SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TENOLOGIJA

Sveučilišni studij

WEB APLIKACIJA ZA NATJECANJE U PROGRAMIRANJU

Završni rad

Tomislav Gudelj

Osijek, 2016.

SADRŽAJ

1. UVOD

Cilj ovog završnog rada je napraviti web aplikaciju koja omogućava natjecanje u pisanju programa u programskom jeziku C++, a i također omogućava korisnicima da testiraju i usporede rezultate pokretanja vlastitog programskog koda s unaprijed poznatim rezultatima koji su očekivani prilikom rješavanja nekakvog ponuđenog programskog zadatka ili problema. Drugim riječima, korisnici će moći „riješiti" programski problem tako što će osmisliti i stvoriti vlastiti programski kod koji za cilj ima izračunavanje vrijednosti i veličina zatraženih od unaprijed definiranih programskih zadataka. Opisana web aplikacija imat će mogućnost izvijestiti korisnika o točnosti njegovog programskog koda, pa čak i pomoću rang liste usporediti točnost više različitih programskih kodova s ciljem kreiranja kompetitivne atmosfere.

1.1. Zadatak završnog rada

Potrebno je napraviti internet stranicu za bodovanje ispravno napisanog programa (C++). Korisnici bi postavili svoje rješenje i na temelju rezultata kompajliranja dobili bi određeni broj bodova. Cilj je relativno usporediti dva suparnika, odnosno donijeti odluku tko je bolji.

2. PRIMJENJENE TEHNOLOGIJE I ALATI

Prije započinjanja procesa pisanja programskog koda za bilo kakvo programsko rješenje, potrebno je osmisliti „stog“ programskih tehnologija – niz tehnologija počevši od platfore do konačnog programskog jezika – na temelju kojih će ono biti realizirano. Naime, iako su programske tehnologije uglavnom dizajnirane bez implikacija okolnih tehnologija uz koje će biti korištene, odnosno pojedinačni se dijelovi nekakvog rješenja uvijek mogu integrirati u samo rješenje uz ostale dijelove različitih tehnologija, uz, naravno, moguće kompromise između njih, svakako je potrebno pažljivo planirati „stog rješenja“ uz viziju zahtjeva na konačno rješenje. U slučaju web aplikacije, kao pojedinačne dijelove stoga rješenja potrebno je definirati operativni sustav na kojem će se aplikacija izvršavati, softver web poslužitelja koji će omogućiti pristup aplikaciji s interneta (uz pretpostavku da je računalo koje je zaduženo za posluživanje aplikacije spojeno na internet) kao i programski jezik ili skup programskih jezika na temelju kojih je ista stvorena. Neki od već definiranih i popularnih programskih stogova su: [1]

- LAMP
  - Linux (operativni sustav)
  - Apache (web poslužitelj)
  - MySQL ili MariaDB (sustav održavanja baza podataka)
  - Perl, PHP ili Python (programski jezici)
- WINS
  - Windows Server (operativni sustav)
  - Internet Information Services (web poslužitelj)
  - .NET (*software framework* – svojevrsan skup programskih jezika)
  - SQL Server (sustav održavanja baza podataka)

Arhitekt nekakve aplikacije ili čak kompletnog softverskog rješenja ima slobodu ne pratiti konvencije ili popularne kombinacije prilikom definiranja softverskog stoga svoga budućeg projekta, ali tada mora biti spreman na moguću potrebu razrade dodatnoga plana saniranja „nesuglasica“ na razini pojedinih programskih tehnologija – članova softverskog stoga. Naime, neke programske tehnologije impliciraju postojanje neke druge programske tehnologije unutar istog softverskog stoga, poput .NET („dot net“) *framework*-a koji većinom zahtjeva Microsoft Windows operativni sustav (drugim riječima, .NET većinom nije *cross-platform*, nije osmišljen za korištenje na više različitih platformi, gdje ulogu platforme vrši operativni sustav). Osobina ponekih tehnologija da zahtijevaju postojanje nekih drugih tehnologija u programskom stogu

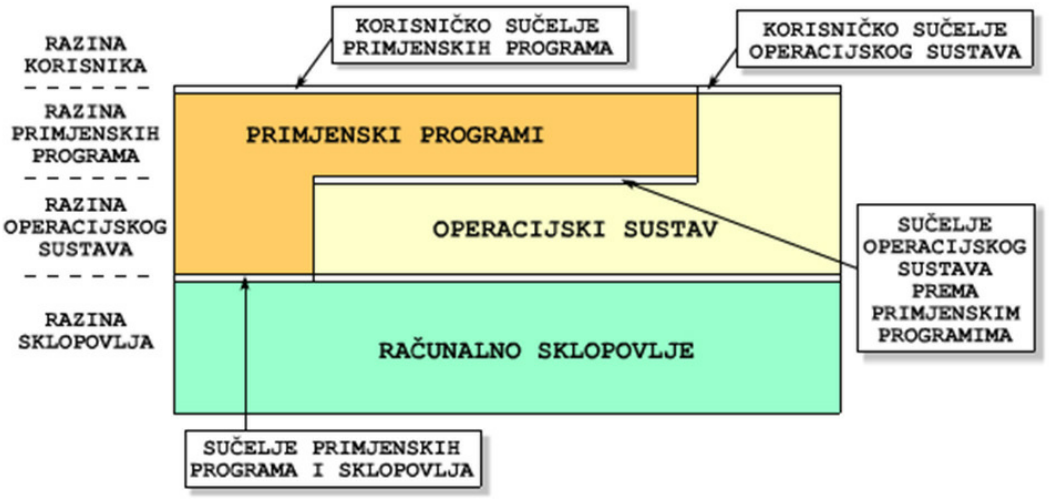
ponekad stvara potrebu dodatnih prilagodbi postojeće softverske opreme od strane samog arhitekta.

Prema poznatom algoritmu rješavanja problema odluke, *Ockhamovoj britvi*, ono rješenje koje uvodi najmanji broj implikacija ili pretpostavki je ujedno najčešće i optimalno. Ockhamova britva često služi kao nepisano pravilo i u svijetu programiranja, a manifestira se u činjenici da programeri nesvjesno većinom odabiru onaj način rješenja problema koji ostavlja najmanje nepoznanica, odnosno uvodi najmanje pretpostavki, što je i jedan od razloga zbog kojeg konvencije u programiranju bivaju ustanovljene. Stoga, u realizaciji projekta web aplikacije za natjecanje u programiranju upotrijebljena je takva skupina programskih tehnologija koja favorizira dugovječne i oprobane tehnologije. U predstojećim su poglavljima definirane i pobliže opisane sve programske tehnologije korištene u realizaciji web aplikacije za natjecanje u programiranju, počevši od operativnog sustava.

2.1. Operativni sustav

Operativni sustav dio je sustavskog softvera nekog računala, softvera zaduženog za pružanje „usluga“ nekom drugom softveru istog računala. No, operativni sustav po važnosti nije ravan bilo kojem drugom softveru, već važniji. Operativni sustav takav je softver koji za zadaću ima stvoriti softversku infrastrukturu za efikasno iskorištenje fizičkih resursa – sklopovlja računala. Naime, sklopovlje računala je skup mikroelektroničkih uređaja koji definira izrazito oskudan i intolerantan način korištenja imenom strojni kod (engl. *machine code)*. Strojni jezici definirani su od strane pojedinačnih uređaja, često uz razlike pri svakoj novoj seriji istonamjenskih uređaja, a opisani tehničkom specifikacijom istih. Naredbe strojnih jezika temelje se na operacijama binarnog računanja i čitanjem te pisanjem rezultata istog na određene memorijske lokacije fizičkih memorijskih sklopova računala. Razvijati bilo kakvu aplikaciju koristeći strojni kod nevjerojatno je neefikasan pristup (zahtjeva velik trud oko preciznog određivanja postojećih i budućih memorijskih lokacija operacija i njihovih rezultata), a osim toga i sklon razvijanju nenamjernih grešaka u programu (engl. *bugs*). Uz navedene nedostatke, željena bi se aplikacija morala nanovo razvijati u slučaju izmjene ikakvog uređaja *hardware*-a, računalnog sklopovlja. Uloga operativnog sustava je implementirati razinu apstrakcije između programera i njegovog računalnog sklopovlja, preuzimajući dio odgovornosti oko predviđanja prednosti i mana svakog računalnog sklopa, a na temelju poznatih tehničkih specifikacija istih. Operativni je sustav, dakle, softver koji programeru omogućava rad uz manje varijabli nastalih zbog razlika pojedinih uređaja računalnog sklopovlja, što za krajnji rezultat ima spontani razvoj apstraktnijih

programskih jezika više razine (engl. *high-level programming languages*), a ostale su mu zadaće i upravljanje hardverom, održavanje memorijske alokacije izvršavanih programa kao i automatsko reguliranje vremena izvršavanja svakog od trenutno aktivnih programa, radi kreiranja pseudoparalelnog modela izvršavanja trenutno aktivnih programa.



Slika 2.1: Smještaj operativnog sustava u računalu. [2]

Slika 2.1 opisuje smještaj operativnog sustava u računalu. Duljinama doticaja bloka koji predstavlja operativni sustav sa svim drugim blokovima slikovito je dočaran udio svih sučelja u računalu: glavna je funkcija operativnog sustava posredništvo između računalnog sklopovlja i primjenskih programa, koji, iako samostalno imaju pristup nekim dijelovima sklopovlja (primjerice, aplikacija za reprodukciju glazbe izravno može utjecati na zvučnike), za većinu svojih funkcija koje zahtijevaju isto moraju putem tzv. sustavskih poziva – poziva operativnog sustava „u pomoć“ – dobiti pravo i metodu na pristup nekim dijelovima *hardware*-a.

Operativne je sustave (OS-e) moguće podijeliti na tipove: [3]

- *Singletasking* – OS-i koji istovremeno mogu izvršavati najviše jedan program.
- *Multitasking* – OS-i koji mogu paralelno izvršavati i više programa putem nekakvog algoritma raspodjele vremena na svaki od procesa.
- *Singleuser* – OS-i koji dozvoljavaju jedan set korisničkih postavki, odnosno samo jednog korisnika. Nisu izravno povezani sa *Singletasking* ili *Multitasking* OS-ima.
- *Multiuser* – OS-i koji omogućavaju više setova korisničkih postavki radi kreiranja okoline u kojoj više različitih korisnika može koristiti računalo, pojedinačno ili

istovremeno, ovisno o broju krajnjih terminala. Nisu izravno povezani sa *Singletasking* ili *Multitasking* OS-ima.

- Distribuirani – OS-i koji mrežu računala objedinjuju tako da nalikuje na jedno računalo. Ovakvi distribuirani sustavi korisni su u znanstvenim istraživanjima koja zahtijevaju paralelnu obradu podataka različitim tipovima računala.
- Ugrađeni (engl. *embedded*) – OS-i prenosivih ili izoliranih uređaja, uglavnom visoke efikasnosti i razine specijalizacije za zadatak.
- OS-i realnog vremena (engl. *Real-time*) – OS-i koji koriste napredne algoritme zakazivanja zadataka tako da bi se postigao determinizam u ponašanju računala. Korisni su kod sustava koji apsolutno zahtijevaju OS koji garantira procesuiranje zadataka u određenom vremenu.

Web aplikacija za natjecanje u programiranju aplikacija je takve prirode da ne zahtijeva nikakve vrlo precizne događaje, matematički ili vremenski, niti posebna prava na izvršavanje u sustavu, pa stoga ni ne zahtijeva visoko specijaliziran operativni sustav kao platformu koja joj pruža usluge. Stoga je za realizaciju takve aplikacije dovoljan bilo koji operativni sustav osobnih računala današnjice. Osim toga, takva je aplikacija lako *portabilna* (engl. *portable*, prenosiv) između sličnih opisanih sustava uz obvezne, ali malobrojne prilagodbe, koje će biti objašnjene u predstojećim poglavljima koja se tiču tehnologija u središtu problema prilagodbe.

Među popularnim potrošačkim operativnim sustavima današnjice mogu se izdvojiti Microsoft Windows i pojedinačne distribucije besplatnih operativnih sustava objedinjene nazivom Linux, a za svrhe izrade web aplikacije za natjecanje u programiranju iskorišten je Microsoft Windows 7 kao operativni sustav i platforma.

