi podataka 20

Slika 4.9 Tablica timova u bazi podataka 21

Slika 4.10 Tablica timova u bazi podataka 21

Slika 4.11 Tablica zadataka u bazi podataka 22

Slika 4.12 Prikaz datoteka u bazi podataka 22

Slika 4.13 Prikaz razgovora u bazi podataka 23

Slika 4.14 Spremanje lozinke u bazu podataka 25

Popis kôdova

Kod 2.1 Primjer korištenja PHP-a unutar HTML koda 4

Kod 2.2 Primjer korištenja jQuery-a 5

Kod 4.1 Dohvat zadataka iz baze podataka 15

Kod 4.2 Petlja za ispis zadataka 16

Kod 4.3 Prikupljanje podataka o učitanoj datoteci 18

Kod 4.4 Razdvajanje imena datoteke od ekstenzije 18

Kod 4.5 Spremanje podataka o datoteci 19

Kod 4.6 Brisanje datoteke 19

Kod 4.7 Provjera zahtjeva za brisanje 19

Kod 4.8 Varijabla za spremanje identifikacijskog broj 19

Kod 4.9 Prikaz svih učitanih datoteka 19

Kod 4.10 Unos podataka u tablicu baze podataka 23

Kod 4.11 Dohvaćanje podataka iz tablice baze podataka 23

Kod 4.12 Izmjena retka u tablici podataka 24

Kod 4.13 Primjer definiranja sesije i poništavanje iste 24

Kod 4.14 Dodijeljivanje vrijednosti sesiji 24

Kod 4.15 PHP funkcija za izbjegavanje SQL ubrizgavanja 25

Literatura

[1] BOŽAJIĆ I. PHP PROGRAMIRANJE: PRIRUČNIK, ZAGREB, ALGEBRA, 2014.

[2] http://www.srce.unizg.hr/files/srce/docs/edu/osnovni-tecajevi/d350_polaznik.pdf, SIJEČANJ 2018.

[3] http://php.net/manual/en/function.password-hash.php, PROSINAC 2017.

[4] https://medium.com/omarelgabrys-blog/ux-a-quick-glance-about-the-5-elements-of-user-experience-part-2-a0da8798cd52, SIJEČANJ 2018.

[5] http://www.wpbeginner.com/glossary/apache/, PROSINAC 2017.