Dinamičke web aplikacije

Jajac, Lovre

SVEUČILIŠTE U SPLITU

POMORSKI FAKULTET

Lovre Jajac

Dinamičke web aplikacije

ZAVRŠNI RAD

Split, 2017.

SVEUČILIŠTE U SPLITU

POMORSKI FAKULTET

Pomorske elektrotehničke i informatičke tehnologije

Dinamičke web aplikacije

ZAVRŠNI RAD

MENTOR:
dr. sc. Anita Gudelj

STUDENT:
Lovre Jajac
(0171259023)

Split, 2017.

SAŽETAK

Korisnici interneta diljem svijeta svakodnevno se susreću sa dinamičkim web aplikacijama i dinamičkim web stranicama. Takav tip web sadržaja, zbog svog bogatog sadržaja, jednostavnog i mobilnog pristupa, danas je sve popularniji, a njegova popularnost će sve više rasti.

U ovom radu opisani su principi na kojima dinamičke web stranice funkcioniraju, njihova strukturalna podjela i arhitektura. Objašnjen je razvoj web aplikacija na strani klijenta i na strani poslužitelja te programski jezici i tehnike koje se koriste prilikom razvoja web aplikacija. Za potrebe ovog rada i za primjerni prikaz funkcioniranja web aplikacija izrađena je jednostavna web aplikacija za izračun duljine ugovora pomoraca.

Ključne riječi: *dinamička web aplikacija, dinamička web stranica, razvoj web aplikacija*

ABSTRACT

Internet users around the world are faced with dynamic web applications and pages on a daily basis. This type of web content is becoming increasingly popular, because of its rich content and simple and mobile access.

This thesis describes the principels on which dynamic web sites work, their structural division and architecture. Also, client side and server side web development and programming languages and techniques used for development of web applications are explained. For the purposes of this thesis and for example of developing web application a simple web application for calculating seaman's contract length has been made.

Keywords: *Dynamic web application, dynamic web site, development of web applications*

SADRŽAJ

1. UVOD

Početkom 90-ih godina prošlog stoljeća dolazi do pojave World Wide Web-a (skraćeno WWW ili samo web). U početku su web stranice bile statičke, pisane isključivo HTML (eng. *Hypertext Markup Language*) kodom, nije se obraćala pozornost na izgled stranice prema korisniku, već samo na lakše snalaženje i bolju preglednost informacija. Popularizacijom weba, došlo je do potrebe prilagođavanja web stranica tržištu, pojavljivanja raznih dinamičkih sadržaja na stranicama (razne animacije, flash sadržaji, vremenske prognoze) u svrhu postizanja ugodnijeg i lijepšeg prikaza stranice, ali i boljeg dopiranja do krajnjeg korisnika u komercijalne svrhe. Danas su popularnije dinamičke web stranice, koje podrazumijeva podatke na web stranici koji se mijenjaju za vrijeme korištenja web stranice. Dinamičke web stranice se razvijaju različitim web tehnologijama uz korištenje programskih jezika (npr. PHP, JavaScript, ASP.NET), dok je kostur same stranice izveden u HTML jeziku. Statičke stranice više nisu zanimljive korisniku, neki korisnici ih čak smatraju i neprimjerenima u današnjici pa one polako odlaze u prošlost.

World Wide Web (skraćeno WWW ili Web) je jedan od servisa na internetu stvoren za razmjenu informacija u HTML formatu. Dokumenti mogu sadržavati tekst, slike i drugi multimedijalni sadržaj.

HTML je temelj svake Web stranice, to je opisni jezik kojim se opisuju svi elementi i sav sadržaj prikazan na nekoj web stranici, temeljna zadaća HTML-a je uputiti preglednik kako da ispiše sadržaj neke stranice, a pri opisu koristi se oznakama (eng. *Tag)* koje su unaprijed definirane za sve elemente. Pomoću „slaganja“ *tagova* u jednu cjelinu dobiva se smislena web stranica.

Evolucijom tehnologije, Interneta i WWW-a, ali i samom promjenom načina života, zadnjih nekoliko godina sve popularnije postaju i web aplikacije. Web aplikacije su aplikacije kojima se pristupa putem web-a pomoću internet mreže. Korisnik više nije ograničen na pristup aplikaciji samo preko lokalnog računala na kojemu je ona instalirana, već sada to može učiniti sa bilo kojeg mjesta na svijetu koje ima pristup internetu. Sve web aplikacije se u potpunosti pokreću i izvršavaju na poslužitelju, a korisnik im pristupa pomoću internet preglednika. U prilog velikoj popularnosti Web aplikacija ide i to što su Web preglednici, pomoću kojih se

pristupa aplikacijama, implementirani u svim računalima, ali i mobitelima, tabletima, pa čak i televizijama. Također njihove prednosti su što web aplikacije ne zauzimaju prostor na korisnikovom čvrstom disku, s obzirom da se cijeli sadržaj web aplikacije nalazi na poslužitelju, nije potrebno izvršavati nikakve procedure nadogradnje aplikacije jer se aplikacija nadograđuje na poslužitelju, jednostavne su za korištenje i bitno uštеđuju vrijeme korisnika ili tvrtki koji ih koriste, jer nikakve instalacije prije izvršetka aplikacije nisu potrebne, dovoljno joj je samo pristupiti pomoću internet preglednika.

Neki od nedostataka web aplikacija su ovisnost o brzini klijentove internetske veze, dostupnost poslužitelja na koji je aplikacija postavljena, nekompatibilnost određenog internet preglednika sa elementima korištenim u web aplikaciji, mogućnost korisnika da u internet pregledniku sam postavi veličinu fonta i opcije prikaza sa čime bi se narušila dosljednost prikaza aplikacije te najveća opasnost na internetu – sigurnost mreže (opasnost od neželjenih upada, virusa i sl.).

Primjena web aplikacija vrlo je široka, gotovo sve postojeće aplikacije već imaju realiziranu svoju inačicu kao web aplikacije, a ako nemaju, zbog velikog broja prednosti koje web aplikacije nude, u bližoj budućnosti će zasigurno imati.
Najčešće web aplikacije sa kojima se korisnici susreću su servisi elektroničke pošte, servisi za pregledavanje video sadržaja, alati za uređivanje multimedije, čitanje tekstualnih dokumenata, internet bankarstvo i mnogi drugi.

Prilikom korištenja web aplikacija mogu se definirati dva osnovna pojma WWW-a bez kojih web aplikacije ne mogu funkconirati:

- Poslužitelj (server) – računalo ili software koje je zaduženo za slanje i primanje podataka od mnogobrojnih klijenata. Na poslužitelju je pohranjena web aplikacija i svi podaci koji su potrebni za izvođenje aplikacije.
- Klijent – krajnji korisnik web aplikacije.

2. DINAMIČKE WEB STRANICE

U ranim početcima razvoja interneta, u velikoj većini slučajeva korištene su statičke web stranice. Statička web stranica može se još opisati kao skup povezanih HTML datoteka s predodređenim, fiksno definiranim sadržajem koji ostaje nepromijenjen neovisno o korisnikovim zahtjevima prilikom pregleda. Kod statičkog okruženje ne postoji nikakav oblik korisničkog sučelja za upravljanje web sadržajem. Za ažuriranje i uređivanje statičke web stranice potrebne su stručne vještine, a samim time što su sadržaj i kod isprepleteni u datotekama, klijent bez stručnog znanja nema mogućnost samostalnog mijenjanja sadržaja [7].

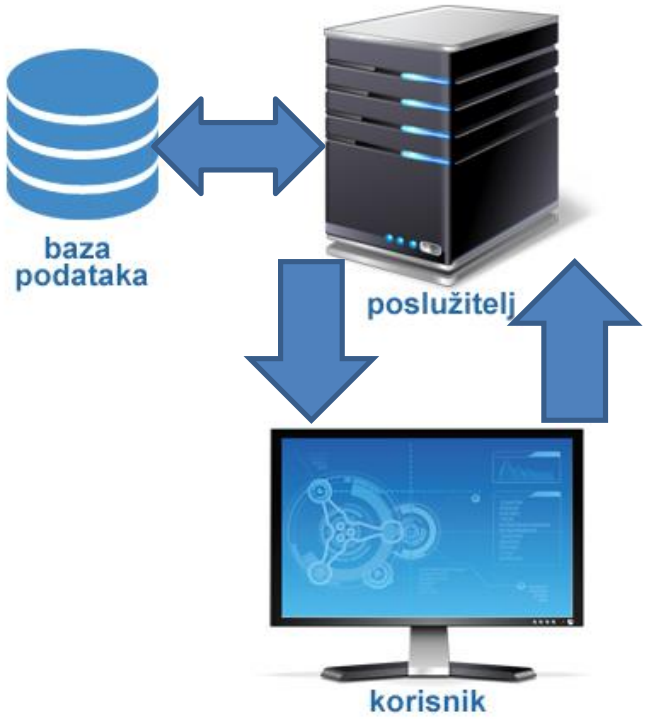
U posljednje vrijeme statičke web stranice koje su pisane samo u HTML-u sve više se mijenjanu u dinamičke. Dinamičke web stranice su puno zanimljivije krajnjem korisniku, oku ugodnije za rad i, prateći napredak tehnologije, daleko naprednije i modernije u odnosu na statičke web stranice. Nazivaju se dinamičkim prvenstveno zbog njihove naravi dinamičkog mijenjanja sadržaja koji se prikazuje krajnjem korisniku. To mogu biti funkcije provjere preglednika (npr. Google Chrome, Mozilla Firefox, IE) i vrste uređaja (osobno računalo, tablet, mobitel) sa kojeg korisnik pristupa određenoj web stranici u svrhu boljeg prikaza web stranice, određivanje lokacije korisnika putem njegove IP adrese u svrhu automatske promjene jezika stranice ili prikaza vremenske prognoze za grad iz kojeg korisnik pristupa web stranici, te bezbroj drugih mogućnosti.