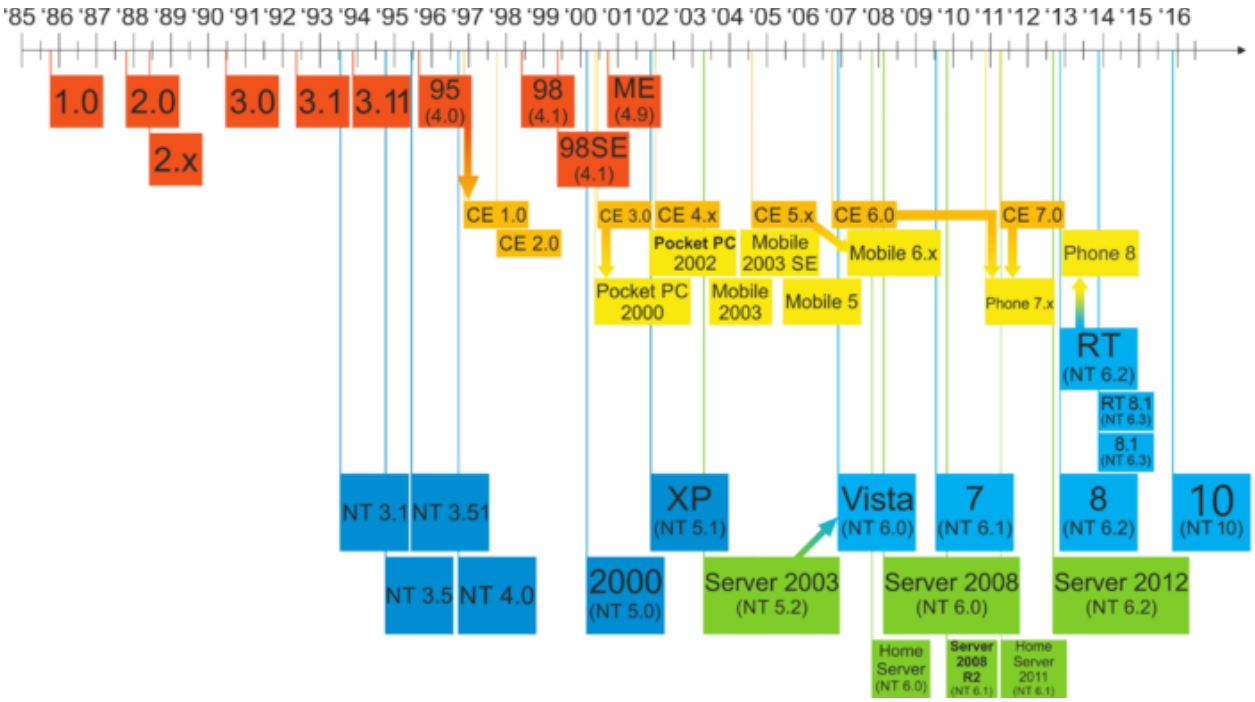
Microsoft Windows naziv je koji objedinjuje niz komercijalnih grafičkih operativnih sustava proizvedenih od strane američke tvrtke Microsoft. Potkategorije trenutno aktivnih Microsoft Windows operativnih sustava su: [4]

- Windows NT
  - „Windows“ (među kojima je i Windows 10)
  - Windows Server
  - Windows PE
- Windows Embedded (sustav osmišljen za ugrađene uređaje, poput digitalnih satova)
- Windows Phone

Prijašnje, danas neaktivne potkategorije Microsoft Windows operativnih sustava su:

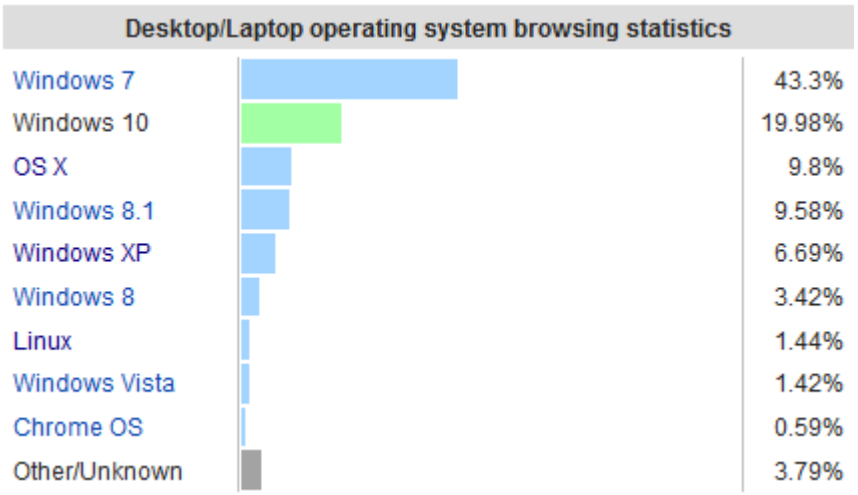
- Windows 9x
- Windows Mobile

Prve verzije Windows operativnih sustava proizvedene su u studenom 1985. godine kao odgovor Apple-ovom operativnom sustavu, ali nisu ostvarile velik tržišni uspjeh. [5] Tadašnji Windows 1.0 nije bio cjelovit operativni sustav, već proširenje Microsoftovog MS-DOS tekstualnog (bez bogatog grafičkog sučelja) operativnog sustava. [5] U kontrastu na većinu Linux distribucija, Microsoft Windows operativni sustavi su *closed-source*, što znači da izvorni programski kod na temelju kojeg su stvoreni nije dostupan javnosti za svojevoljno uređivanje, a krajnji rezultat čega je nepostojanje brojnih nijansi operativnih sustava kakve poznajemo kod Linux-a.



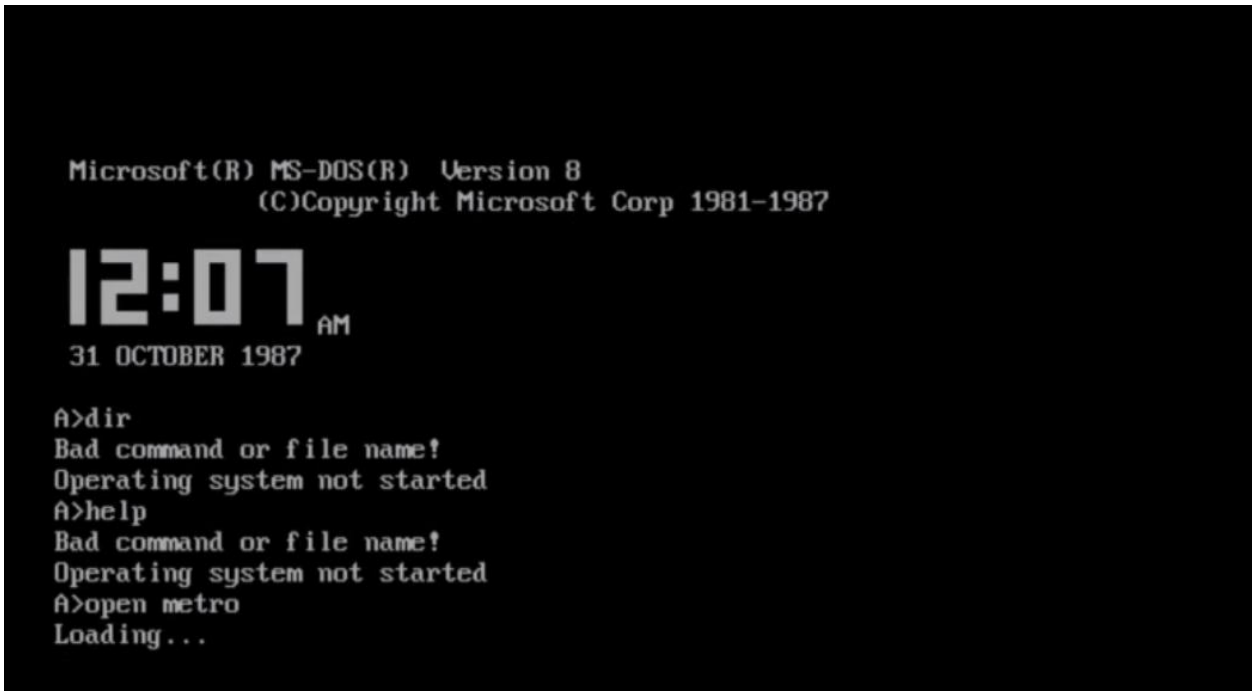
Slika 2.2: Vremenska lenta Microsoft Windows operativnih susava [6]

Na slici 2.2 prikazan je kronološki razvoj Microsoft Windows operativnih sustava na vremenskoj osi, a vidljivo je i kako isti započinje sustavom Windows 1.0, a završava današnjim modernim operativnim sustavom Windows 10.

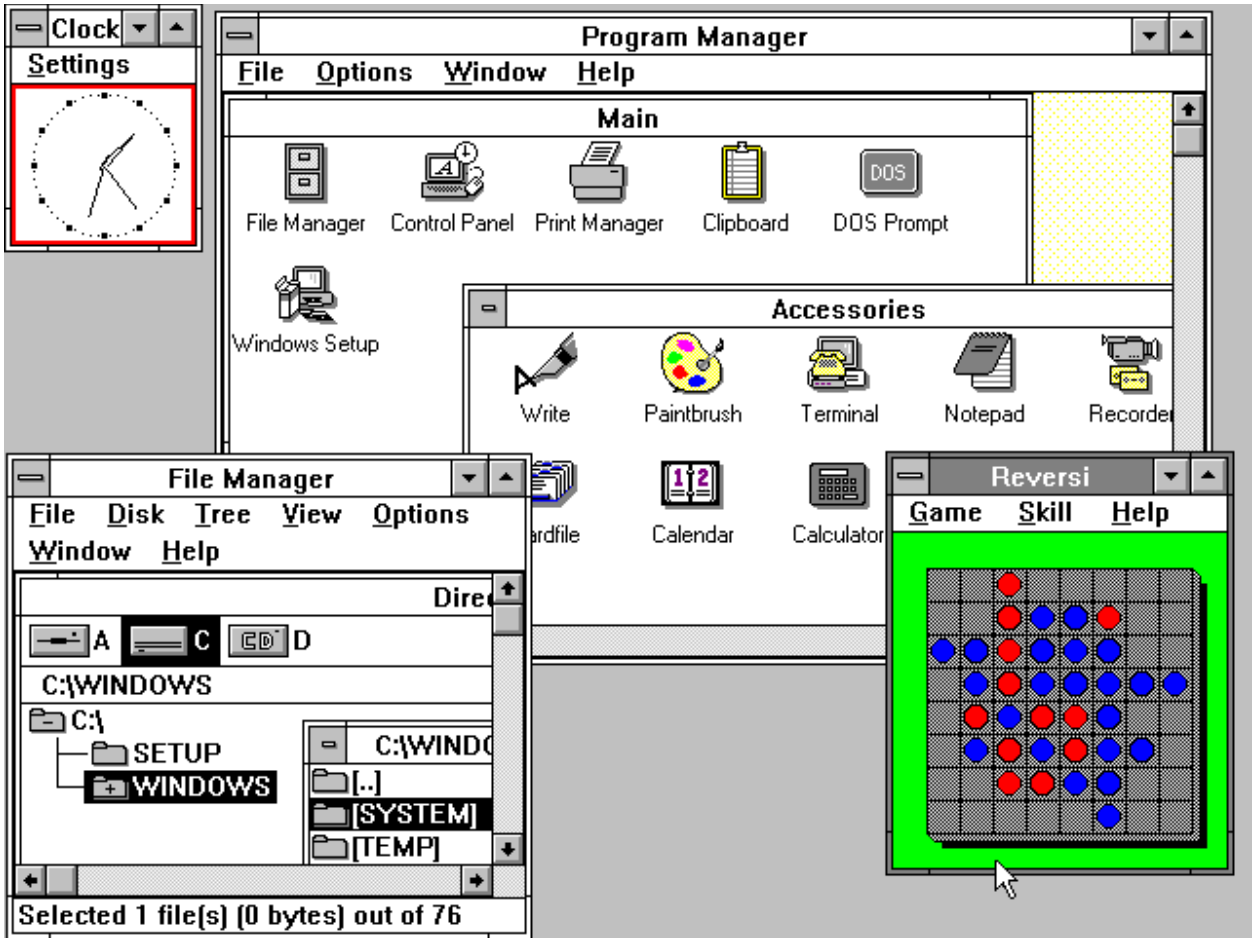


Slika 2.3: Udio pojedinih operativnih sustava u računalima današnjice koja su spojena na internet, a prema statistici alata za analizu internetskog prometa StatTraffic, svibanj 2016. godine [7]

Slika 2.3 prikazuje zastupljenost najpopularnijih operativnih sustava na aktivnim računalima današnjice, prema statistici alata za analizu internetskog prometa StatTraffic obavljenoj u svibnju 2016. godine. Izostanak eksplozije popularnosti operativnog sustava Windows 10 vjerojatno leži u činjenici da od svojih prethodnika ne uvodi velike novitete, osim unaprijeđenog grafičkog sučelja.



Slika 2.4: Sučelje operativnog sustava MS-DOS [8]



Slika 2.5: Grafičko sučelje operativnog sustava Windows 3.0 iz 1990. godine [9]

Na slikama 2.4 i 2.5 moguće je primijetiti razliku u skoku s tekstualnog korisničkog sučelja na grafičko korisničko sučelje. Iako grafičko korisničko sučelje nije neophodno za realizaciju sofisticiranih poslužiteljskih računala, mali su korisnici favorizirali grafičko sučelje zbog pojednostavljenja korištenja većine aplikacija, što je dovelo do popularizacije sustava s grafičkim korisničkim sučeljem na budućem tržištu. Svi operativni sustavi navedeni na slici 2.3 grafičkog su korisničkog sučelja.

Slika 2.6: Grafičko sučelje operativnog sustava Windows 10 iz 2015. godine [10]

Prilagođena inačica Windows 8 operativnog sustava (danas Windows 10) napaja i igraću konzolu Xbox One.

2.2. „HTTP/Web server“ – aplikacija web poslužitelja

Web poslužitelj je računalni sustav koji procesuira zahtjeve kroz *HTTP* protokol (*Hypertext Transfer Protocol*). Ovaj termin može označavati cijeli računalni sustav, ali i samo softver koji nadgleda HTTP zahtjeve. [11] U daljnjem tekstu pojam web poslužitelja označavat će samo softver koji nadgleda HTTP zahtjeve.