Dinamičke web stranice prikazuju sadržaj „u hodu“ odnosno prilikom svakog pristupa nekoj web lokaciji generira se sadržaj koji se u internet pregledniku prikazuje krajnjem korisniku. Sastoje se od kostura web stranice izvedenog u HTML-u, baze podataka i skripti izvedenih u nekom od programskih jezika, za razliku od statičkih web stranica, gdje se na poslužitelju nalazi samo HTML i CSS dokumenti koji se prilikom pristupa stranici uvijek jednako prikazuju.

Najpoznatije programske tehnologije za skriptiranje na strani poslužitelja su PHP, Java, ASP i mnoge druge. Najčešće tehnologije za dizajniranje i rad sa bazama podataka su MySQL i XML.

2.1. PRINCIP RADA DINAMIČKIH WEB STRANICA

Korisnik na svojem računalu pokreće Internetski pretraživač (klijent) kako bi poslao upit određenoj web lokaciji, klijent se povezuje na poslužitelja i šalje korisnikov upit. Zatim, poslužitelj zaprima korisnikov upit i obrađuje ga. Nakon obrade poslužitelj dohvaća potrebne podatke te ih obrađene šalje natrag do klijenta koji prezentira podatke krajnjem korisniku na njegovom računalu. Ova radnja se po potrebi može ponavljati nebrojeno puta. Pojednostavljen princip rada dinamičkih web stranica prikazan je shematski na slici 1.



Slika 1. Princip rada dinamičke web stranice

2.2. PROCES DOHVAĆANJA WEB STRANICE

Proces korisnikovog dohvaćanja web stranice sa poslužitelja, te njenog prikaza u korisnikovom internet pretraživaču je slijedeći:

- Podaci, kostur stranice i slike su odvojeno pohranjene na poslužitelju
- Poslužitelj čeka zahtjev klijenta

- Klijent učitava Web stranicu sa svim njenim dinamičkim komponentama, kao što su:
    - HTML kostur
    - slike
    - podaci
    - i ostale komponente
- Klijent ispunjava upit u obliku:
    - pritiska na tipku izbornika
    - popunjavanja tekstualne forme
    - ili neki drugi interaktivni zahtjev
- Klijent šalje upit i zahtjev poslužitelju
- Poslužitelj procesuira klijentov zahtjev, obrađuje ga i slaže HTML kostur, podatke i slike prema korisnikovom zahtjevu
- Internetski preglednik procesuira HTML tagove i prikazuje ih klijentu.

2.3. KLIJENT-POSLUŽITELJ ARHITEKTURA

Prema osnovnoj strukturi mrežne aplikacije dijele se na dvije skupine: peer to peer (P2P) i klijent-poslužitelj.

Velika većina web aplikacija temelji se na klijent-poslužitelj arhitekturi. Kao što je već navedeno sastoje se od klijentskog i poslužiteljskog dijela. Najjednostavnije rečeno, klijent postavlja zahtjev, traži neku uslugu, a poslužitelj taj zahtjev ispunjava. U ovakvoj arhitekturi klijent uvijek prvi inicira komunikaciju sa poslužiteljem. Klijentski dio aplikacije nalazi se u korisnikovom uređaju (osobno računalo, mobitel, tablet...), najčešće u obliku Internet pretraživača. Poslužiteljski dio web aplikacije obično radi na većim i snažnijim računalima, koji su stalno spojeni na mrežu i uvijek dostupni. Za razliku od poslužitelja koji je uvijek aktivan, spojen na mreži i čeka klijentovu komunikaciju, klijenti ne moraju uvijek biti aktivni, korisnik svoj uređaj može i isključiti ili se odspojiti sa Interneta.

3. RAZVOJ WEB APLIKACIJA

Razvoj web aplikacija, ali i web stranica općenito, može se podijeliti na dvije osnovne cjeline:

- programiranje na strani poslužitelja (eng. Back-end development) – razvoj svih komponenti potrebnih za rad aplikacije na poslužitelju.
- programiranje na strani klijenta (eng. Front-end development) – razvoj svih komponenti potrebnih za prikaz aplikacije krajnjem korisniku tijekom korištenja aplikacije.

S obzirom na složenost programskih tehnika potrebnih za razvoj navedenih cjelina web aplikacije, osobe zadužene za razvoj web aplikacija (eng. Developeri) dijele se na Back-end i Front-end developere (slika 2).

Slika 2. Podjela razvoja web aplikacija na dvije osnovne cjeline [11]

3.1. FRONT-END RAZVOJ

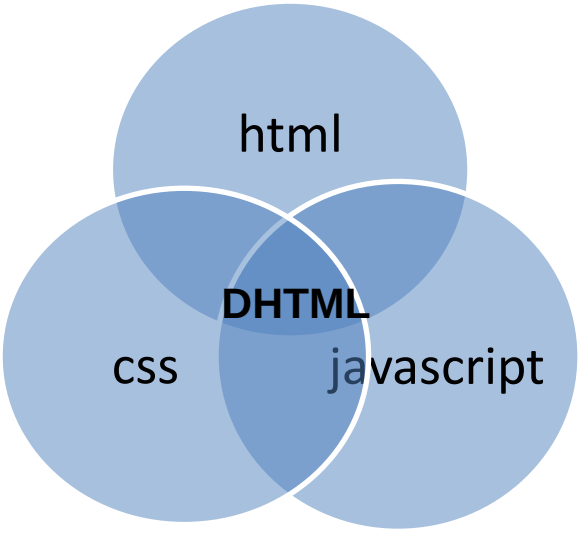
Pojam Front-end obuhvaća sve ono što krajnji korisnik vidi pri korištenju neke web aplikacije, uključujući i dizajn same stranice te omogućava korisniku da „komunicira“ sa stranicom. Front-end razvoj može se nazvati i skriptiranjem na strani klijenta, jer se sav sadržaj, koji obuhvaća front-end, pokreće i odvija u

korisnikovom internet pregledniku. Skripte koje se izvršavaju na strani klijenta najčešće se nalaze ugrađene u HTML dokumentima, ali također mogu se nalaziti i u odvojenoj datoteci te pozivati iz HTML dokumenta. Prilikom korisnikovog zahtjeva, određene datoteke se šalju sa poslužitelja na kojem se nalaze na korisnikovo računalo. Skripta se zatim izvršava u korisnikovom internet pregledniku i prikazuje se dokument koji sadrži vidljive rezultate izvršavanja skripte.

Primani cilj kojem treba težiti prilikom razvoja weba na strani klijenta je omogućiti korisniku da na stranici lako uoči sadržaj koji traži te da je stranica čitljiva i ugodna za rad. U današnjici, ovo je malo teže postići s obzirom da korisnici sadržaju mogu pristupati sa različitih uređaja koji imaju različite dimenzije i rezolucije ekrana pa je samim time zadaća front-end developera da omogući normalan prikaz sadržaja na svim rezolucijama i uređajima. Web stranice i web aplikacije moraju imati sposobnost prikaza na svim uređajima (eng. *Cross-device*), svim operativnim sustavima (eng. *Cross-platform*) i svim internet preglednicima (eng. *Cross-browser*).

Često se miješaju pojmovi Front-end developera i web dizajnera, no zapravo oni imaju skroz različite uloge u izradi web sadržaja, web dizajneri ne trebaju poznavati niti jednu razvojnu tehnologiju i programski jezik, oni su tu da samo osmisle kako će neka Web stranica ili aplikacija izgledati, te izrade koncept stranice i svu potrebnu grafiku u programima za obradu grafike. Tada nastupaju front-end developeri koji grafički dizajn stranice rade funkcionalnim u svakom pogledu komunikacije sa korisnikom i prikaza stranice te na kraju sam taj prednji dio stranice spajaju sa back-end aplikacijom koja se pokreće.

Najčešće tehnologije koje služe za front-end razvoj su HTML, CSS i JavaScript, koji prilikom korištenja skupa čine Dinamički HTML sadržaj – DHTML (slika 3).



Slika 3. JavaScript kombiniran sa HTML-om i CSS-om čini Dinamički HTML (DHTML) [6]

3.1.1. HTML

HTML *(eng. Hypertext Markup Language)* jezik temelj je svake Web stranice. HTML nije programski jezik, već opisni, jer njime se samo opisuje izgled i raspored sadržaja na Web stranici.

Temeljna zadaća HTML-a je uputiti Web preglednik kako da prikaže hipertext dokument, a da pri tome taj dokument uvijek bude jednak, bez obzira na kojem se uređaju, operacijskom sustavu ili pretraživaču on otvara. HTML datoteke su zapravo obične tekstualne datoteke, ekstenzija im je .html ili .htm. Osnovni dijelovi svakog dokumenta su tagovi, koji nam opisuju kako će se nešto prikazivati u Web pregledniku. Također, jedan od najvažnijih dijelova HTML elemenata su linkovi, oni nam omogućuju povezivanje Web stranice sa podstranicama ili drugim Web stranicama i tako čine kompaktnu cjelinu. Svi tagovi označeni su početnim znakom „<“ i završnim znakom „>“. Svaki tag može imati i svoje atribute, kojima se dodatno definira kako se određeni elementi prikazuje. Tri osnovna taga čine HTML stranicu, glavni tag je <html> tag, unutar kojeg se nalaze dva taga, <head> kao zaglavlje i <body> u kojem se nalazi sadržaj stranice. Unutar zaglavlja, odnosno <head> taga, stavljaju se podaci koji se ne prikazuju direktno na stranici, ali ju također opisuju i pomažu pri njenom prikazu. Tu se najčešće dodaje <title> tag, koji nam omogućuje prikazivanje naziva stranice u Web pregledniku, ali i razna stilska obilježja stranice, bila ona direktno ugrađena ili u obliku poveznice na

CSS dokument. Unutar <body> taga nalazi se kompletan sadržaj stranice, svi tagovi koji se tamo upišu biti će prikazani u pregledniku i utjecati će na prikaz dokumenta. Na slici 4 prikazan je primjer HTML koda koji u zaglavlju definira naziv stranice i ispisuje neki tekst na samoj stranici.

```
<html>
<head>
<title>Naziv stranice </title>
</head>
<body>


Tekst koji će se prikazati na stranici.


</body>
</html>
```

Slika 4. Primjer HTML koda

Posljednja verzija Hypertext Markup Language opisnog jezika je HTML5. Prijašnja verzija HTML-a bila je HTML 4.01. i izašla je davne 1999. Godine. U međuvremenu internet je doživio velike promjene, te je bilo potrebno razviti ovaj opisni jezik kako bi zadovoljio i najveće potrebe korisnika, ali i programera. HTML5 je razvijen da podržava sav bogati sadržaj na stranici bez potrebe za drugim dodacima za pregledavanje sadržaja. HTML5 obuhvaća apsolutno sve, od animacija, grafike, filmova pa čak i do razvoja samih Web aplikacija.

Cilj stvaranja HTML-a 5 je razvoj standarda koji može pokretati cijele aplikacije unutar Web preglednika i sadrži široki skup novih tehnologija. Uključuje nove HTML tehnologije, geolokacijska programska sučelja, drag&drop programska sučelja, podršku za audio i video, JavaScript 2.0, CSS3 itd. Također, HTML5 nudi mogućnost ostvarivanja načina rada na Web aplikacijama bez trenutne povezanosti s mrežom (eng. Offline mode). Svi poznati Internet preglednici podržavaju nove HTML5 elemente i programska sučelja za izradu programa.

HTML5 sve više dobiva na popularnosti, tako polako bacajući Flash sadržaj u prošlost. Svoj najveći trijumf HTML5 doživio je kada je YouTube u potpunosti odbacio Adobeov Flash i prešao na HTML5 kao njihovu platformu za gledanje video sadržaja.

3.1.2. CSS

CSS (eng. Cascading Syle Sheets) jezik je formatiranja koji služi za stilsko oblikovanje hipertekstualnih dokumenata. Koristi se za opisivanje izgleda i oblikovanje HTML elemenata dodavanjem raznih vizualnih svojstava prilagođavajući izgled stranice koja se prikazuje krajnjem korisniku. CSS je napravljen od strane W3C organizacije u svrhu izbjegavanja prekomjernog nakupljanja HTML tagova unutar dokumenta, kako bi se HTML više orijentirao na sadržaj stranice, dok CSS brine o njegovom izgledu.

CSS je primarno zamišljen kako bi omogućio odvajanje dokumenta sadržaja i dokumenta prezentacije web stranice, koji uključuje i kontrolu rasporeda na stranici te kontrolu prikaza slova i boja web stranice. Ovakva podjela omogućava jednostavniji i fleksibilniji pristup sadržaju kao i lakšu kontrolu nad određenom grupom HTML elemenata. Sintaksa CSS jezika je krajnje jednostavna i koristi određeni broj engleskih riječi kako bi definirala pojedine elemente. Svaki opis nekog elementa se sastoji od poziva tog elementa, njegovog svojstva i dodane vrijednosti. CSS je moguće koristiti i direktno unutar HTML dokumenta, no takva praksa se ne koristi. Zbog preglednosti sadržaja i lakšeg snalaženja u kodu CSS dokumenti spremaju se u zasebnu datoteku koja uvijek ima ekstenziju .css. Ove datoteke povezuju se sa HTML datotekom na koju se odnose pomoću koda prikazanog na slici 5 koji se nalazi u zaglavlju HTML dokumenta.

```
<link href="naziv-css-dokumenta.css" rel="stylesheet">
```

Slika 5. Primjer povezivanja CSS datoteke sa HTML kodom stranice

CSS3 je najnovija verzija CSS tehnologije, koja se javlja pojavom HTML-a 5, te omogućava niz novih mogućnosti za vizualno uređenje Web stranica. Jedna od novih mogućnosti je također smanjenje veličine koda koji se trebao napisati da bi se postigle neke manje vizualne promjene na stranici. CSS3 je u potpunosti izbacio Flash tehnologiju koja je prije bila potrebna za izradu animacija, te je sada pojednostavnio i olakšao izradu animacija u samom CSS kodu. Neke od novih mogućnosti koje su se pojavile sa CSS3 verzijom su: 2D i 3D animacije, tekstualni efekti, animacije, pozadine, obrubi, itd.

3.1.3. JavaScript

JavaScript je skriptni programski jezik koji se izvršava u internet pregledniku na strani korisnika. Klijentsko skriptiranje u potpunosti se izvršava u korisnikovom web pregledniku, nebitno da li su stranice smještene na poslužitelju u ASP, PHP ili nekoj drugoj tehnologiji, i upravo u tome leži popularnost JavaScripta. JavaScript se javlja 1995. godine u doba dok nije bilo razvijenih dinamičkih web stranica i poslužiteljskog programiranja. Jedini izbor za razbijanje monotonije statičkih web stranica i unošenje malo dinamičkog sadržaja bilo je korištenje JavaScript programskog jezika. Naziva se skriptnim jezikom jer se sastoji od serije komandi koje se izvršavaju direktno u interpreteru, bez prethodne potrebe za kompajliranjem koda, već se komande direktno čitaju iz samog koda. Zbog ovakve karakteristike JavaScript je jezik koji se izvršava na strani klijenta, odnosno u Internet pregledniku na kojem je pokrenuta JavaScript skripta.

Sam HTML kod podržava funkciju da korisnik, nakon unesenih podataka (npr. u neku web formu za unos podataka), pošalje podatke natrag na poslužitelja te se provjerava da li je unos podataka ispravan i rezultat se šalje natrag korisniku, ako unos podataka nije ispravan cijeli proces mora se obaviti iz početka. Upravo zbog ovakvih razloga nastao je JavaScript, koji ovakvu provjeru podataka može obaviti na klijentovoj strani u web pregledniku, i tako uštedjeti dosta vremena, olakšati posao poslužitelju i smanjiti propusnost mreže.

Neke od poznatijih funkcija JavaScripta sa kojima se korisnici susreću svakodnevno su: skočni izbornici, padajući izbornici, zabrana desnog klika mišem na stranici, detekcija preglednika kojeg korisnik koristi, web kolačiči, provjera sadržaja prilikom unosa podataka, itd.

3.2. BACK-END RAZVOJ

Pojam back-end razvoja obuhvaća razvoj svih komponenti potrebnih za rad Web aplikacije na poslužitelju. Back-end web aplikacije može se nazvati i mozgom cijele aplikacije. Back-end razvoj je dio razvoja web aplikacije koji nikada nije vidiljiv korisniku. Pozadinski dio web aplikacije sastoji se od poslužitelja, baze podataka i programskog koda. Razvoj programskog koda aplikacije, ali i njegovo povezivanje sa bazom podataka odvija se korištenjem tzv. poslužiteljskih programskih jezika (eng. Server-side scripting).

Glavni zadatak programiranja na strani poslužitelja je osigurati da aplikacija ispravno radi, da svi procesi koji se događaju u pozadini aplikacije ispravno funkcioniraju i samu aplikaciju povezati sa korisničkim sučeljem razvijenim od strane front-end developera. Također zadaće back-end programera su razvoj samog koda aplikacije, unaprijeđivanja aplikacije i stvaranja njihove nadogradnje, rad sa bazama podataka, stabilizacija aplikacija i otklanjanje poteškoća prilikom izvršavanja aplikacija.

Najčešće korišteni programski jezici za programiranje na strani poslužitelja su PHP, Java, Python, Java, ASP, ASP.NET i ColdFusion, pomoću kojih programeri razvijaju aplikacije, ali i povezuju aplikacije sa bazama podataka koje su najčešće izvedene u SQL tehnologiji, od kojih je najpoznatija MySQL platforma za rad sa bazama podataka.

3.2.1. PHP

PHP je jedan od najnaprednijih i najkorištenijih programskih jezika za programiranje na strani poslužitelja današnjice, koji se temelji na C i Perl sintaksi, prvenstveno namjenjen programiranju i razvoju dinamičkih Web stranica, ali je svoju uporabu našao i kao *stand alone* programski jezik.

PHP se izvršava na poslužitelju i omogućava sve standardne poslužiteljske akcije – komunikaciju s bazom podataka, slanje e-pošte, čitanje i pisanje u datoteke na poslužitelju, manipuliranja grafikom te rada sa XML-om. Velika prednost PHP-a je njegova podrška za kumunikaciju sa širokom paletom baza podataka. Podržava sve popularnije baze podataka kao što su MySQL, dBase, Oracle, itd...