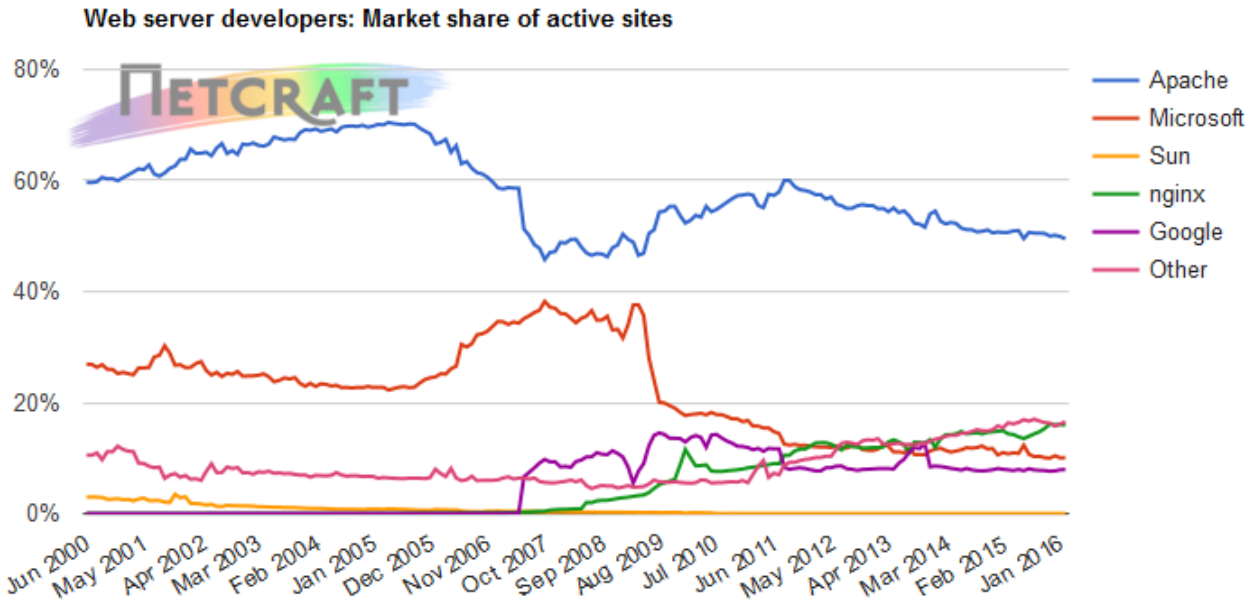
Svako računalo prilikom spajanja na internet od strane svog davatelja usluge dobiva *Internet Protocol* adresu, 32-bitnu brojčanu vrijednost koja za cilj ima tom računalu pružiti jedinstveni identifikator na internetu kako bi ono moglo biti kontaktirano. Proces dodjeljivanja *IP* adresa od strane davatelja usluge u konačnici je reguliran od strane američke neprofitne privatne tvrtke *Internet Assigned Numbers Authority – IANA*. Kada računalo ne bi imalo dodijeljenu IP adresu, ono ne bi moglo biti kontaktirano ni od kojeg drugog računala na Internetu jer nema jedinstvenu oznaku na mreži. No, čak uz dodijeljenu IP adresu, računalo u pravilu ne može poslužiti (engl. *host*) nekakvu web stranicu na Internetu, gdje je web stranica u suštini neka skupina datoteka na tom računalu. Uloga web poslužitelja je, dakle, učiniti određene datoteke istog računala dostupne na Internetu.

Postoji nekoliko već oprobanih web server programa, a statistiku popularnosti najpopularnijih je odredila udruga NetCraft:

Proizvođač	Naziv	Veljača 2016 [broj aktivnih poslužitelja]	Udio [%]
Apache	Apache	84,790,816	49.50%
nginx	nginx	27,327,583	15.95%
Microsoft	IIS	17,257,986	10.08%
Google	GWS	13,596,295	7.94%

Tablica 2.1: Najpopularniji web poslužitelji [12]

Tablica 2.2 predstavlja zastupljenost najpopularnijih programa za web posluživanje kao i njihov udio u svim aktivnim računalima poslužiteljima.



Slika 2.7: Tablica 2.1 predočena dijagramima [12]

S obzirom na to da je web aplikaciju za natjecanje u programiranju potrebno poslužiti na Internet da bi joj potencijalni korisnici mogli pristupiti, softver web poslužitelja neophodan je u realizaciji ovog projekta. Uz pretpostavku da poneki korisnici Microsoft Windows operativnog sustava nemaju na raspolaganju IIS, jedan od navedenih programa za web posluživanje, a inače softver sadržan u svim modernim inačicama Microsoft Windows operativnih sustava, ili da istim ne znaju rukovati, za realizaciju ovog projekta odabran je Apache web poslužitelj.

Apache HTTP Server web poslužitelj je osmišljen 1995. godine od strane udruge *The Apache Software Foundation*, a najpopularnijim takvim softverom na Internetu ostao je do danas. No,

često dolazi kao dio skupine programa (engl. *bundle*) koji zajedno tvore skup tehnologija za izradu web aplikacija nalik ideji programskog stoga opisanoj u poglavlju 2. Kao takav programski paket za svrhe ovog projekta odabran je *XAMPP* (neslužbena kratica od *Cross-platform Apache + MariaDB/MySQL + PHP + Perl*), Apache distribucija implementirana za Windows, Linux i OS X. Trenutačno, na datum izrade ovog rada najnovija je stabilna verzija ovog paketa bila v5.6.21, sukladno istoj verziji PHP poslužitelja sadržanog u samom paketu.

Dužnost web poslužitelja je „mapirati" – povezati, uskladiti – *URL* (*Uniform Resource Locator*) s nekakvim podatkovnim resursom računala na kojeg se pokušava pokazati istim URL nazivom, gdje je URL tekstualna referenca na neku datoteku računala. Drugim riječima, kada kontaktirajući udaljeno računalo korisnik zatraži datoteku iz određenog direktorija čiju je simboličku adresu u memoriji računala definirao URL referencom, dužnost web poslužitelja je korisniku dostaviti tu datoteku, prijaviti da ona ne postoji ili u potpunosti odbiti zahtjev, uz objašnjenje ili bez istog. Da bi bilo moguće predstaviti primjer ovog procesa, potrebno je definirati *web pretraživač* (engl. *web browser*).

Web pretraživač (u daljnjem tekstu *browser*) je program koji šalje zahtjeve za primanjem tekstualnog sadržaja preko interneta, obrađuje eventualno primljen sadržaj i, na temelju pravila zadanih od strane programskih jezika web programiranja, prevodi tekstualni sadržaj u pripadajući vizualni sadržaj, sastavljen od teksta, grafike kao i interaktivnih objekata s kojima korisnik može izmjenjivati informacije, poput obrazaca (engl. *forms*). Svaki browser svojevrstan je sustav jer, unatoč jasno definiranim pravilima navedenih programskih jezika, autori istog i dalje samostalno odlučuju o istinskom načinu interpretacije primljenih podataka, što implicira razlike u interpretaciji istog resursa od više različitih browsera.

Prvi browser bio je *WorldWideWeb* proizveden 1990. godine od strane direktora udruge *World Wide Web Consortium* (*W3C*), udruge koja koordinira kontinuiranim razvojem web tehnologija. Današnji najpopularniji browseri su *Google Chrome*, *Safari*, *Mozilla Firefox*, *Internet Explorer* i *Opera*.

Promotrimo sad primjer URL obrađivanja: [13]

Korisnik pomoću svojeg browsera zatraži sljedeći resurs:

```
http://www.example.com/path/file.html
```

Njegov browser tada prevodi zahtjev u sljedeći kod:

```
GET /path/file.html HTTP/1.1
Host: www.example.com
```

Web poslužitelj na simboličkoj adresi `www.example.com` shvatit će da klijent želi pristupiti datoteci

```
/home/www/path/file.html
```

Jer po zadanim postavkama isti poslužitelj prvo pretražuje direktorij `/home/www/` (na Windows sustavu to bi bilo `C:/Users/korisnik/www/`). Ako u opisanom direktoriju postoji direktorij imenom `path` u kojem postoji datoteka imenom `file.html`, web poslužitelj klijentu će poslati sadržaj te datoteke, što će njegov browser obraditi.

Prostor svih URL referenci objedinjen je nazivom *World Wide Web*.

Za svrhe razvoja web aplikacije za natjecanje u programiranju iskorišten je programski paket XAMPP koji objedinjuje Apache web poslužitelj, kao i još nekoliko tehnologija.

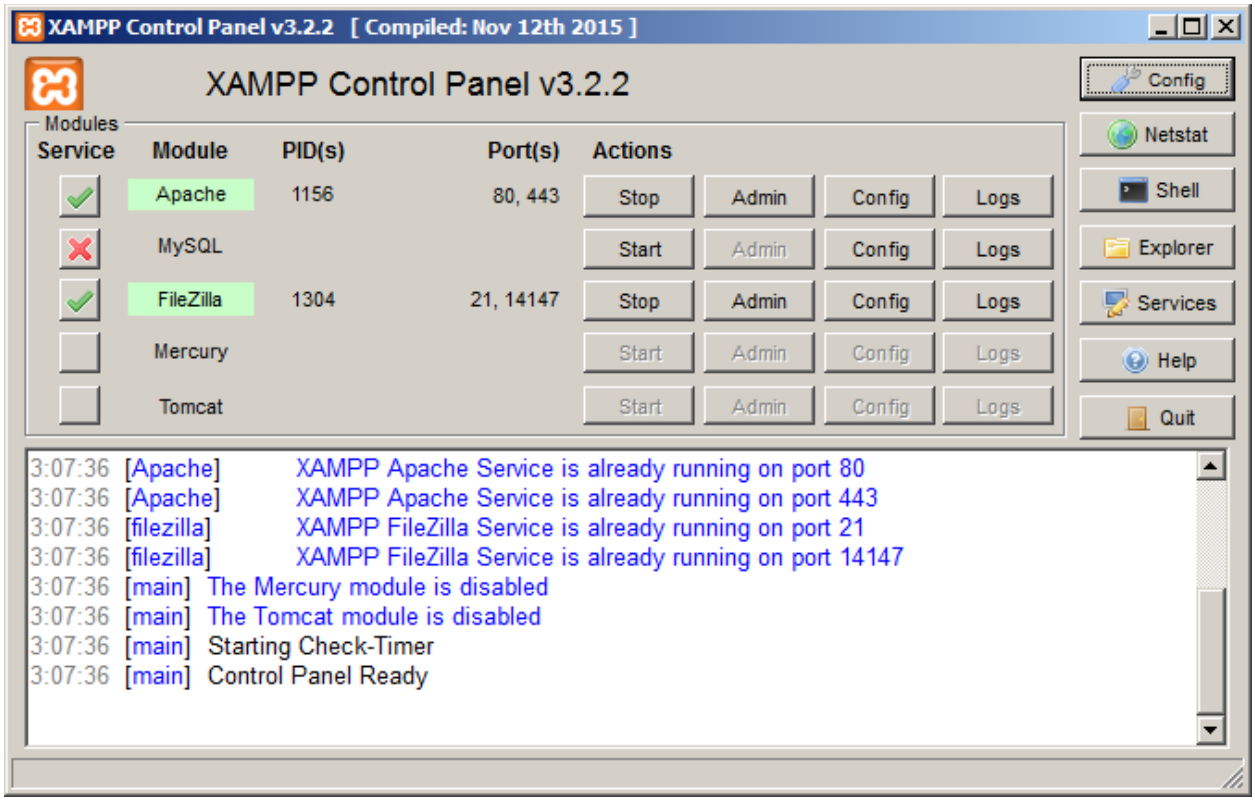
„XAMPP je najpopularnije okruženje za razvoj PHP aplikacija

XAMPP je u potpunosti besplatna Apache distribucija jednostavna za instalirati, a sadrži MariaDB, PHP i Perl podrške. XAMPP open-source paket je osmišljen s vizijom jednostavnog procesa instalacije.“ [14]

XAMPP je moguće preuzeti na web lokaciji udruge *Apache Friends*. [15]

Tijekom instalacije, korisnik mora odabrati koje članove ovog programskog paketa želi instalirati, kao i direktorij u koji će oni biti spremljeni.

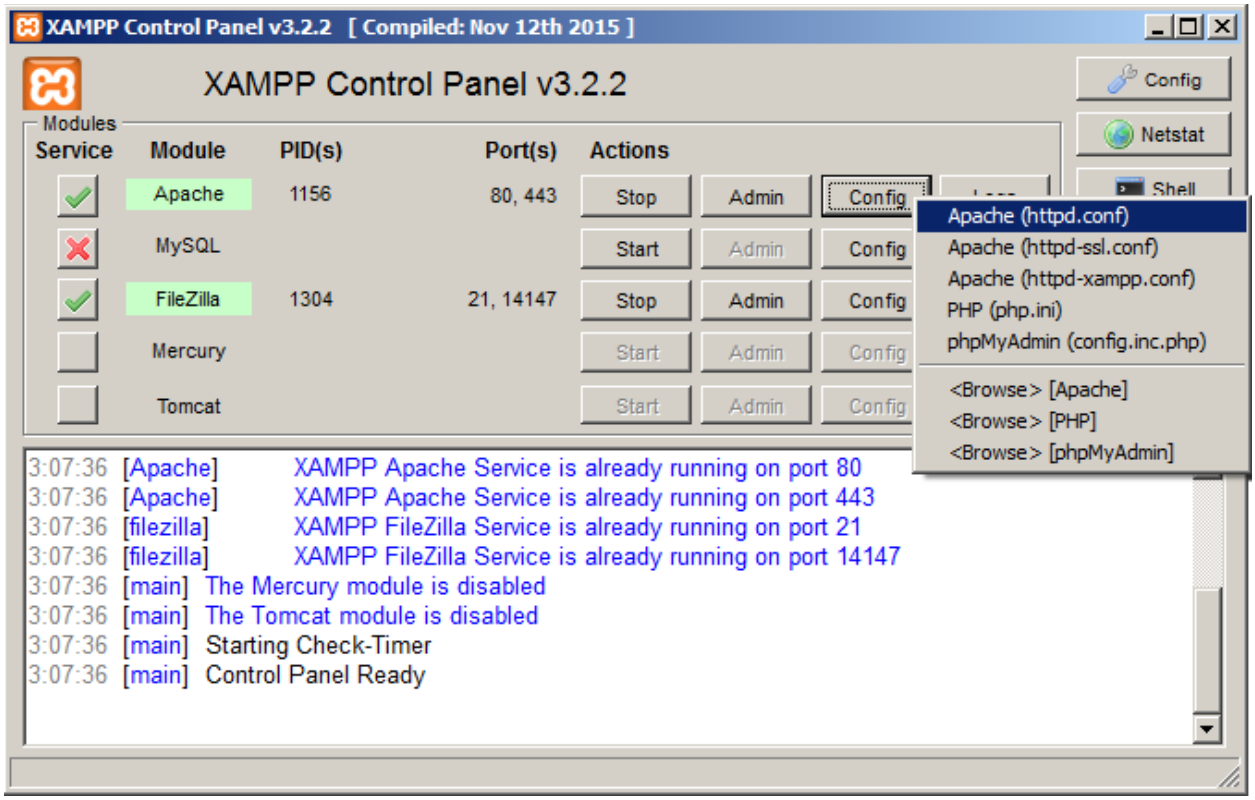
XAMPP Control Panel v3.2.2 grafičko sučelje izgleda ovako:



Slika 2.8: Grafičko sučelje XAMPP Control Panel v3.2.2 programa

Tijekom instalacije programa vidljivog na slici 2.8 odabrane su opcije instaliranja *Apache* i *FileZilla* komponenti i kao servisa, pozadinskih procesa koji u pripravnom, neaktivnom stanju koriste nekakvu količinu radne memorije, ali jedva dovoljnu da bi mogli odgovoriti na prvi upit koji prime, u slučaju čega alociraju potrebnu memoriju za izvršenje zadatka. Ovo u suštini znači da će *Apache server* i *FileZilla server* biti uvijek dostupni.