Također prednost PHP-a je što može biti ugrađen u HTML kod, bez potrebe za odvajanjem u dvije različite datoteke. PHP tako svaku statičku HTML web stranicu može pretvoriti u dinamičku. PHP kod se jednostavno implementira u HTML dokument upisivanjem <? i ?> tagova. Korištenjem tih tagova u HTML kodu govorimo poslužitelju da se između njih nalazi PHP kod te da je prvo potrebno njega izvršiti i tek nakon toga korisniku poslati dobiveni sadržaj i ostatak statičkog HTML koda (slika 6).

```
<html>
<head>
<title>Primjer PHP implementacije</title>
</head>
<body>

<?
echo "Dobrodosli na web stranicu";
?>

</body>
</html>
```

Slika 6. Primjer implementacije PHP-a u HTML dokument

3.2.2. Java

Java je jedan od najpopularnijih programskih jezika na svijetu. Ovaj objektno-orijentirani programski jezik napravljen je da se u njemu može programirati gotovo sve. No, Java nije samo programski jezik, može se slobodno reći da je Java skup programskih alata, poznatih pod nazivom *The Java Platform*. Čine ga razne knjižice (libraries), framework-ovi, Aplikacijsko programsko sučelje (eng. API – application programming interface), Java dodaci, Java Virtual Machine i Java Runtime Enviroment.

"Write once, run anywhere" – Javin je glavni moto. To bi značilo da kod možemo napisati samo jednom i kasnije ga pokretati na svim vrstama uređaja. Upravo činjenica da je java cross-platform čini je najpopularnijim programskim jezikom na svijetu. Pokretanje na više uređaja omogućeno je pomoću Java bytecode tehnologije. Program pisan u Javi se ne kompajlira na razini računala, u strojnom jeziku kao ostali programski jezici, već se generira međukod (eng. Bytecode). Upravo zbog toga potrebni su nam već gore navedeni Java Virtual Machine i Java Runtime Environment dodaci.

Java se ponajviše bazira na sintaksi programskih jezika C i C++, ali je jednostavnija i lakša za korištenje u odnosu na navedena dva programska jezika. Svoju najveću popularnost dobila je pojavom Android uređaja, jer je upravo Java temeljni jezik svih Android operativnih sustava.

3.2.3. Python

Python je programski jezik nastao 1990. godine, a zanimljivost je da je svoje ime dobio po popularnoj TV seriji Monthy Python. Upravo sa tim imenom programskog jezika, njegov osnivač Nizozemac Guido van Rossum želi poručiti da proramiranje može biti zabavno.

Python je vrlo jednostavan i pregledan, ali istodobno i snažan programski jezik. Podržava objektno-orijentirani, struktruni i aspektni stil programiranja. Baziran je na C-u ali sa vrlo preglednijom i jednostavnijom sintaksom. Kod Pythona upotreba vitičastih zagrada, točki-zareza i sličnih simbola svedena je na minimum. Na primjer, za definiranje bloka naredbi nije potrebno koristiti vitičaste zagrade i tako ih odvajati od ostatka koda, već se jednostavno blokovi naredbi razlikuju jednostavnim uvlačenjem koda.

Python kod sprema se u jednostavnu tekstualnu datoteku sa nastavkom „.py“. Kao i kod Jave, prilikom izvođenja programa, program se kompajlira u međukod (bytecode) i tek nakon toga program se prevodi na razinu računala na kojem se taj kod izvodi.

3.2.4. ASP

ASP je Microsoftov programsi jezik za razvoj na strani poslužitelja. ASP je pokrata za Active Server Pages i predstavlja Microsoftovo rješenje za izradu dinamičkog web sadržaja. Baziran je na JScript i VBScript programskim jezicima. ASP kod se implementira u HTML kod, slično kao i već spomenuti PHP. Korisnik koji je pristupio nekoj Web stranici ni neće znati da je na njoj ASP skriptra, već će nakon izvršenja skripte dobiti normalnu HTML Web stranicu koju je generirala skripta. Jedino po čemu korisnik može zaključiti da je dobio generiranu ASP skriptu jest ekstenzija web stranice koja će završavati na nastavak .asp umjesto .html.

ASP je zastarjela, i prilično limitirana metoda generiranja dinamičkog web sadržaja pa je Microsoft razvio novu, napredniju platformu nazvanu ASP.NET

3.2.5. ASP.NET

ASP.NET javlja se 2002. godine kao Microsoftov nasljednik ASP-a. Napravljen je da programerima olakša izradu dinamičkih Web stranica, Web aplikacija i drugih Web servisa. ASP.NET je .NET bazirano okruženje koje

programerima omogućuje razvijanje aplikacija u bilo kojem .NET podržanom jeziku, uključujući VB.NET, C# i JScript.NET.

ASP.NET dolazi u tri različita razvojna okruženja koja su potpuno nezavisna jedno od drugog. Svako od njih cilja na drugačiju publiku ili vrstu primjene, a odabir može ovisiti i o razini znanja programera i vrsti aplikacije koja će se razvijati. Postoje slijedeće tri opcije u ponudi:

- ASP.NET Web Forms
- ASP.NET MVC
- ASP.NET Web Pages.

3.2.6. ColdFusion

ColdFusion je komercijalno okruženje za razvoj Web aplikacija. Koristi CFML programski jezik. Osnovna namjena mu je bila omogućiti povezivanje HTML stranice sa bazom podataka, ali ubrzo nakon toga postaje stand-alone platforma za razvijanje Web aplikacija sa svojim razvojnim okruženjem i programskim jezikom. Ono što ga posebno izdvaja iznad drugih platformi za razvoj je to što ima svoj vlasititi programski jezik – CFML (Cold Fusion Markup Language). To je programski jezik koji po namjeni i mogućnostima podsjeća na PHP i ASP jezike, dok po korištenju tagova podsjeća na HTML jezik. ColdFusion se najčešće koristi u razvoju stranica pokrenutih podacima i intranet mrežama. Također, može upravljati i razmjenom tekstualnih poruka i SMS-ovima.

3.2.7. MySQL

MySQL je najpopularniji sustav za upravljanje bazama podataka. Najčešće se koristi u kombinaciji sa PHP programskim jezikom. MySQL baze podataka su relacijskog tipa, što se pokazalo kao najbolji način skladištenja i pretraživanja velikih količina podataka.

Iako se mogu koristiti na svim operacijskim sustavima i u raznim vrstama aplikacija, MySQL se ipak najviše koristi kao baza podataka za Web aplikacije. MySQL baze podataka nalaze se na poslužitelju i podržava pristup za više korisnika na istu bazu podataka. Na poslužitelju može biti više nezavisnih baza podataka, a unutar pojedinog projekta mogu se koristiti podaci iz različitih baza podataka. Svakom korisničkom računu na poslužitelju mogu se dodijeliti različita administracijska prava, koja mogu obuhvaćati određene baze podataka ili čak cijeli

poslužitelj. Administracijska prava određuju pravo pristupa bazama podataka, pravo na stvaranje novih baza podataka, pravo izmjena podataka i sl.

MySQL baze centralna su komponenta paketa programa otvorenog koda za razvoj Web aplikacija pod nazivom LAMP. LAMP je platforma za razvoj Web sadržaja koju čine **L**inux kao operacijski sustav, **A**pache kao Web server, **M**ySQL kao relacijski sustav za upravljanje bazama podataka i **P**HP (nekada to mogu biti Python ili Perl) kao objekto orjentirani programski jezik.

Upravljanje MySQL bazama podataka vrši se SQL (*eng. Structured Querry Language*) programskim jezikom. To je strukturalni jezik za upravljanje bazama podataka koje se izvršava preko upita (eng. Query). Upiti služe da se bazi pošalje određeni zahtjev ili naredba. Pomoću PHP-a povezuje se sa bazom podataka, izvršavaju se SQL upiti i upravlja se sa svim podacima u bazi podataka. Za pristup bazi podataka potrebno je unijeti korisničko ime i lozinku. Osnovna naredba za pristup bazi podataka pomoću PHP-a glasi *mysql_connect (server, korisnicko_ime, lozinka);* gdje *server* označava adresu poslužitelja na kojem se nalazi baza podataka, ako se baza nalazi na istom poslužitelju na kojemu se izvrašava ova skripta tada nije potrebno unijeti ime poslužitelja. *Korisnicko_ime* i *lozinka* označavaju polja gdje se unosi korisnikovo korisničko ime i lozinka potrebna za pristup bazi podataka. Primjer povezivanja na bazu podataka pomoću PHP-a prikazan je na slici 7.

```
<?php
$spajanje = mysql_connect("brod","pomorac1","123");
if (!$konekcija) // Provjera da li je tvrdnja
$spajanje tocna
{
        die('Neuspjelo spajanje: ' .
mysql_error());
}
?>
```

Slika 7. Primjer spajanja na MySQL bazu podataka pomoću PHP programskog jezika

3.3. APLIKACIJSKO PROGRAMSKO SUČELJE

Aplikacijsko programsko sučelje (eng. *Application Programming Interface – API)* je skup rutina, protokola i alata za razvoj aplikacija i software-a. Aplikacijska programska sučelja sadrže već definirane bibilioteke, dijelove programa i definiraju načine komunikacije između dva različita software-a. Osnovna namjena API-ja je povezati dvije različite aplikacije i omogućiti njihovo komuniciranje i razmjenu podataka, najjednostavniji primjer je obična copy-paste funkcija koja omogućuje kopiranje teksta (ili slika i drugog sadržaja) iz jednog programa i njihovo korištenje u drugom programu (npr. slika nacrtana u Paintu može se kopirati i zalijepiti u Word). Povezivanje aplikacija omogučeno je pomoću dijelova programskog koda jednog programa te određene funkcije koju on izvršava i korištenja toga koda u drugom programu. Takvu komunikaciju između dva programa i njihovo integriranje omogućavaju nam razni API-ji. U praksi API najčešće dolazi u obliku biblioteke koja sadrži specifikacije za funkcije, strukture podataka, klase objekata i varijable.

3.3.1. Web API

Kada se API-ji koriste u svrhu razvoja web stranica i web aplikacija, obično definiraju Hypertext Transfer Protocol (HTTP) poruke zahtjeva i strukturu poruke odgovora. Koriste se i kao već gotove biblioteke prilikom front-end JavaScript programiranja na strani korisnika kako bi programeru skratile vrijeme programiranja, ali i biblioteke u nekom od back-end programskih jezika za razvoj same aplikacije. Također, API-ji definiraju i pravila komunikacije između back-end i front-end djelova aplikacije.

Danas su API-ji sve popularniji kod programera web aplikacija jer im znatno skraćuju vrijeme potrebno za razvoj određenih funkcija programa ali i olakšavaju posao jer nije potrebno pisati duge linije koda za neku funkciju. Neke od najpopularnijih API implementacija koda, koje omogućuju velike internetske kompanije, prilikom razvoja web aplikacija su:

- Facebook login i komentiranje – jedna od načešćih stvari koja se javlja u današnjici, opcija koja omogućava prijavu na web stranice i ostale web aplikacije pomoću Facebook računa, bitno skraćuje vrijeme i resurse potrebne za registrirati se na određenu stranicu kako bi se pristupilo njenom

potpunom sadržaju. Također, sve popularnije je i komentiranje članaka nekih web sadržaja pomoću svog Facebook računa.

- Google maps – najčešće se koristi na web stranicama kada je potrebno označiti vlastitu lokaciju na mapi, prikazati upute kako pronaći neko mjesto, ali i web aplikacijama koje se baziraju na prikaz određenih restorana, ili drugih korisniku zanimljivih sadržaja u blizini, radnog vremena određenih objekata prikazanih na mapi i njihovih ocjena usluga koje pružaju.
- Ostali sadržaji – vremenske prognoze koje se mogu javljati u aplikacijama ili na web sadržajima koje nam omogućuju Yahoo Weather ili neki drugi meteorološki servis, razne mogućnosti postavljanja Youtube videa na web stranice, korištenje Dropbox servisa koji se često implementira u web aplikacije kako bi korisnici mogli sačuvati određene dokumente, umjesto da developeri sami razvijaju vlasitit prostor na poslužitelju za spremanje sadržaja zanimljivih nekom korisniku.

4. IZRADA DINAMIČKE WEB STRANICE ZA IZRAČUN DULJINE TRAJANJA UGOVORA POMORACA

Za potrebe završnog rada izrađena je jednostavna web stranica koja dinamički obrađuje podatke nakon korisnikovog unosa podataka te ispisuje numerički rezultat i grafički rezultat u obliku grafikona. Aplikacija je osmišljena za lakše računanje vremena trajanja ugovora kod pomoraca te se dinamički dio aplikacije u potpunosti izvodi na klijentovoj strani mreže, odnosno u samom internet pregledniku klijenta korištenjem JavaScript skriptiranja.

Princip rada aplikacije je vrlo jednostavan, klijent pristupa stranici za izračun duljine trajanja ugovora, u ponuđena polja za unos potrebno je unijeti datum početka trajanja ugovora te datum isteka ugovora pomorca. Pomoću unesenih parametara, i parametra današnjeg datuma, aplikacija sama radi izračun ukupnog broja dana koji moraju proteći između dva navedena datuma, te računa koliko dana je prošlo od početnog dana ugovora do današnjeg dana te u obliku grafikona prikazuje postotak odrađenog dijela ugovora.

Prilikom izrade dinamičke web stranice korištene su tri osnovne tehnologije koje čine DHTML (Dinamički HTML) – HTML, CSS i JavaScript. Svi kodovi pisani su u besplatnom programu namjenjenom pisanju raznih skripti i kodova „Notepad++“.

4.1. MOZAK APLIKACIJE (JAVASCRIPT)

Pomoću JavaScripta te njegovih već gotovih programskih biblioteka i nekih već gotovih skripti otvorenog koda *(Eng. Open-source)* dostupnih na webu izrađene su skripte koje čine glavni dio ove web aplikacije – skripte koje obavljaju matematičke operacije prilikom izračuna broja dana između unesenih datuma i izračun dana koji su prošli od početnog datuma do današnjeg datuma, ali i grafički prikaz rezultata u obliku animirajućeg grafikona i padajući izbornici za unos datuma.

Za izradu svih JavaScript elemenata na stranici korištena je najpoznatija programska bibilioteka otvorenog koda za JavaScript pod nazivom jQuery. jQuery je biblioteka već gotovih programskih skripti i rješenja, vodi se pod motom *„Do more, write less“* što bi u prijevodu značilo - „postigni više, piši manje“. Može se

reči da je jQuery biblioteka olakšala pisanje koda milijunima JavaScript programera. Biblioteka je dostupna na internetu, te se prije samog izvršenja koda ona dohvaća jednostavnom HTML naredbom, zatim se u JavaScript skripti dohvaćaju određeni već gotovi elementi iz biblioteke koji su potrebni za rad skripte koja se programira.

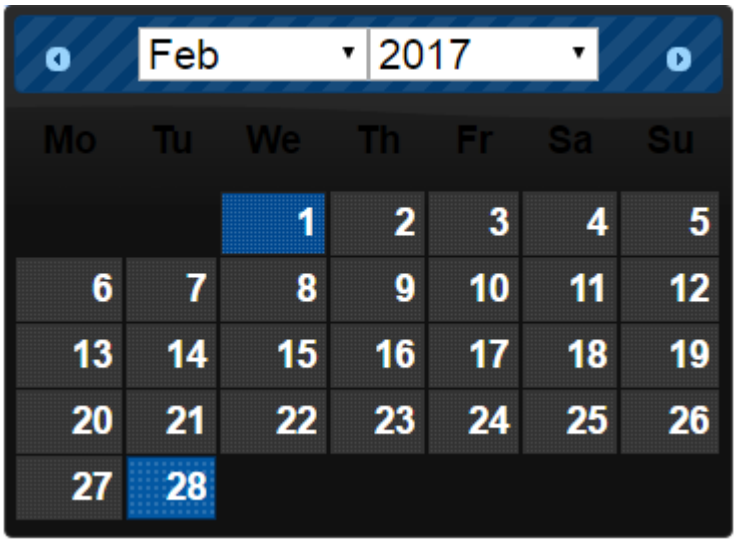
4.1.1. Unos datuma

Prvo je potrebno omogućiti klijentu unos datuma početka i kraja ugovora, što je obavljeno jednostavnim dohvaćanjem *Datepicker* elementa iz jQuery biblioteke. *Datepicker* je naziv alata koji omogućuje jednostavniji rad sa datumima, jedan je od poznatijih elemenata JavaScripta sa kojim se korisnici gotovo svakodnevno susreću na webu. Na slici 8 prikazan je JavaScript kod za unos datuma te pozivanje *Datepicker* elementa iz jQuery biblioteke.

```
1   $(document).ready(function() {
2
3
4
5   $( "#startdate,#enddate").datepicker({
6   changeMonth: true,
7   changeYear: true,
8   firstDay: 1,
9   dateFormat: 'dd/mm/yy',
10  })
11
12  $( "#startdate" ).datepicker({ dateFormat: 'dd-mm-yy' });
13  $( "#enddate" ).datepicker({ dateFormat: 'dd-mm-yy' });
```

Slika 8. JavaScript kod za unos datuma pomoću *Datepicker* funkcije

Poziva se *Datepicker* funkcija iz jQuery biblioteke, definira se format datuma i mogućnost odabira mjeseca i godine na kalendaru, te prvi dan u tjednu na kalendaru, u ovom slučaju broj 1 označava ponedjeljak. Na slici 9 prikazan je izgled *datepicker* elementa nakon što je definiran njegov prikaz. Također, definirane su i klase elementa i njihova imena. Klase nam služe za povezivanje elemenata JavaScripta sa elementima HTML koda, u ovom slučaju to su *#startdate* i *#enddate* koji će se kasnije povezati sa HTML poljima za unos podataka, odnosno datuma početka ugovora i završnog datuma trajanja ugovora.



Slika 9. Izgled JavaScript *Datepicker* elementa za unos datuma

4.1.2. Operacije računanja broja dana između navedenih datuma

Nakon unesenih datuma početka i kraja ugovora, potrebno je izračunati broj dana između dva navedena datuma, te koliko je dana prošlo od ukupnog broja dana. Navedene varijable računaju se pomoću varijabli unesenih datuma te varijable današnjeg datuma kojeg skripta automatski dohvaća iz operativnog sustava. JavaScript kod kojim su obavljene računske operacije računanja broja dana i postotka dana koji su prošli, dohvaćanje današnjeg datuma te krajnji tekstualni i grafički prikaz rezultata prikazan je na slici 10.