Odabirom sljedeće opcije u ovom sučelju:



Slika 2.9: Opcija konfiguracije Apache poslužitelja

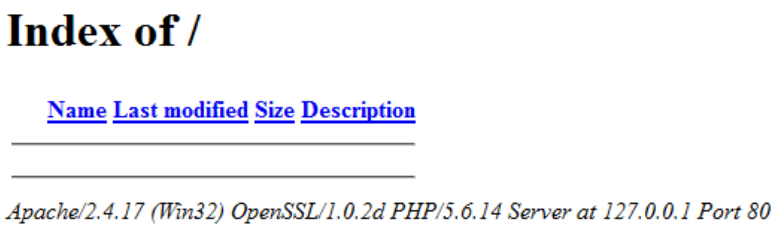
i izmjenom vrijednosti koja stoji uz odrednice `DocumentRoot` i `<Directory` tako da navode:

```
C:/Users/administrator/www
```

postiže se modifikacija nad Apache poslužiteljem koja će zadani početni direktorij poslužitelja postaviti na navedeni. Kao rezultat ove akcije, budući klijenti moći će tražiti samo datoteke koje postoje u stablastoj strukturi navedenog direktorija, direktorija `www`.

Naš direktorij `www` sada je dostupan na Internetu, ali što ako želimo zatražiti resurse istog direktorija koristeći browser na istom računalu na kojem se taj direktorij nalazi? U tu svrhu, Apache definira URL *localhost*, na IP adresi *127.0.0.1*.

Kada bi direktorij `www` bio prazan, rezultat kontaktiranja URL-a *localhost* browserom bio bi:



Slika 2.10: Rezultat kontaktiranja *localhost* lokacije kada je direktorij definiran kao `DocumentRoot` i `Directory` prazan

Po zadanim postavkama, web poslužitelji u *localhost* direktoriju pokušavaju pronaći i vratiti datoteku imenom *index.php*. Ako ona ne postoji, prioritet tad pada na *index.html*, a ako ni takav dokument ne postoji, vraćen je ishod na slici 2.10, izlistanje sadržaja direktorija `www`. Navedene datoteke redom su dokumenti napisani u *HTML* odnosno kombinaciji *HTML* i *PHP* programskih jezika.

2.3. HTML

HTML (engl. *Hyper Text Markup Language*) opisni (engl. *markup*) je jezik za stvaranje web stranica. Razvoj HTML-a započeo je Tim Bernes-Lee 1980. godine u CERN-u [16], a prva inačica HTML jezika stvorena je 1993. godine u CERN-u, a najnovija, HTML5, 2014. godine. Razvojem HTML jezika, kao i drugih jezika web programiranja rukovodi udruga W3C, Kao i svi drugi markup jezici, u HTML-u objekti se definiraju uz pomoć ključnih riječi (engl. *tags*, u daljnjem tekstu *tags*), a svaka ključna riječ opisuje neku vrstu objekta. [17] Osnovan HTML „program“, odnosno osnovna web stranica definirana pomoću HTML-a sastoji se od barem ovih tag-ova:

- `<!DOCTYPE html>` – tag koji definira dokument u kojem se nalazi kao HTML5 dokument, dokument koji podliježe standardu pete verzije HTML tehnologije.
- `<html>` – tag koji započinje odlomak koda koji opisuje cjeloviti HTML dokument – web stranicu
- `<head>` – tag koji započinje odlomak koda u kojem je moguće dokumentu definirati neke posebne vrste osobina
- `<body>` – tag koji započinje odlomak koda u kojem se definira vidljivi dio web stranice

Većinu tagova potrebno je i zatvoriti, što se čini navođenjem istog uz dodanu prednju kosu crtu (engl. *slash, forward slash*), ali unutar svojevrsnih zagrada, znakova manje i veće.

- `</head>` – Napomena: *head* odlomak potrebno je zatvoriti prije započinjanja *body* odlomka
- `</body>`
- `</html>`

HTML je programski jezik koji ne zahtijeva formalni prevoditelj, nego funkciju prevođenja HTML koda vrši browser. Zadaća web browsera je, među ostalim, interpretirati (prevesti)

HTML kod poslan od strane poslužitelja od kojeg je klijent – korisnik tog browsera – zatražio nekakav sadržaj. Interpretacija HTML koda rezultira vizualnim elementima.

Link Name is a link to another nifty site

This is a Header

This is a Medium Header

Send me mail at support@yourcompany.com.

This is a new paragraph!

This is a new paragraph!
This is a new sentence without a paragraph break, in bold italics.

Slika 2.11: Jednostavan primjer kratkog HTML koda [18]

Slika 2.11 prikazuje vizualne rezultate nekakvog kratkog HTML koda. Na slici je vidljiva slika kao rezultat pozivanja `<img>` taga uz navedeni fizički resurs slike spremljen u istom direktoriju kao i sam HTML programski kod, kao i tekst namijenjen za poglavlja ili tijelo istog.

Svaki browser ima posebno definirane algoritme interpretiranja HTML koda u vizualne objekte, pa je za očekivati da će u interpretaciji svake HTML stranice postojati razlike na razini različitih web browsera, ali je dobra programerska praksa što preciznije se držati definiranih standarda radi eliminacije ovih nepoznanica.

Poduzetni duh programera ubrzo je ishodio prilagodbom već postojećeg WYSIWYG (engl. *What you see is what you get*) softvera, softvera za preciznije iscrtavanje dizajna dokumenata s ciljem da rezultati ispisa istih na papir budu gotovo identični svojim digitalnim verzijama, tako da

podržava i HTML programiranje. Ovakav softver omogućava korisniku povlačenje (engl. *drag'n'drop*) predefiniranih objekata na budući HTML dokument i uređivanje njihovih svojstava poput lokacije, oblika i boje slova i slično, a sve bez dubokog znanja HTML programiranja. No, takav softver nikad nije mogao pružiti zadovoljavajuće rezultate egzotičnijih zahtjeva na HTML dokument, pa model ručnog ispisivanja HTML koda i koda pripadajućih tehnologija nikada nije iščezao iz prakse web programera.

HTML dokument može sadržavati i programski kod nekih potkrepljujućih web tehnologija, među kojima su *PHP*, *CSS* i *JavaScript*.

Granica kvalitete web stranica narasla bi kada bi bilo moguće na temelju širokog spektra okolnosti prilikom slanja HTML sadržaja klijentu dinamički proizvesti HTML kod, tako da on nije ručno stvoren od strane nekog programera. Takav čin moguć je korištenjem pretprocesorskih tehnologija, među kojima je PHP.

2.4. PHP

PHP (engl. *Hypertext Preprocessor*) vrlo je popularan, besplatan, *open-source scripting* programski jezik. Pojam *scripting* označava osobinu PHP-a da je uglavnom korišten za stvaranje kratkih programa posebne namjene koji automatiziraju nekakav proces koji je dotad bio upravljan korak po korak od strane sporijeg automatiziranog sustava ili čovjeka. Iako ova definicija pojma *scripting* naglašava kratkoću takvog programskog koda, ne postoji nikakvo pravo ograničenje duljine istog.

Prva verzija PHP-a pojavila se u 1995. godini u vlasništvu tvrtke Zend Technologies, ali je prva službena specifikacija stvorena tek 2014. godine, što znači da je do onda PHP bio interpretiran donekle proizvoljno od strane kreatora raznih PHP poslužitelja.

PHP je svojevrsno proširenje HTML jezika, iako nije definiran u standardu niti jedne verzije istog. Sufiks datoteka koje sadrže PHP kod je *.php*, dok je sufiks čistih HTML datoteka – *.html*. U samom nazivu jezika stoji pojam *Preprocessor*, što sugerira nekakvu radnju pretprocesuiranja, preliminarnih modifikacija nad nečim s ciljem obogaćivanja funkcija istog kada je ono u radu. Konkretno, PHP kod programski je kod koji se na računalu poslužitelju izvršava prije nego ikakav HTML sadržaj biva poslan klijentu koji je pomoću svog browser-a zatražio isti. Na temelju okolnosti, PHP kod može dinamički ispisati novi HTML kod ili pozvati nekakvu funkciju nad već postojećim HTML kodom spremnim za slanje ili pak lokalno izvršiti naredbu

na računalu poslužitelju. Da bi ovo bilo moguće, računalo poslužitelj mora posjedovati poseban softver zadužen za izvršavanje ovih radnji, a koji se aktivira u trenutku kada web poslužitelj primi zahtjev za HTML sadržajem (engl. *HTTP request*, u daljnjem tekstu: HTTP zahtjev), što znači da takav softver mora biti dio web poslužitelja, a naziva se PHP poslužitelj (engl. *PHP server*). Zadaća PHP poslužitelja je, dakle, interpretirati (izvršavati red po red) PHP kod nekog HTML dokumenta.

Blok PHP koda u HTML dokumentu započinje navođenjem niza znakova

```
<?php ,
```

a završava navođenjem niza znakova

```
?> .
```

Nakon što PHP poslužitelj obradi PHP kod nekog zatraženog HTML dokumenta, niti jedan blok koji odgovara gornjem opisu neće preostati, ali umjesto takvih blokova može biti izgeneriran novi tekst ili HTML objekti. Prema tome, klijent nikad neće primiti niti jedan blok PHP koda, samo rezultate interpretacije istih. Preciznije, klijent uopće ne može dokazati da web poslužitelj koristi PHP tehnologiju za generiranje dinamičkog HTML koda, ali iskusan programer i dalje može na temelju „ponašanja“ stranica ovisno o njegovom upitu ustanoviti radi li se o tehnologiji takve prirode.

Pokretanje sljedećeg PHP koda

```
<?php echo "<p>Hello World</p>"; ?>
```

u nadležni će HTML dokument ispisati

```
<p>Hello World</p>
```

što je HTML objekt paragrafa.

Pokretanjem sljedećeg PHP koda na Windows računalu uz administratorska prava

```
<?php exec("shutdown -s"); ?>
```

pokrenut će proceduru gašenja računala.

Funkcionalnost web aplikacije za natjecanje u programiranju opisane u uvodu ovoga rada intenzivno ovisi o moćima PHP tehnologije jer zahtijeva komunikaciju klijenta i servera u obliku

izmjene programskog koda, a zatim rezultata prevođenja i izvršavanja istog, što je proces kojem pogoduje koncept pretprocesuiranja HTML dokumenta.

2.5. CSS

Cascading Style Sheets (CSS) stilski je jezik pomoću kojeg se opisuje kakvih će vizualnih svojstava biti elementi *markup* jezika koji poznaju strukturu. CSS dozvoljava definiranje velikog niza svojstava poput veličine, boje, boje pozadine, prozirnosti, dimenzija, oblika, izgleda slova i drugih i to za jedan ili više elemenata, ovisno o HTML ID oznakama svakog elementa ili grupe elemenata definiranim od strane programera. CSS je vrlo skalabilan jer se ista *CSS skripta* može primijeniti na neograničeno HTML dokumenata, uz pretpostavku da imaju navedene identifikatore i klase definirane u skladu sa samom CSS skriptom.

Prvi put objavljen 1996. godine, a današnja je najnovija verzija, CSS2.1, još uvijek zasnovana na verziji CSS2 iz 1998. godine.

Kao i ostalim dosad navedenim tehnologijama, razvojem CSS-a rukovodi organizacija W3C.