Definiraju se varijable *start* i *end* koje predstavljaju unese datume početka i kraja ugovora pomoću *datepicker* alata. Varijabla *danas* predstavlja današnji datum, ona se definira pomoću osnovne JavaScript funkcije *Date* koja služi za dohvaćanje trenutnog datuma i vremena. Važno je napomenuti kako funkcija *Date* datum zapisuje u obliku milisekundi koje su prošle od 00:00 sati 1. siječnja 1970. Godine. U ovom slučaju za potrebe računanja dana među datumima, odnosno oduzimanja dvije vrijednosti datuma zapisanih u milisekundama bilo je potrebno prilikom određivanja varijable današnjeg datuma unaprijed postaviti vrijednosti sati, minuta, sekunda i milisekunda proteklih u današnjem danu na nultu vrijednost.

```
$('#enddate').change(function() {

var start = $('#startdate').datepicker('getDate');
var end   = $('#enddate').datepicker('getDate');
var danas = new Date ();
danas.setHours(0,0,0,0)

if (start<end && start<danas && end>danas) {

var dodanas = Math.round (danas - start)/1000/60/60/24;
$('#doDanas').text('Od kojih je prošlo ' + dodanas +' dana.');
var days   =  Math.round ((end - start)/86400000);
$('#days').text('Ukupno trajanje ugovora: ' + days +' dana');
var postotak = Math.round ((dodanas / days)*100);
$('#postotak').text(postotak + '%');

//opcije grafičkog prikaza postotka
    var options = {
        height: "250px",
                width: "250px",
        line_width: 12,
        color: "#5795ff",
        starting_position: 0,
        percent: 0,
        text: "percent" }

    var progress_circle = $("#progress-circle").gmpc(options);
    progress_circle.gmpc('animate', postotak, 3000);
}
else {
alert ("Pogrešan unos datuma!");
$('#startdate').val("");
$('#enddate').val("");
$('#days').val(""); }
}); //end change function
}); //end ready
```

Slika 10. JavaScript kod za izračunavanja broja dana i ispis rezultata

Kada su definirane sve potrebne vrijednosti skripta klasičnom *if* funkcijom provjerava ispravnost unesenih datuma, ako su datumi pogrešno unešeni, vraća sve vrijednosti unesenih datuma na nultu vrijednost i ispisuje poruka upozorenja na pogrešan unos. Kad je sve ispravno uneseno nastavlja se na računske operacije. Računskim operacijama dobivaju se slijedeće varijable:

- Varijabla *dodanas* – predstavlja broj dana koji su prošli od prvog dana do današnjeg dana, kao što se može vidjeti iz gore prikazanog programskog koda, nakon oduzimanja varijabli *danas* i *start* bilo je potrebno podijeliti dobivenu vrijednost sa 86400000 (broj milisekundi u jednom danu) kako bi dobivena vrijednost bila prikazana u danima.

- Varijabla *days* – predstavlja ukupan broj dana od prvog dana trajanja ugovora do zadnjeg dana trajanja ugovora. Računa se po istom principu kao i prethodno opisana varijabla.
- Varijabla *postotak* – predstavlja omjer dana koji su prošli i ukupan broj dana zapisan u obliku postotka. Jednostavnije rečeno, prikazuje postotak koliko dana ugovora je odrađeno.
- Nakon svake varijable definirane su njihove klase kako bi se kasnije mogle pozvati i ispisati u HTML-u.

Za grafički prikaz odrađenog postotka u obliku kružnice sa animacijom punjenja korištena je već gotova skripta otvorenog koda za koju je bilo potrebno samo promijeniti postavke veličine i boje kružnice, te unijeti do koje vrijednosti će se kružnica „puniti“, u ovom slučaju tu vrijednost predstavlja varijabla *postotak*.

4.2. KOSTUR WEB STRANICE (HTML)

Pomoću HTML-a, odnosno HTML tagova izrađuju se sve web stranice. HTML je opisni jezik pomoću kojeg se definira sadržaj web stranica te način na koji je taj sadržaj prikazan na web stranici. JavaScript skriptu potrebno je implementirati u HTML kako bi korisniku bila omogućena interakcija sa skriptom, tj. Unos podataka, prikaz dobivenih rezultata i sl. Na slici 11 prikazan je kompletan HTML kod web stranice na kojem je smještenja web aplikacija za izračun trajanja ugovora kod pomoraca.

HTML kod sastoji se od dva osnovna dijela, <*head*> i <*body*> elemenata. U zaglavlju dokumenta, odnosno u <*head*> elementu definiraju se skripte kreirane u JavaScriptu, stilski izgled stranice koji se nalazi u CSS dokumentu, kao i sam naziv stranice te jezični formati. U navedenom primjeru u zaglavlju dokumenta pomoću <*script*> oznake pozivaju se sve JavaScript skripte potrebne za pravilno prezentiranje web stranice, u konkretnom slučaju to su jQuery biblioteke, skripta za grafički prikaz postotka u obliku kružnice te gore opisana JavaScript skripta za izračun duljine ugovora pomoraca. Pomoću oznake <style> pozivaju se CSS dokumenti kojima se definira stilski prikaz web stranice.

```
<html>
<head>
 <meta charset="utf-8">
 <title> Trajanje ugovora pomoraca </title>
<script src="https://code.jquery.com/jquery-1.8.3.js"></script>
<script src="https://code.jquery.com/ui/1.10.0/jquery-ui.js"></script>
<script src="./scripts/grasp_mobile_progress_circle-1.0.0.js"></script>
<script src="./scripts/script.js"></script>
<link rel="stylesheet" href="http://code.jquery.com/ui/1.12.1/themes/dot-luv/jquery-ui.css">
<link rel="stylesheet" type="text/css" href="./styles/style.css">
</head>
<body>
<div id="kontenjer">
<br>
  <div id="block"></div>
  <div id="text">
  Danas je: <div id=datum>
  <script> document.write(new Date().toLocaleDateString()); </script>
  </div> <br><br>
  <table width=150px>
<tr>
<td>
Početak ugovora </td><td>Kraj ugovora </td></tr>
<tr><td>
<input size="10" type="text" placeholder="DD/MM/GGGG" id="startdate"> </td><td>

<input size="10" type="text" placeholder="DD/MM/GGGG" id="enddate"></td></tr></table>
<br> <div id="days"> </div> <br>
<div id="doDanas"> </div>
<div class="kruznica" id="progress-circle"></div>
</div>
</div>


</div>
</body>
</html>
```

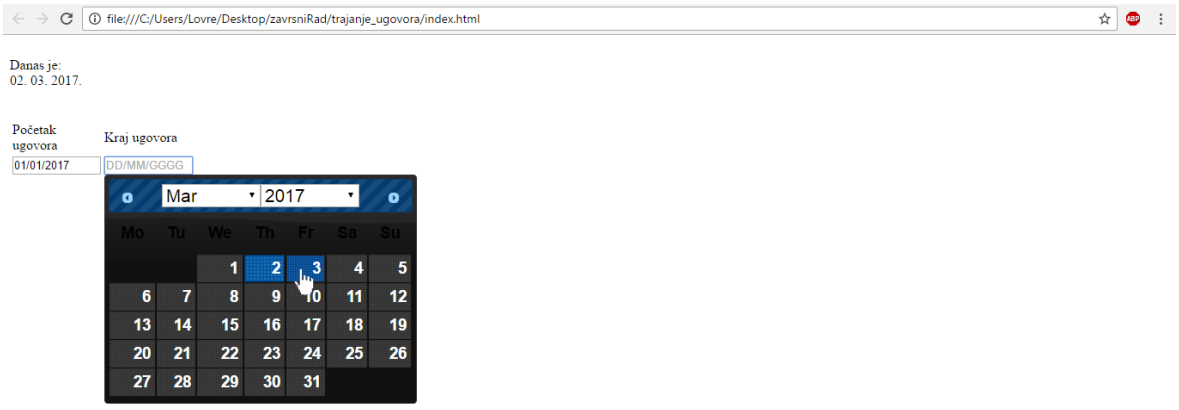
Slika 11. HTML kod web stranice na kojoj je smještena web aplikacija za izračun trajanja ugovora kod pomoraca

U oznaci <*body*> nalaze se svi elementi i sadržaj web stranice koji se prikazuju korisniku. Pomoću <*div*> oznake definira se logička cjelina sadržaja na web stranici, u konkretnom primjeru sav sadržaj stavljen je u jednu logičku cjelinu pod nazivom kontenjer, te se unutar te cjeline sadržaj dijeli na više manjih cjelina.

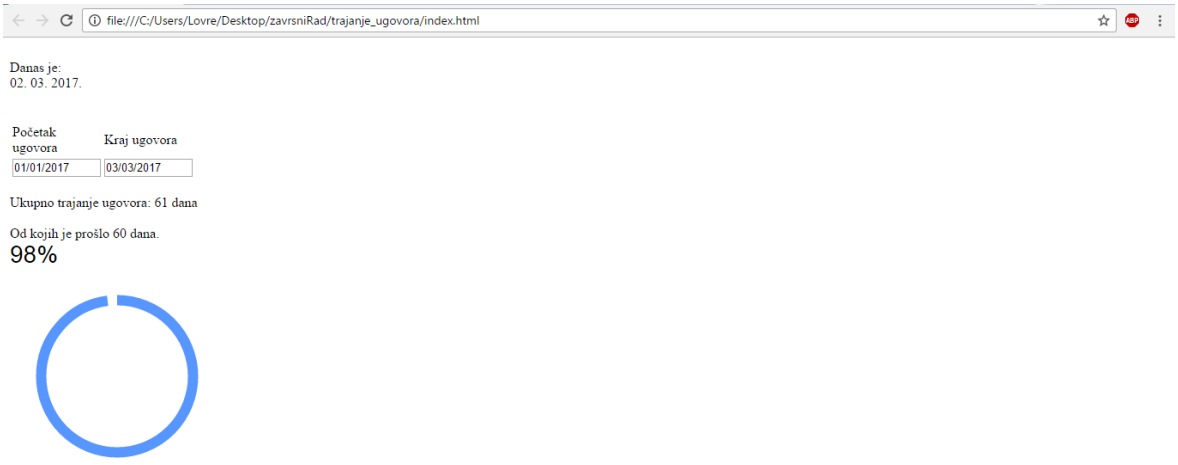
U prvom retku sadržaja, zbog informativnog sadržaja prema korisniku, ispisuje se današnji datum. Nakon ispisa današnjeg datuma, na stranicu su postavljena dva elementa za unos podataka upotrebom HTML oznake <*input*>. Svakom od dva navedena dva polja za unos podataka jednostavnom opisnom oznakom elementa pod nazivom *id="startdate"* odnosno *id="enddate"* pridružena je već prije definirana klasa iz JavaScript skripte, u ovom slučaju to će biti *Datepicker* alat za unos datuma. Također, svakom polju za unos dodan je i njegov

tekstualni opis, a sve to je raspoređeno upotrebom <*table*> oznake, odnosno tablice izrađene u HTML-u.

Nakon što su polja za unos funkcionalna i korisnik može unijeti određene datume skripta će obaviti sve računske operacije, potrebno je ispisati njihove rezultate. Ispis rezultata prilično je jednostavno riješen, pomoću <*div*> oznake stvaraju se logičke cjeline kojima se pomoću opisnih oznaka *id=“ “* dodaju prije definirane klase varijabli iz JavaScript skripte. Tako će se u konkretnom slučaju, svaka u svojoj <*div*> logičkoj cjelini, ispisati klase *days*, *doDanas* i grafički prikaz postotka u obliku kružnice *progress-circle*. Sada, kada je korisniku omogućen unos svih podataka potrebnih za rad JavaScript skripte i prikaz rezultata dobivenih nakon izvršenja skripte, web stranica je u potpunosti funkcionalna i spremna za interakciju sa korisnikom (slike 12 i 13).



Slika 12. Izgled web aplikacije prilikom unosa podataka nakon što je HTML skriptiranje završeno



Slika 13. Izgled web aplikacije prilikom ispisa rezultata nakon što je HTML skriptiranje završeno

4.3. STILSKA PRILAGODBA STRANICE (CSS)

Nakon što je web stranica u potpunosti funkcionalna, potrebno je promijeniti njen stilski izgled, postići lijepši izgled stranice za ugodniji rad krajnjem korisniku. Za stilske promjene na web stranicama koristi se stilski opisni jezik CSS (eng. *Cascading Style Sheets*). CSS-om se opisuje svaki element HTML-a te mu se određuje način na koji će biti prikazan. U zasebnom dokumentu poziva se svaki element HTML-a te mu se unutar vitičastih zagrada dodjeljuju svojstva prikaza.

Na slici 14 prikazan je kod CSS dokumenta u kojem je definiran stilski izgled stranice na kojoj je smještena web aplikacija.

```
body {
    background-image: url('background.jpg');
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-position: center;
    background-size: cover;
    color: fff;
    font-family:
    "Century Gothic",CenturyGothic,AppleGothic,sans-serif;
    font-size: 14px;}

input[type="text"] {
    width=100%;
  padding: 10px;
  border: solid 1px #00000;
  transition: box-shadow 0.3s, border 0.3s;}

input[type="text"].focus {
  border: solid 1px #707070;
  box-shadow: 0 0 5px 1px #969696;}

table{
width: 100%;
font-family:
"Century Gothic",CenturyGothic,AppleGothic,sans-serif;
    font-size: 13px;}
#datum {
float: right;
text-align: right;}

#kontenjer {
float: right;
margin-right: 10%;
margin-top: 5%;
text-align: left;
  position: relative;
  width: 20%;
```

```
#block {
  background: #000;
  filter: alpha(opacity=40);
  /* IE */
  -moz-opacity: 0.4;
  /* Mozilla */
  opacity: 0.4;
  /* CSS3 */
  position: absolute;
  top: 0;
  left: 0;
  height: 100%;
  width: 100%;
  -webkit-box-shadow: 10px 10px 5px 0px rgba(0,0,0,0.75);
-moz-box-shadow: 10px 10px 5px 0px rgba(0,0,0,0.75);
box-shadow: 10px 10px 5px 0px rgba(0,0,0,0.75);
border-radius: 10px 10px 10px 10px;
-moz-border-radius: 10px 10px 10px 10px;
-webkit-border-radius: 10px 10px 10px 10px;}

#text {
  position: absolute;
  top: 2.5%;
  left: 2.5%;
  width: 95%;
  height: 95%;}

#days, #doDanas, #datum
    { font-weight: bold;
    color: #f0f7ff;}

div.ui-datepicker, .ui-datepicker td{
 font-size:12px;
}
```

Slika 14. CSS kod korišten za stilsko uređivanje web stranice

Unutrar HTML elementa body definiraju se osnovna obilježja stranice kao što su pozadina, boja fonta te vrsta fonta koja se koristi. Slika pozadine određuje se unutar *background-image* oznake, te se poziva pomoću *url('/adresa_slike/slika.jpg')* naredbe, budući da je u konkretnom slučaju slika pozadine smještena u istoj mapi kao i CSS datoteka, pod adresom slike dovoljno je upisati samo naziv slike koja se koristi kao pozadina. Kad je slika postavljena kao pozadina, potrebno je definirati njena svojstva, poziciju na stranici, veličinu, ponavljanje prilikom *scrollanja*, itd. Nakon slike pozadine, oznakama *color, font-*

size i *font-style* definirani su boja, veličina i vrsta fonta koji se koristi na web stranici.

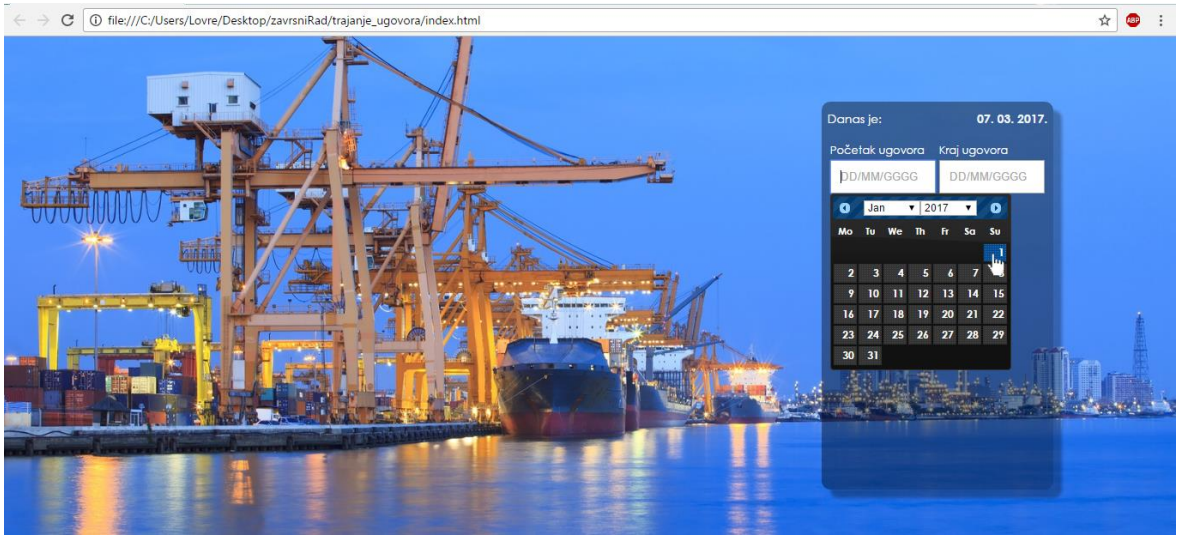
Unutar *input* oznake definira se stilski izgled polja za unos datuma, njihov obrub, animacija tranzicije prilikom označavanja određenog polja klikanjem miša na polje te izgled onog polja koje je trenutno fokusirano, odnosno u koje se trenutno unosi određeni datum. Pomoću tablice u HTML-u su raspoređena polja za unos datuma te njihovi opisni naslovi, te je u CSS-u definirana širina navedene tablice te font koji se koristi u tablici, tj. u ovom slučaju naslovima koji opisuju polja za unos.

Klasa pod nazivom *kontenjer* u kojoj su smješteni svi elementi ove stranice, zbog estetskih razloga smještena je na desnu stranu web stranice, pozicija ove logičke cjeline definirana je razmakom postavljenim na 10% od krajnje desne točke web stranice pomoću oznake *margin-right: 10%* te na 5% od vrha stranice pomoću oznake *margin-top: 5%.* Tekst unutar logičke cjeline *kontenjer* poravnat je u lijevo, a veličina logičke cjeline je određena pomoću oznaka *width* i *height*, kojima je definirana širina logičke cjeline u iznosu od 20% cijele stranice, te visina koja iznosi 70% stranice.