Opisivanje stila određenog elementa veoma je jednostavan proces. Neka je definiran nekakav tekstualni HTML element:

```
<span>Hello World</span>
```

Uređivanjem elementa na sljedeći način postiže se njegova plava boja i bold-an font.

```
<span style="color: blue; font-weight: bold">Hello World</span>
```

S druge strane, pretpostavimo da neka neovisna skupina elemenata prije kraja svoje prve zatvorene zagrade ima naveden sljedeći identifikator:

```
class="abc"
```

Dodavanjem sljedećeg HTML/CSS koda u `<head>` prostoru HTML dokumenta postiže se ista promjena nad tekstom tih elemenata:

```
<style>
    .abc{
       color: blue;
       font-weight: bold;
    }
</style>
```

2.6. GCC

Zadatak ovog rada je napraviti svojevrsno okruženje za programiranje u C++-u s udaljenog računala. S obzirom na to da Windows računala po zadanim postavkama ne posjeduju moć prevođenja C++ programskog koda, potrebno je instalirati nekakav prevoditelj istog.

Postoji nekoliko dostupnih C++ prevoditelja napisanih za Windows operativne sustave, a za potrebe ovog rada odabran je GCC (*GNU Compiler Collection*). GCC je besplatni skup prevoditelja za jezike *C, C++, Obj-C, Fortran, Java, Ada, Go*. [19] GCC je originalno bio stvoren za GNU operativne sustave, ali kroz razne distribucije nalazi primjenu i na Windows-u.

U svrhu omogućavanja GCC funkcionalnosti u ovom radu odabran je programski paket, okruženje za razvoj C++ i drugih aplikacija, imenom MinGW (*Minimalist GNU for Windows*). MinGW [20] je moguće preuzeti na sljedećoj web lokaciji. [21]

Pretpostavljajući da se nekakva C++ datoteka nalazi u `MinGW/bin` direktoriju, prevesti ju u izvršnu Windows datoteku (.exe) moguće je sljedećom naredbom u terminalu Windows sustava (cmd.exe):

```
g++ program.cpp
```

Rezultat pokretanja takve naredbe izvršna je datoteka `a.exe`.

Ukoliko je potrebno stvoriti takvu datoteku specifičnog imena, potrebno je koristiti prilagođenu naredbu demonstriranu gore:

```
g++ -o naziv.exe program.cpp
```

MinGW je okruženje za stvaranje *nativnih* (engl. *native*, prirodan, urođen) Windows aplikacija, takvih da ih svi Windows korisnici koji posjeduju standardne Windows biblioteke mogu otvoriti. No, neka funkcionalnost C++ programa i dalje zahtijeva od programera da korisnicima priloži i neke potrebne biblioteke, poput u slučaju programiranja višenitnih (engl. *multi-threaded*) programa.

Prilikom instalacije MinGW potrebno je naznačiti da je željena instalacija barem C++ okruženja radi potreba web aplikacije za programiranje u C++.

2.7. JavaScript

JavaScript je jedan od osnovnih programskih jezika za web programiranje, izvršava se na strani korisnika, visoke je razine, podržavaju ga svi moderni internet preglednici, a ne prevodi (kompajlira) se u cijelosti nego se interpretira, što znači da ga navedeni preglednici prevode i izvršavaju red po red. JavaScript je pogonjen događajima, „eventima“, što znači da određene pojave, bile one uzrokovane od strane korisnika ili ne, poput klika ili pritiska tipke, izazivaju izvršavanje određenih procedura isprogramiranih od strane programera. Zbog ove se osobine JavaScript često koristi kako bi se mijenjala vizualna svojstva pojedinih HTML elemenata, ali tek pri određenim događajima, poput omogućavanja klika neke tipke tek kada su sva polja obrasca ispunjena.

```
<script>
      check_for_blank_field();
      document.getElementById("task_name").addEventListener("keyup",
check_for_blank_field);
</script>
```

Većina modernih internet preglednika tretira sav programski kod koji se nađe između `<script>` i `</script>` oznaka kao JavaScript kod. Gornji navedeni JavaScript kod, primjerice, prvo izvršava korisnički definiranu funkciju `check_for_blank_field()`, a zatim dohvaća HTML element identifikacije `"task_name"` i počinje slušati događaje (engl. „events“) koje generira taj element. Pri pronalasku događaja otpuštanja tipke tipkovnice dok je korisnikov kursor pozicioniran unutar tog elementa (`"keyup"`), poziva se ista ranije navedena funkcija.

U realizaciji web aplikacije za natjecanje u programiranju JavaScript će se koristiti samo kako bi se onemogućilo slanje zahtjeva za izradom zadatka takvog da nisu svi potrebni podaci pruženi, kao i za prisilno slanje zahtjeva za dohvaćanjem teksta nekog zadatka ukoliko korisnik odabere isti pomoću padajućeg menija.

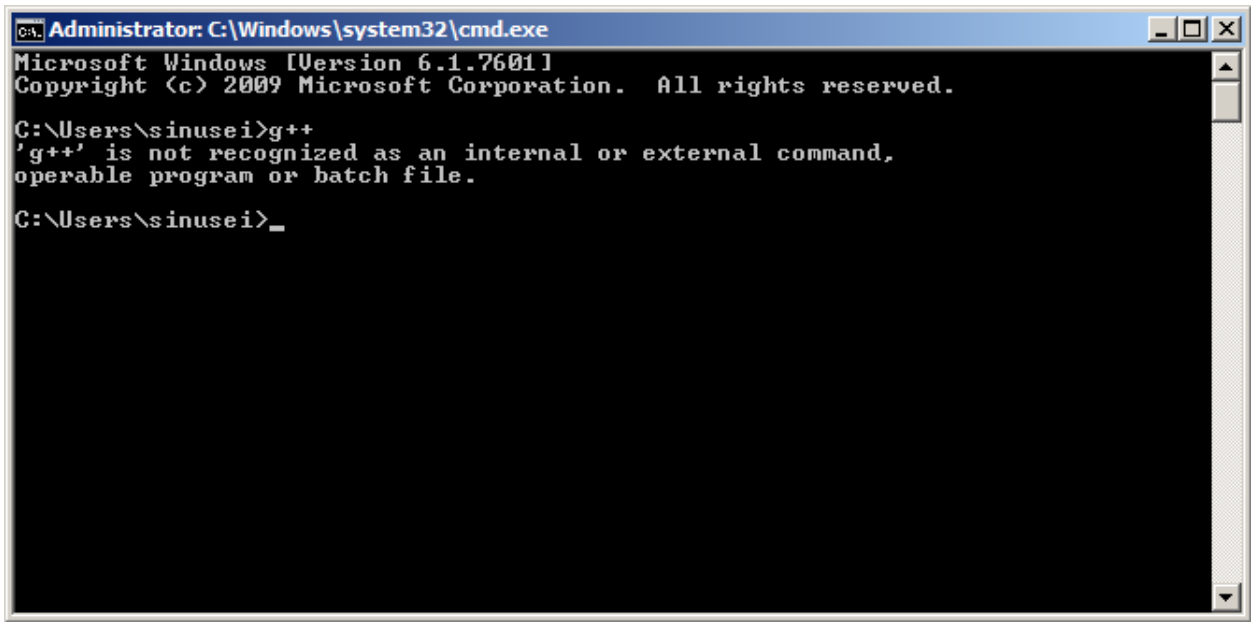
3. REALIZACIJA SUSTAVA

Prije procesa pisanja programskog koda potrebno je konfigurirati svu tehnologiju tako da je prevođenje ili interpretiranje programskog koda uopće moguće.

3.1. Konfiguriranje sustavskog softvera radi omogućavanja rada same web aplikacije

Prije svega, potrebno je omogućiti da se C++ prevoditelj, g++.exe, koristi iz bilo kojeg direktorija. Naime, pomoću PHP-a web aplikacija ima moć pokretati sustavske naredbe u terminalu nadležnog operativnog sustava, cmd.exe. Web aplikacija, dakle, poziva C++ prevoditelj, gcc.exe, za datoteke koje se nalaze u nekim direktorijima koji nisu nužno ranije navedeni direktorij, `MinGW/bin`, direktorij u kojem se nalazi sam prevoditelj, g++.exe.

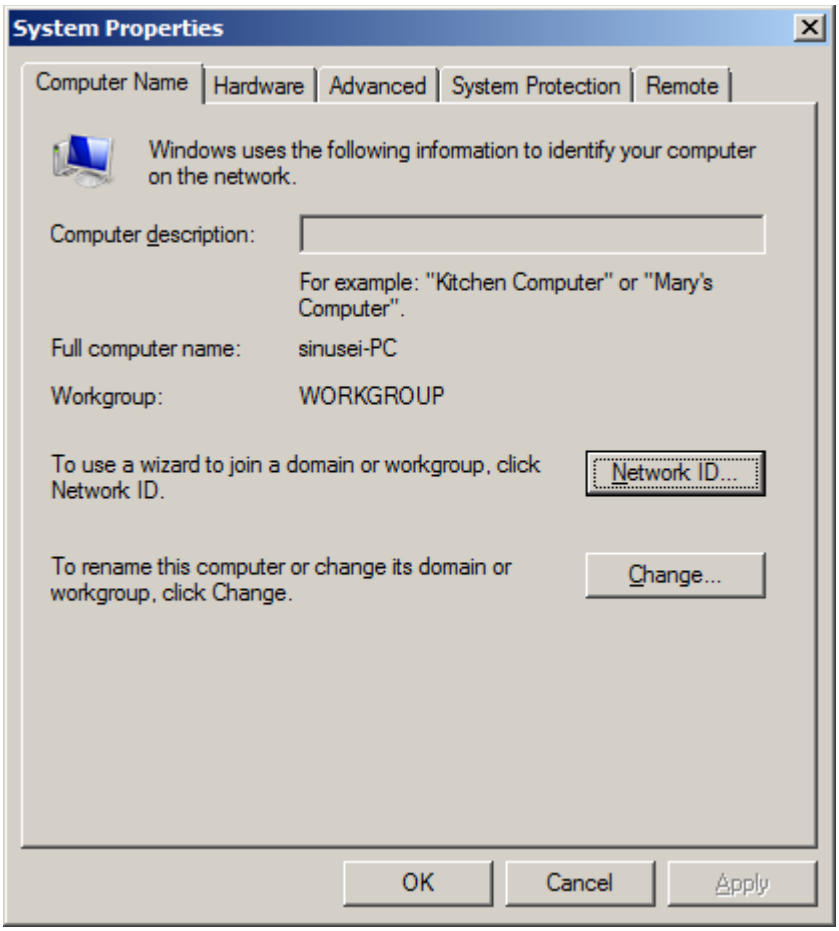
Rezultat pozivanja istog prevoditelja po zadanim postavkama rezultira ovako:



Slika 3.1: Rezultat poziva g++.exe po zadanim postavkama Windows 7 operativnog sustava

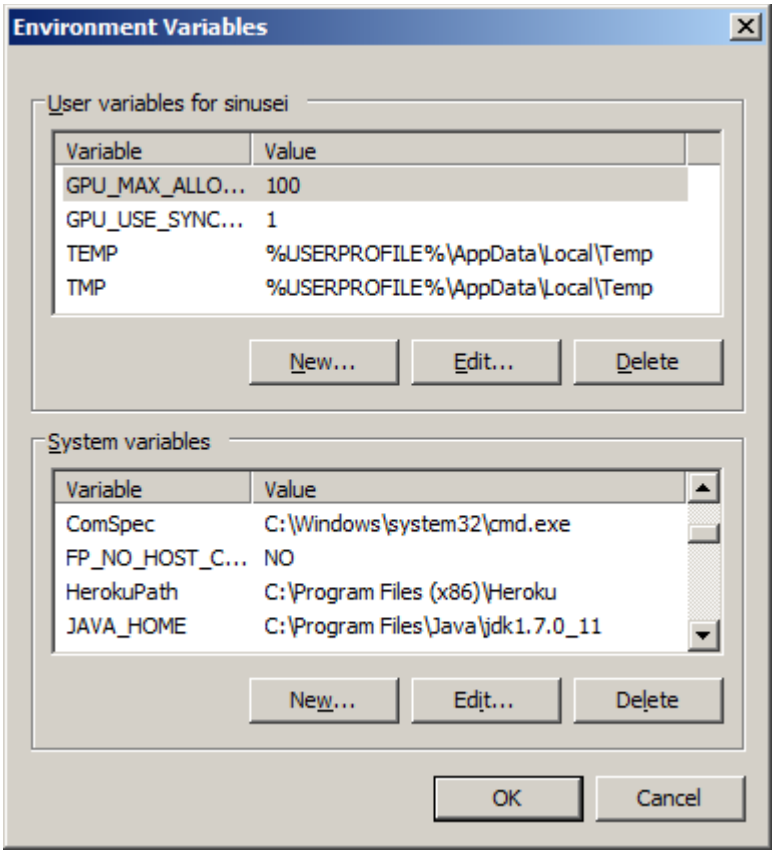
Sukladno slici 3.1, Windows zadanih postavki ne može koristiti C++ prevoditelj iz bilo kojeg radnog direktorija, makar prevoditelj postojao na tvrdom disku. Uz sljedeće je modifikacije moguće sanirati ovaj problem:

Klikom na *Start*, a zatim desnim klikom na *Computer*, odabiranjem opcije *Properties*, a zatim opcije *Advanced system settings* na lijevoj strani novootvorenog prozora, korisniku biva predstavljen sljedeći prozor:



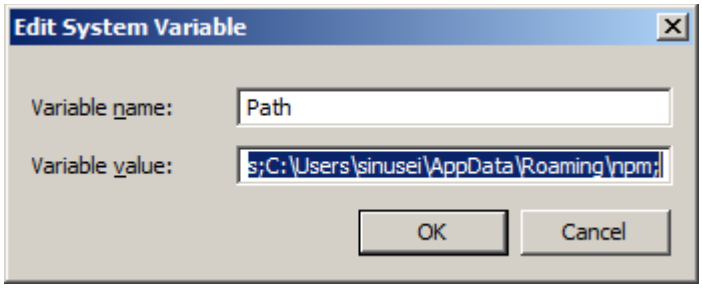
Slika 3.2: „Svojstva sustava“

Odabirom kartice *Advanced*, a zatim opcije *Environment Variables*, korisniku se otvara sljedeći prozor:



Slika 3.3: „Varijable okruženja“

Pronalaskom unosa „Path“ u donjem popisu, a zatim odabirom opcije *Edit*... korisnik dobiva uvid u sljedeći dijalog:

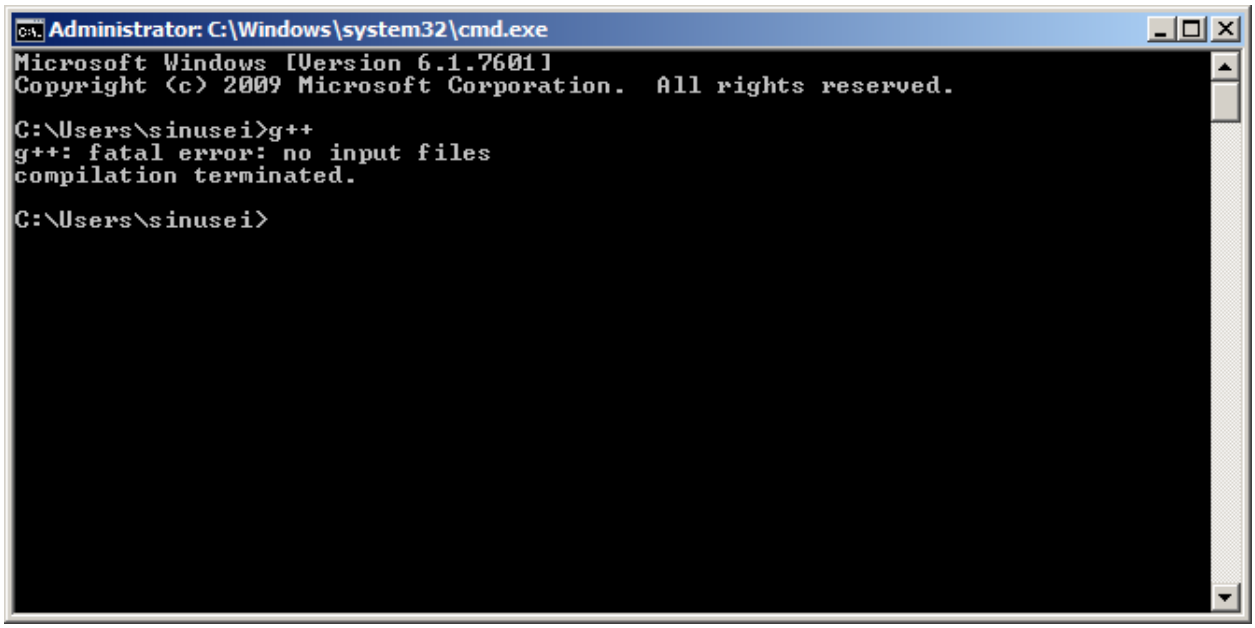


Slika 3.4: Uređivanje pojedine sustavske varijable

Ova sustavska varijabla, varijabla *Path*, definira putanje svih direktorija koji će u budućim instancama terminala biti dostupne iz bilo kojeg direktorija. Za korištenje g++.exe prevoditelja potrebno je na kraj podatka zapisanog u *Variable value* dodati putanju istog. Sukladno lokaciji instalacije C++ prevoditelja na računalu autora, na kraj ove varijable dodana je sljedeća putanja:

```
D:\MinGW\bin;
```

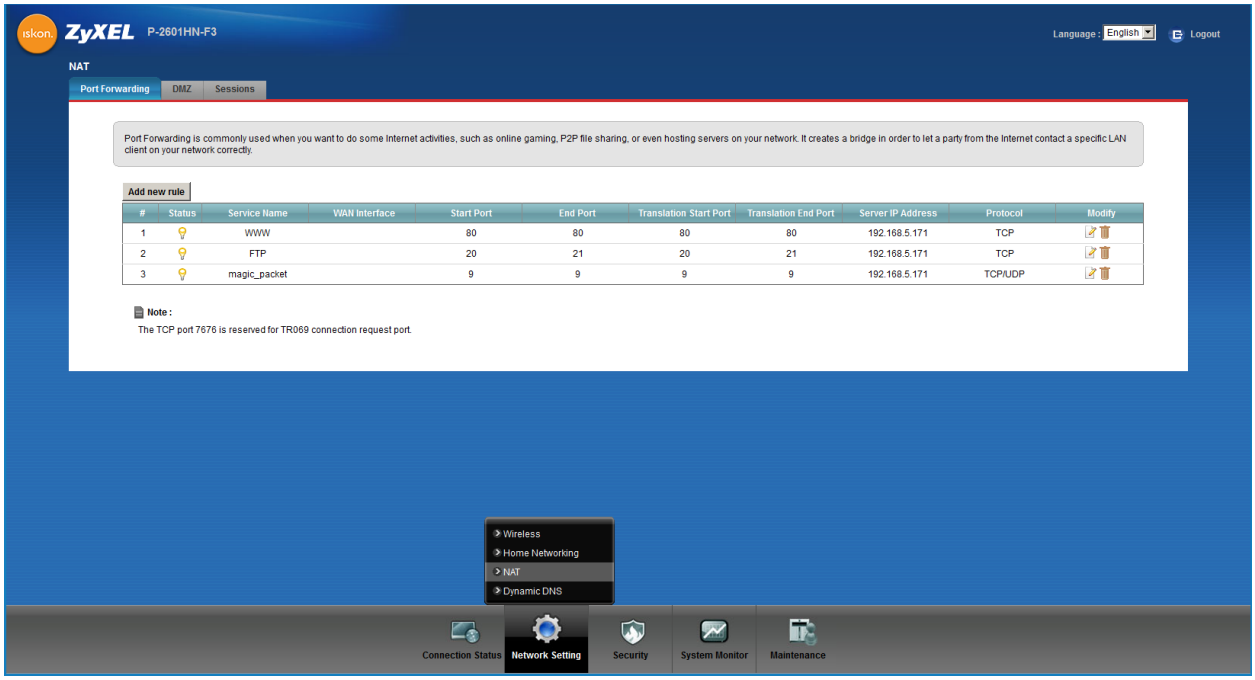
Svaka novootvorena instanca cmd.exe terminala dozvoljavat će korištenje navedenog prevoditelja u bilo kojem folderu:



Slika 3.5: Rezultat poziva g++.exe prevoditelja nakon obavljenog ažuriranja sustavske varijable *Path*

Nadalje, potrebno je konfigurirati web poslužitelj, a isti je proces konfiguriranja već opisan slikama 2.8 i 2.9, kao i njihovim objašnjenjima neposredno nakon istih.

Sljedeću prepreku koju je potrebno eliminirati je činjenica da je računalo koje je spojeno na Internet i dalje u pravilu nedostupno kao poslužitelj, jer je moguće da na njegovoj lokalnoj mreži postoji više računala, a tada sva dijele istu vanjsku IP adresu. Klijent, kontaktirajući istu šalje zahtjev za HTML sadržajem, ali infrastruktura lokalne mreže možda ne posjeduje podatak s kojeg računala je potrebno dohvatiti sadržaj, odnosno kojem računalu proslijediti zahtjev. Procedura „učenja“ iste infrastrukture kojem računalu je potrebno proslijediti zahtjeve s Interneta ovisno o okolnostima zove se *Network Address Translation* (*NAT*). Pojedini koraci procedure izrazito ovise o realizaciji same infrastrukture lokalne mreže, što je prevelika varijabla da se konkretna procedura definira za bilo kojeg programera željnog implementacije ove web aplikacije. Sljedeća procedura definirat će istu proceduru na računalu autora ovog rada:



Slika 3.6: Finalna konfiguracija „prevođenja port-ova“ lokalne mreže.

Računalo autora na lokalnoj je mreži opisano unutarnjom IP adresom 192.168.5.171. Uvođenjem pravila da se svi upiti koji zahtijevaju port 80 proslijede na navedenu IP adresu postiže se ispravna komunikacija web browser-a klijenta i računala poslužitelja – autorova računala.

Potencijalni korisnici web aplikacije za natjecanje u programiranju sada mogu kontaktirati računalo poslužitelj, autorovo računalo, ali samo pomoću brojčane IP adrese. Kako bi isto računalo bilo dostupno putem nekakve simboličke URL adrese, potrebno je prijaviti se na besplatnu *Dynamic Domain Name System* (*DDNS*) uslugu, uslugu pruženu od treće strane. Jedan od takvih servisa danas je Dynu. [22] Prilikom registracije moguće je i poželjno preuzeti *desktop* aplikaciju zaduženu za konstantno ažuriranje vanjske IP adrese računala s informacijama istih o kojima brine tvrtka Dynu, s obzirom na to da su IP adrese većine osobnih računala na prostoru Republike Hrvatske dinamičkog tipa, odnosno doživljavaju izmjene više puta tijekom dana, a minimalno jednom po danu, što bi inače onemogućilo korisniku web aplikacije da u dva različita dana pozivanjem iste adrese dospije na istu web lokaciju. Krajnja aplikacija ponuđena od tvrtke Dynu koja vrši ovu uslugu zove se DynuIUSvc.exe.

U suradnji sa servisom Dynu, simbolička je web adresa autorova računala sljedeća:

tgudelj.dynu.net

Ukoliko računalo na toj adresi i dalje nije dostupno, potrebno ga je prvo izbaciti iz neaktivnog stanja u pripravno stanje slanjem „čarobnog paketa“, standarda osmišljenog za upravo ovu svrhu.

Internet raspolaže besplatnim web aplikacijama za slanje istog paketa, a jedna od njih se nalazi na web stranici www.depicus.com [23]. Da bi se autorovo računalo dovelo u pripravno stanje, potrebno je poslati čarobni paket sa sljedećim parametrima:

MAC address	00-25-22-9A-B3-15
IP address	tgudelj.dynu.net
Subnet mask	255.255.255.255
Port	9

Tablica 3.1: Parametri čarobnog paketa korištenog za aktiviranje računala poslužitelja u pripravno stanje

Ako je računalo dostupno, web aplikacija za natjecanje u programiranju nalazi se na web adresi [24]:

http://tgudelj.dynu.net/

3.2 Realizacija sustava programskim kodom

Iako je cjeloviti programski kod realizacije web aplikacije za natjecanje u programiranju prikazan u poglavu 5. PRILOZI, ovdje će biti navedeni isječci programskog koda koji rješavaju fundamentalne probleme ovog zadatka, potkrijepljeni objašnjenjima.

```
function generate_random_filename(){
	return substr(md5(rand()), 0, 16);
}
```

Navedena PHP funkcija generira nasumičan niz znakova koji odgovara pravilima imenovanja datoteka.

```
function cmd_exec($cmd, &$stdout, &$stderr){
	//Credit: rustleb at php.net, http://php.net/manual/en/function.shell-
exec.php#67183
	$outfile = tempnam(".", "cmd");
	$errfile = tempnam(".", "cmd");
	$descriptorspec = array(
		0 => array("pipe", "r"),
		1 => array("file", $outfile, "w"),
		2 => array("file", $errfile, "w")
	);
	$proc = proc_open($cmd, $descriptorspec, $pipes);
	if (!is_resource($proc))
		return 255;
	fclose($pipes[0]); //Don't really want to give any input
	$exit = proc_close($proc);
	$stdout = file($outfile);//These can dump huge amounts of senseless output
into a temporary file
	$stderr = file($errfile);//These can dump huge amounts of senseless output
into a temporary file
	unlink($outfile);
```

```
        unlink($errfile);
        return $exit;
}
```

Navedena funkcija `cmd_exec()` autorsko je djelo korisnika *rustleb* PHP repozitorija `php.net`, a služi izvršenju sustavske naredbe i razlikovanju standardnog izlaza greške i standardnog izlaza ispisa pozvane naredbe, s ciljem razlikovanja je li C++ programski kod upisan od strane korisnika ispravan ili ne.

```
while(!isset($filenames)){
        $temp = generate_random_filename();
        if(file_exists($temp . ".cpp") || file_exists($temp . ".in") ||
file_exists($temp . ".exe"))
                continue;
        else{
                $filenames = $temp;
                break;
        }
}
```

Navedeni blok koda generira nasumična imena za buduće datoteke sve dok ne generira barem jedno takvo ime kojem trenutno ne postoji duplikat u istom direktoriju. Naime, sav programski kod napisan i poslan od strane korisnika eventualno mora barem nakratko biti spremljen na računalo poslužitelja, što omogućava koliziju u obliku dva istovremena pokušaja spremanja koda pod istim imenom, što bi ishodilo sustavskom greškom spremanja programskog koda. Zato je bilo potrebno osmisliti algoritam imenovanja datoteka takav da spriječava koliziju. Funkcija koja generira ta imena je `generate_random_filename()`.

```
$shell_command = "g++ " . $shell_command_args . " -o " . $filenames . ".exe " .
$filenames . ".cpp && " . $filenames . ".exe < " . $filenames . ".in 2>&1";

if(file_exists($filenames . ".cpp"))
        $output = cmd_exec($shell_command, $stdout, $stderr);
```