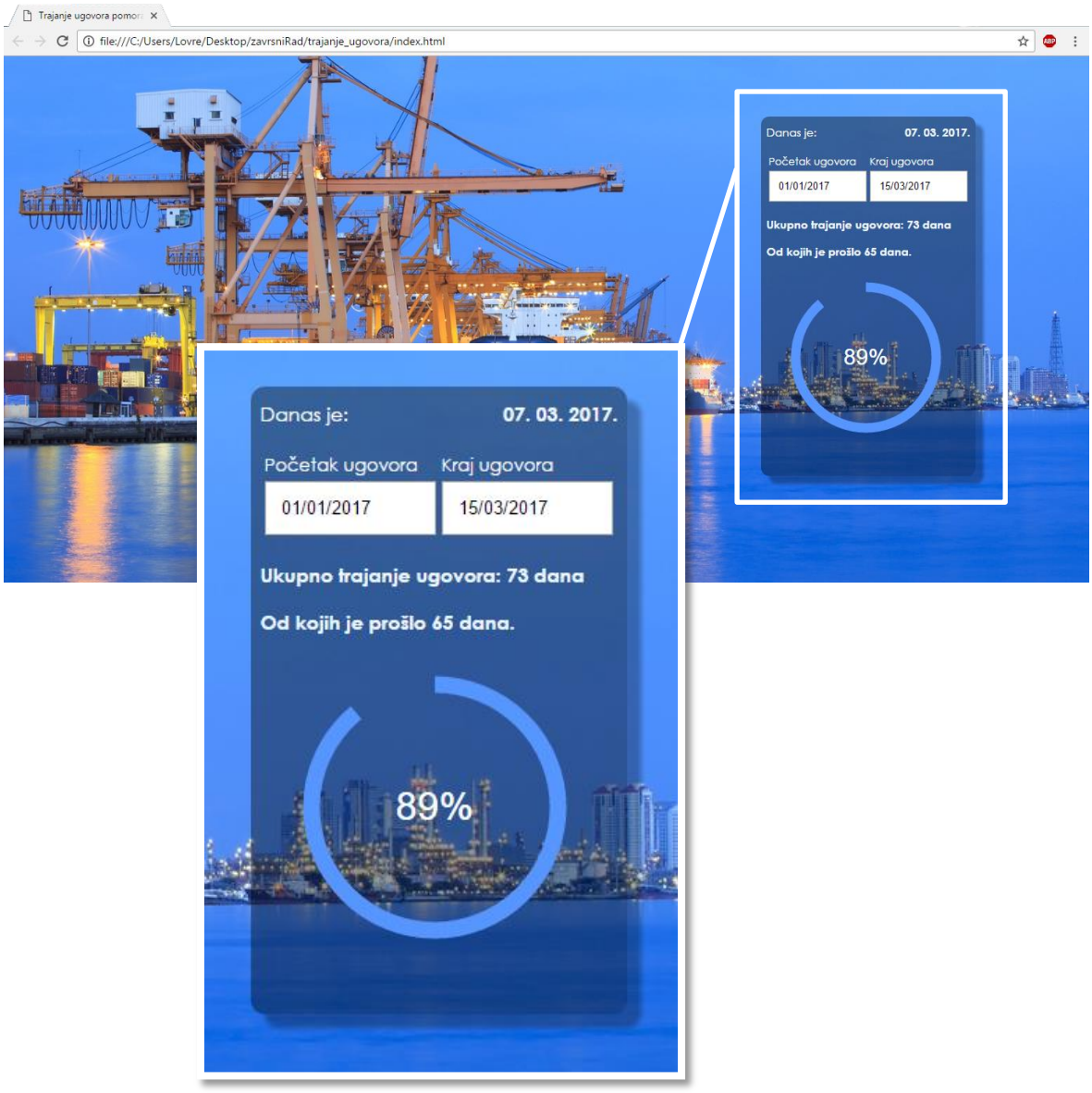
Unutar klase *kontenjer* definirana je logička jedinica *block,* koja određuje izgled okvira u kojem su smješteni svi bitni elementi web stranice. Pomoću online generatora za generiranje izgleda okvira postavljena je pozadina okvira u crnu boju, te smanjena vidiljivost pozadine na 40% kako bi okvir došao do izražaja u odnosu na sliku pozadine web stranice. Također, postavljeni su i zaobljeni rubovi okvira te sjena koja pada sa okvira na sliku pozadine. Širina i visina okvira postavljeni su na 100%, s obzirom da se okvir *block* nalazi u logičkoj cjelini *kontenjer* širina i visina okvira zauzeti će vrijednost 100% širine i visine logičke cjeline *kontenjer.*

Klasa *tekst* definira poziciju teksta koji se ispisuje u logičkoj cjelini *kontenjer,* te njegovu udaljenost od svih rubova logičke cjeline *kontenjer,* odnosno *block.* Klasama *days, doDanas* i *datum* definira se stil teksta koji JavaScript ispiše kao riješenje upita, stil teksta postavljen je na *bold* i pomoću oznake *color* promijenjena je boja teksta.

Nakon što su stilski izgledi svih cjelina i elemenata stranice definirani, stranica je gotova, funkcionalna i spremna za korištenje (slike 15 i 16).



Slika 15. Izgled gotove web aplikacije prilikom unosa podataka



Slika 16. Izgled gotove web aplikacije prilikom ispisa rezultata

5. ZAKLJUČAK

Razvojem tehnologije i evolucijom interneta javljala se potreba za sve pristupačnijim informacijama, bržim djeljenjem podataka, vijesti, slika i multimedije. Statičke stranice odlaze u povijest, a na njihovo mjesto došle su one dinamičke. Sa dinamičkim stranicama korisnici se svakodnevno susreću, u vidu raznih portala, internet trgovina i mnogih drugih. Jednako tako, današnji užurbani stil života, mogućnosti tehnologije i veliki broj elektroničkih uređaja koje čovjek koristi prouzročili su potrebu bržeg i mobilnijeg pristupa raznim računalnim aplikacijama, informacijama i uslugama koje nam one pružaju. Dolazi do pojave web aplikacija – aplikacija koje su smještene na internetu i pristupa im se pomoću web preglednika. Danas, popularnost web aplikacija doživljava veliki procvat. Sve više lokalnih računalnih aplikacija dobiva svoje internetske inačice, tvrtke prebacuju svoja poslovanja na web aplikacije, kako bi uštedile resurse i podigle poslovanje na višu razinu. Fizičke osobe sve više koriste web aplikacije u oblicima internet trgovina, internet bankarstva, raznih programa za uređivanje tekstualnih ili multimedijskih sadržaja i sl.

U ovom radu obrađene su i tehnologije pomoću kojih se web aplikacije razvijaju, a kao uzročno-posljedična veza stalnog rasta i razvoja web aplikacija došlo je do potrebe i razvoja samih tehnologija za razvoj. Tako se nakon više od 10 godina stagniranja na tom području javila potreba za unaprijeđenjem HTML koda, te dolazi do pojave HTML5 inačice. Uz HTML5 javlja se još i unaprijeđena verzija CSS-a pod nazivom CSS3, ali i novija verzija JavaScripta, Microsoftovog ASP.Net-a i mnogih drugih.

Može se zaključiti da su web aplikacije sa svojom popularnošću doprinjele velikom razvoju World Wide Weba, ali i cijelog interneta, te da će se i dalje razvijati i zbog svoje pristupačnosti privlačiti sve više i više korisnika.

LITERATURA

[1] A comparison of frontend and backend web development, URL: https://blog.bloc.io/frontend-vs-backend-web-development/ (15. siječnja, 2017.)

[2] Abrus, L.: *Izrada Weba: Abeceda za webmastere*, Bug, Zagreb, 2003.

[3] Adobe Cold Fusion, URL: https://en.wikipedia.org/wiki/Adobe_ColdFusion (23. siječnja, 2017.)

[4] Barać, A.: Razvoj dinamičkih web aplikacija, Diplomski rad, Pomorski fakultet u Rijeci, Rijeka, 2014.

[5] Croatian PHP User Group, URL: http://php.com.hr/ (22. siječnja, 2017.)

[6] Dynamic HTML, URL: https://en.wikipedia.org/wiki/Dynamic_HTML (20. Siječnja, 2017.)

[7] Dynamic web pages, https://en.wikipedia.org/wiki/Dynamic_web_page (15. siječnja, 2017.)

[8] Free jQuery Plugins and Tutorials, URL: http://www.jqueryscript.net (15. veljače, 2017.)

[9] Java (programming language), URL: https://en.wikipedia.org/wiki/Java_(programming_language) (23. siječnja, 2017.)

[10] MySQL, URL: https://en.wikipedia.org/wiki/MySQL (23. siječnja, 2017.)

[11] NextGen Websites: The Differences between Static and Dynamic Websites, http://www.next-generation-websites.com/the-differences-between-static-and-dynamic-websites/ (16. siječnja, 2017.)

[12] Python (programming language), URL: https://en.wikipedia.org/wiki/Python_(programming_language) (24. siječnja, 2017.)

[13] The ASP.Net Site, URL: https://www.asp.net/ (22. siječnja, 2017.)

[14] URL: http://plusjade.ruhoh.com/ (15. siječnja, 2017.)

[15] W3Schools Online Web Tutorials, URL: http://www.w3schools.com/ (20. siječnja, 2017.)

[16] Web application, URL: https://en.wikipedia.org/wiki/Web_application (17. siječnja, 2017.)

POPIS SLIKA

POPIS KRATICA

API *(eng. Application Programming Interface)*	aplikacijsko programsko sučelje
ASP *(eng. Active Server Pages)*	skriptni jezik koji se koristi na strani poslužitelja
CSS *(eng. Cascading Style Sheet)*	mehanizam oblikovanja izrađenih web-stranica
DHTML *(eng. Dynamic Hypertext Markup Language)*	dinamički html
HTML *(eng. Hypertext Markup Language)*	jezik dizajniran za izradu web-stranica
IE *(eng. Internet Explorer)*	internet pretraživač
IP *(eng. Internet Protocol)*	internet protokol
LAMP	linux, apache, mysql, php
P2P (eng. Peer to peer)	ravnopravna mreža
SQL *(eng. Structurred Querry Language)*	programski jezik koji se koristi za rad s relacijskim bazama podataka
XML *(eng. eXtensible Markup Language)*	proširivi jezik za označavanje