Ovo je ključni dio koda. Definiranjem varijable `$shell_command` definira se sustavska naredba koja će se pozivom funkcije `cmd_exec()` biti izvršena, a rezultat je prevođenje C++ programskog koda u izvršnu datoteku.

```
if(isset($_POST["task_selection"])){
                                if(isset($code_to_run) &&
!(strpos(implode($stderr), 'error') !== false)){//assumes that there are
compile-time errors in the code only if 'error' is present in compiler
return, $stderr
                                        echo '<table id="run_results">';
                                        echo "<tr><td>broj test-primjera</td>";
                                        echo "<td>izlaz autorovog
programa</td></tr>";
                                        for($i = 1; $i <= 10; $i++){
                                                $shell_command = $filenames . ".exe
< ./tasks/".$_POST["task_selection"]."/in_" . $i . ".in 2>&1";
```

```
                                                cmd_exec($shell_command, $stdout,
$stderr);// or die("shell command execution fail");
                                                if(Implode($stdout) ==
file_get_contents("./tasks/".$_POST["task_selection"]."/out_" . $i . ".out"))
                                                      $points++;
                                                if(count($stdout) == 0){
                                                      if(isset($stdout[$i]))
                                                            echo
"<tr><td>".$i."</td><td>".$stdout[$j]."</td></tr>";
                                                      else
                                                            echo
"<tr><td>".$i."</td><td></td></tr>";
                                                }
                                                else{
                                                      echo
"<tr><td>".$i."</td><td>";
                                                for($j = 0; $j < count($stdout);
$j++)
                                                      if(isset($stdout[$j]))
                                                            echo $stdout[$j]."<br
/>";
                                                echo "</td></tr>";
                                                }
                                          }
                                          echo "</table>";
                                    }
                              }
                              else{
                                    echo '<div id="no_task_output"><pre>';
                                    if(isset($stdout))
                                          for($i = 0; $i < count($stdout); $i++)
                                                echo $stdout[$i];
                                    echo "</pre></div>";
                              }
                              if(file_exists($filenames . ".cpp"))
                                    unlink($filenames . ".cpp"); //delete .cpp
                              if(file_exists($filenames . ".in"))
                                    unlink($filenames . ".in"); //delete .in
                              if(file_exists($filenames . ".exe"))
                                    unlink($filenames . ".exe"); //delete .exe
                        ?>
                  echo "run number ". $i . ": " . Implode($stdout);
                  echo "<br />";
            }
      }
```

Ovaj blok koda u suštini pokreće C++ programski kod korisnika onoliko puta koliko postoji predefiniranih test-primjera, a povećavanjem varijable `$points` pamti točnost izlaza korisnikova programa na pojedine test-primjere.

Na sučelju za programiranje nalaze se dvije forme, forma u kojoj se upisuje kod, standardni ulaz, argumenti kompilacije i korisnikovo ime, ali i forma u obliku padajućeg menija. Posljednje je bilo potrebno jer je tekst zadatka potrebno dohvatiti od poslužitelja tek kada korisnik odabere zadatak. Ovo predstavlja problem jer je zadana funkcija PHP jezika da u sljedeću sesiju šalje samo podatke poslane forme koristeći globalne varijable, poput varijable `$_POST`, a potrebno je

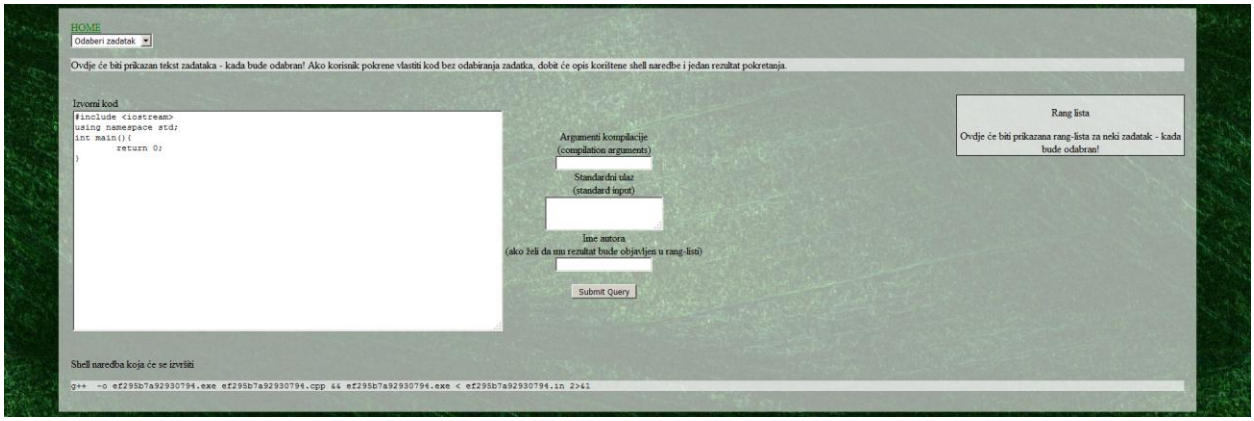
zapamtiti koji zadatak je korisnik odabrao. U ovu svrhu, iskorišten je sljedeći PHP kod u tijelu prve navedene forme:

```
<?php
                                    if(isset($_POST["task_selection"]))
                                        echo '<input type="hidden"
name="task_selection" value="'.$_POST["task_selection"].'" />';
                                ?>
```

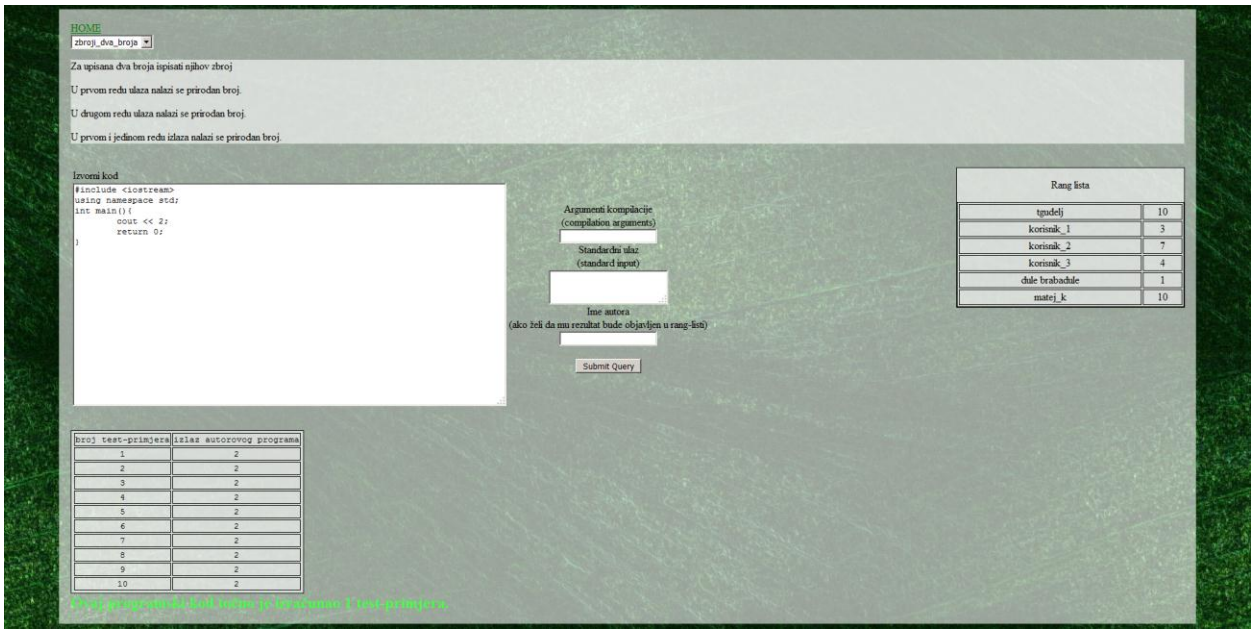
Naime, u slučaju već odabranog zadatka ovaj PHP kod u prvu navedenu formu uvodi još jedan `<input>` element, element koji služi prikupljanju podataka u formi, ali nije prikazan niti dostupan korisniku. Njegova vrijednost postavljena je na istu vrijednost padajućeg menija, pa prva navedena forma šalje u sljedeću sesiju i taj podatak što omogućava da on bude zapamćen i dostupan, makar je potekao iz ranije istekle sesije.

Slika 3.7: Korisničko sučelje web aplikacije za natjecanje u programiranju

Klikom na poveznicu „Programiraj!“ na slici 3.7, korisnik dospijeva na sljedeće:

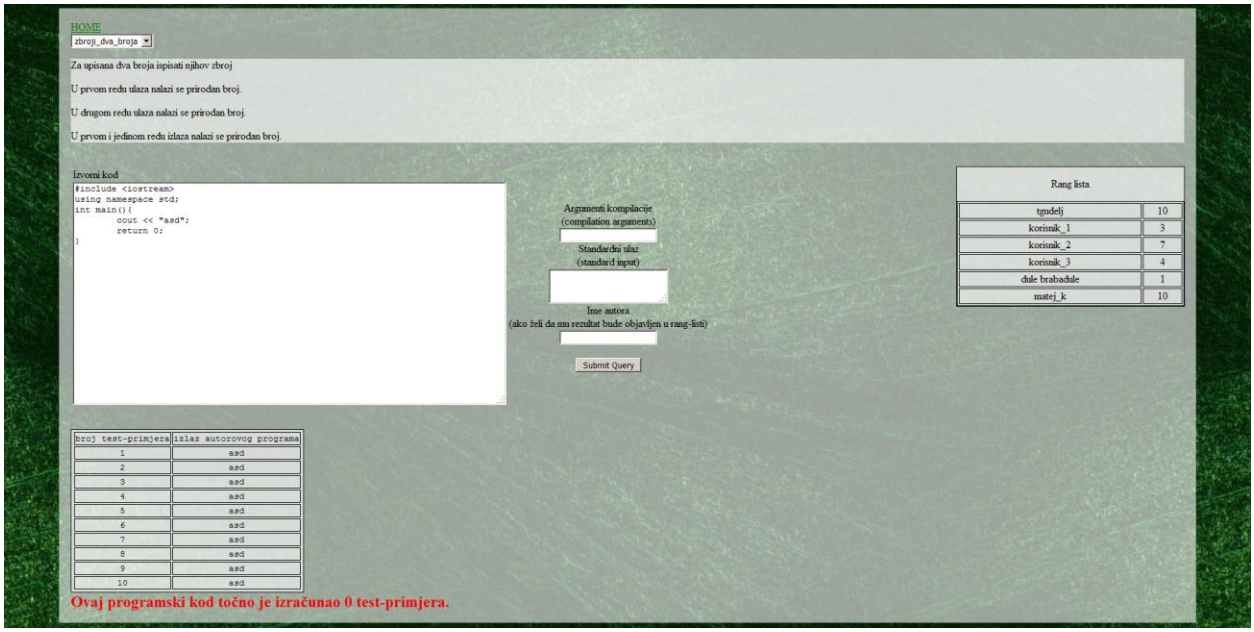


Slika 3.8: Korisničko sučelje web aplikacije za natjecanje u programiranju, sučelje za programiranje, bez odabranog zadatka



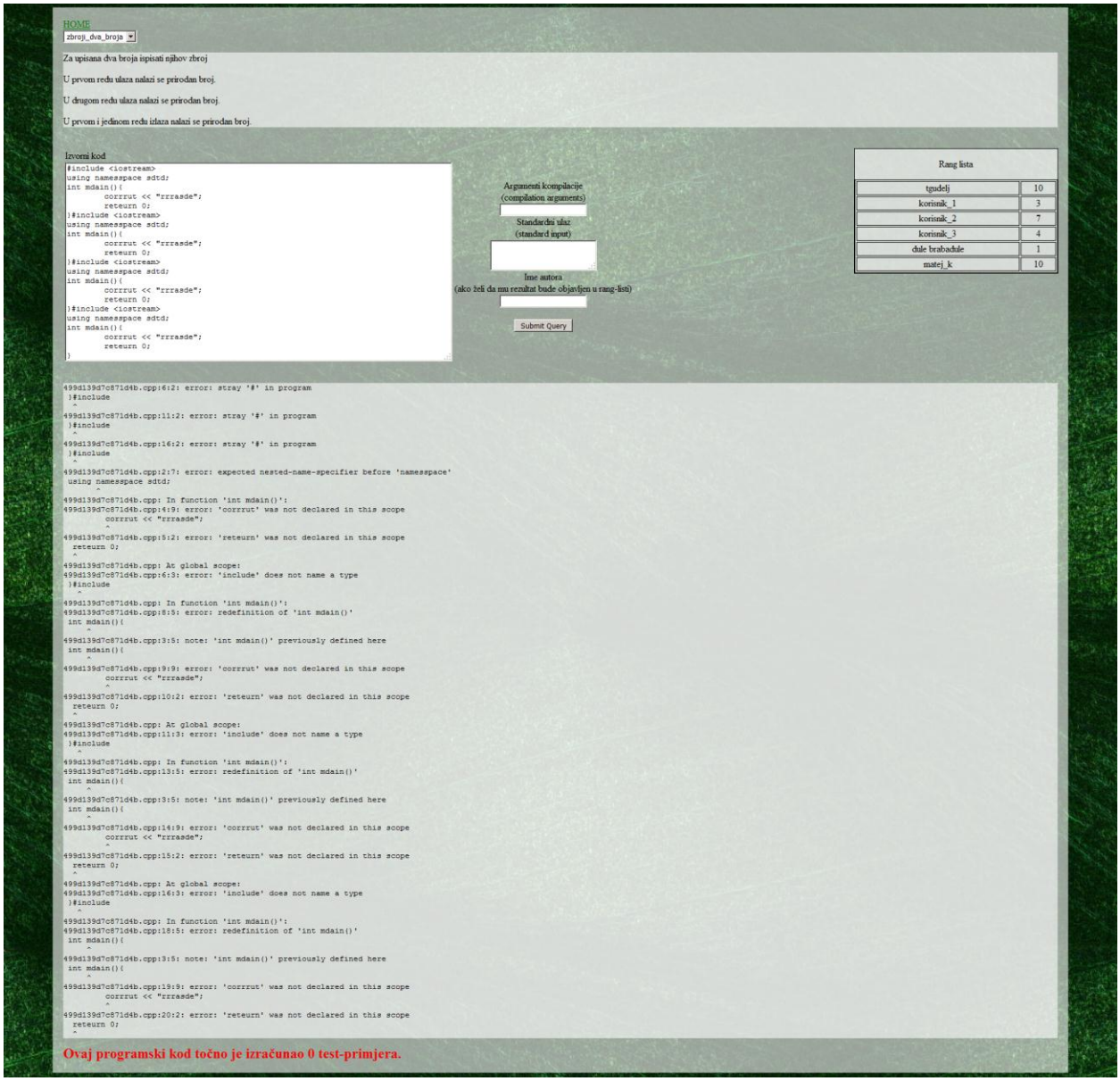
Slika 3.9: Rezultat korištenja web aplikacije s napisanim programskim C++ kodom koji pokušava pogoditi jedan od test primjera nekog odabranog zadatka i to uspješno

Na slici 3.9 demonstrirana je funkcionalnost web aplikacije da ispravno prebrojava broj točnih test-primjera vraćenih od strane programa. Naime, igrom slučaja odabrani zadatak ima samo jedan očekivani izlaz jednak onom koji je vraćen od strane programa *brute-force* metodom - metodom pogađanja.



Slika 3.10: Rezultat korištenja web aplikacije s napisanim programskim C++ kodom koji pokušava pogoditi jedan od test primjera nekog odabranog zadatka i to bezuspješno

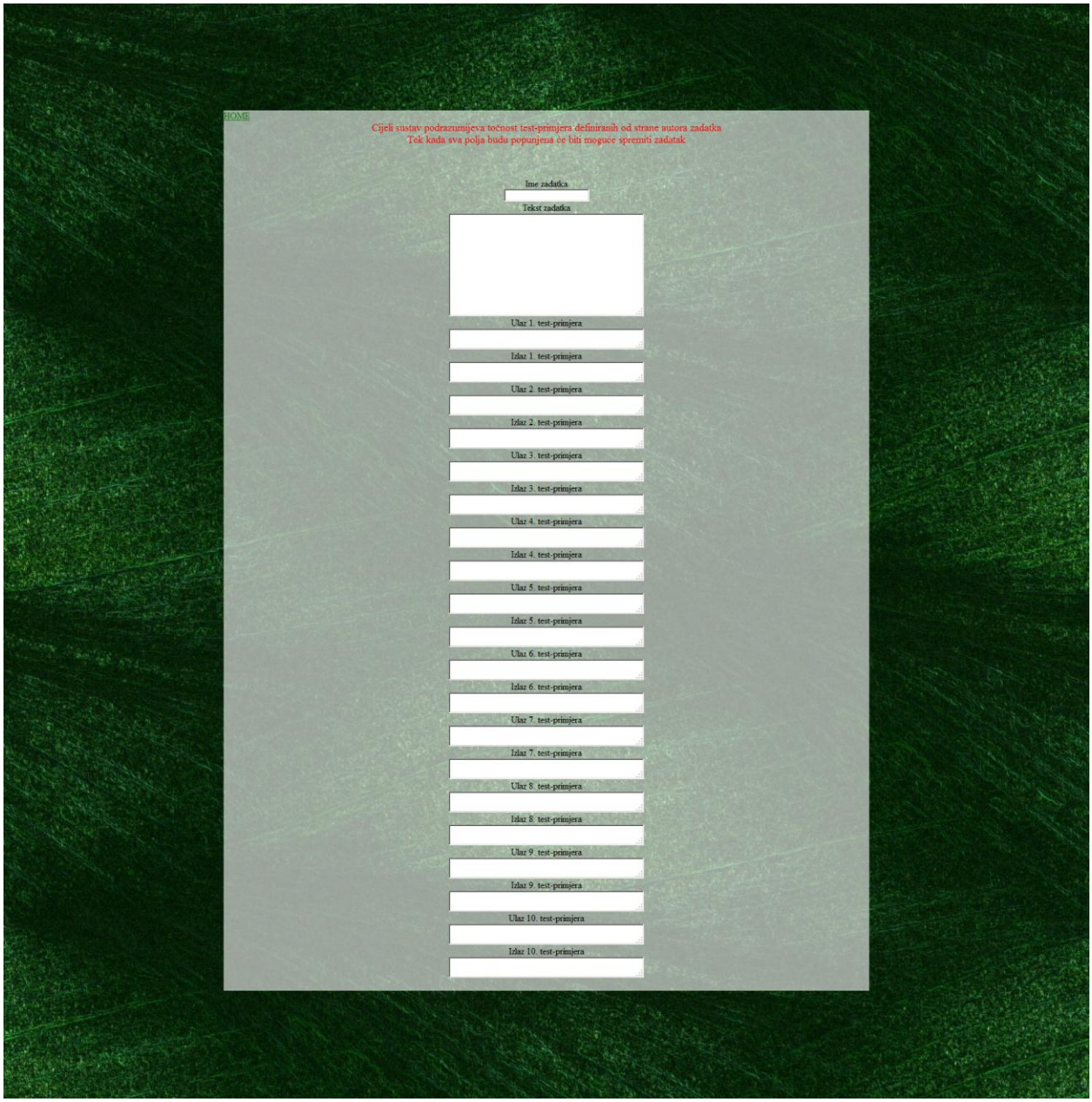
Kao i na slici 3.9, slika 3.10 demonstrira testiranje korisnikova koda na točnost provjeravajući postojeće predefinirane test primjere.



Slika 3.11: Rezultat korištenja web aplikacije sa sintaktički neispravnim programskim kodom

Na slici 3.11 demonstrirana je funkcionalnost web aplikacije da prijavljuje sintaktičke greške C++ programskog koda, baš kao da je on prevođen od strane nekog okruženja za programiranje.

Klikom na poveznicu „Dizajniraj vlastiti zadatak!“ na slici 3.7, korisnik dospijeva na sljedeće:



Slika 3.12: Korisničko sučelje web aplikacije za natjecanje u programiranju, sučelje za dodavanje zadatka

Slika 3.13: Postupak kreiranja novog zadatka koji će trajno biti spremljen u sustav

4. ZAKLJUČAK

U ovom završnom radu napravljena je web aplikacija za natjecanje u programiranju, web stranica koja omogućava korisniku rješavanje programskog problema dizajniranjem C++ programa posebne namjene, a s namjerom rješavanja baš tog programskog problema. Aplikacija od korisnikova programa očekuje nekakve izračune, a na temelju jednakosti istih i unaprijed definiranih „test-primjera“, skupina parova ulaz-izlaz, korisniku je dodijeljen konačan broj bodova. Aplikacija, dakle, unaprijed zna kakvi rezultati izračuna su točni rezultati i istim ulazim podacima podvrgava i korisnikov program. Čak i uz unaprijed poznatu netočnost izračuna, korisnik koristeći ovu aplikaciju može otkriti ima li u njegovom kodu sintaktičkih grešaka. Korisnik ima opciju navesti i svoje ime, ako želi da njegov rezultat bude evidentiran u javno vidljivoj rang-listi (engl. *scoreboard*), čime se kreira kompetitivna atmosfera.

5. PRILOZI

5.1. PRILOG cpp_programming_competition.php

```
<html>
	<?php
		function generate_random_filename(){
			return substr(md5(rand()), 0, 16);
		}
		function cmd_exec($cmd, &$stdout, &$stderr){
			//Credit: rustleb at php.net,
http://php.net/manual/en/function.shell-exec.php#67183
			$outfile = tempnam(".", "cmd");
			$errfile = tempnam(".", "cmd");
			$descriptorspec = array(
				0 => array("pipe", "r"),
				1 => array("file", $outfile, "w"),
				2 => array("file", $errfile, "w")
			);
			$proc = proc_open($cmd, $descriptorspec, $pipes);
			if (!is_resource($proc))
				return 255;
			fclose($pipes[0]); //Don't really want to give any input
			$exit = proc_close($proc);
			$stdout = file($outfile);//These can dump huge amounts of
senseless output into a temporary file
			$stderr = file($errfile);//These can dump huge amounts of
senseless output into a temporary file
			unlink($outfile);
			unlink($errfile);
			return $exit;
		}
		if(array_key_exists("code_to_run", $_POST))
			$code_to_run = $_POST["code_to_run"];
		if(array_key_exists("compilation_args", $_POST))
			$shell_command_args = $_POST["compilation_args"];
		if(array_key_exists("standard_input", $_POST))
			$standard_input = $_POST["standard_input"];
		if(array_key_exists("tester_name", $_POST))
			$tester_name = $_POST["tester_name"];
	?>
	<head>
		<link rel="stylesheet" type="text/css"
href="cpp_programming_competition.css">
		<title>Natjecanje u programiranju</title>
	</head>
	<body>
		<div id="main_container">
			<a href=".">HOME</a><br />
			<form id="task_selection_form"
action="cpp_programming_competition.php" method="POST">
				<select id="task_selection" name="task_selection"
onChange="task_selected();">
					<option <?php
if(!isset($_POST["task_selection"])) echo "selected"; ?> disabled>Odaberi
zadatak</option>
					<?php
						$scandir_result = scandir("./tasks");
```

```
                                    for($i = 2; $i < count($scandir_result);
$i++){//$i=0 and $i=1 are . and ..
                                        if(isset($_POST["task_selection"])){
                                            if($_POST["task_selection"] ==
$scandir_result[$i])
                                                echo "<option
selected>".$scandir_result[$i]."</option>";
                                            else
                                                echo
"<option>".$scandir_result[$i]."</option>";
                                        }
                                        else
                                            echo
"<option>".$scandir_result[$i]."</option>";
                                    }
                                ?>
                            </select>
                        </form>
                        <div id="problem_text">
                            <p><?php
                                if(isset($_POST["task_selection"])){

        if(file_exists("./tasks/".$_POST["task_selection"]."/task_text")){
                                        for($i = 0; $i <
count(file("./tasks/".$_POST["task_selection"]."/task_text")); $i++)
                                        echo
"<p>".file("./tasks/".$_POST["task_selection"]."/task_text")[$i]."</p>";
                                    }
                                }
                                else
                                    echo "Ovdje će biti prikazan tekst
zadataka - kada bude odabran! Ako korisnik pokrene vlastiti kod bez
odabiranja zadatka, dobit će opis korištene shell naredbe i jedan rezultat
pokretanja.";
                                ?>
                            </p>
                        </div>
                        <br />
                        <div id="scoreboard">
                            <p>Rang lista</p>
                            <table id="scoreboard_table">
                                <?php
                                    if(isset($_POST["task_selection"])){

        if(file_exists("./tasks/".$_POST["task_selection"]."/scoreboard")){
                                            $scoreboard =
file("./tasks/".$_POST["task_selection"]."/scoreboard");
                                            for($i = 0; $i <
count($scoreboard); $i += 2){
                                                echo
"<tr><td>".$scoreboard[$i]."</td>";
                                                echo
"<td>".$scoreboard[$i + 1]."</td></tr>";
                                            }
                                        }
                                    }
                                    else
                                        echo "Ovdje će biti prikazana rang-
lista za neki zadatak - kada bude odabran!";
                                ?>
                            </table>
```

```
                </div>
                <table id="form_container">
                        <form action="cpp_programming_competition.php"
method="post">
                                <?php
                                        if(isset($_POST["task_selection"]))
                                                echo '<input type="hidden"
name="task_selection" value="'.$_POST["task_selection"].'" />';
                                ?>
                                <tr>
                                        <td>
                                                <span>Izvorni kod</span>
                                                <br />
                                                <textarea name="code_to_run"
id="code_to_run" rows="20" cols="80" onkeydown="return catchTab(this,event)"
spellcheck="false"><?php
                                                        if(isset($code_to_run))
                                                                echo $code_to_run;
                                                        else{
                                                                echo "#include
<iostream>\n";
                                                                echo "using namespace
std;\n";
                                                                echo "int main(){\n";
                                                                echo "      return 0;\n";
                                                                echo "}";
                                                        }
                                                ?></textarea>
                                        </td>
                                        <td
id="form_without_source_code_textarea">
                                                <div>
                                                        <span>Argumenti
kompilacije</span><br />
                                                        <span>(compilation
arguments)</span><br />
                                                        <input type="text"
id="compilation_args" name="compilation_args" />
                                                </div>
                                                <div>
                                                        <span>Standardni
ulaz</span><br />
                                                        <span>(standard
input)</span><br />
                                                        <textarea id="standard_input"
name="standard_input"></textarea>
                                                </div>
                                                <div>
                                                        <span>Ime autora</span><br />
                                                        <span>(ako želi da mu rezultat
bude objavljen u rang-listi)</span><br />
                                                        <input type="text"
id="tester_name" name="tester_name" />
                                                <br />
                                                <br />
                                                <input type="submit" />
                                                </div>
                                        </td>
                                </tr>
                        </form>
                </table>
```

```
                    <br />
                    <?php
                        //echo "<p>Received code
is:<br/>".$_POST["code_to_run"];
                        while(!isset($filenames)){
                            $temp = generate_random_filename();
                            if(file_exists($temp . ".cpp") ||
file_exists($temp . ".in") || file_exists($temp . ".exe"))
                                continue;
                            else{
                                $filenames = $temp;
                                break;
                            }
                        }
                        if(isset($standard_input))
                            file_put_contents($filenames.".in" ,
$standard_input);
                        if(isset($code_to_run))
                            file_put_contents($filenames.".cpp" ,
$code_to_run);
                        if(!isset($shell_command_args))
                            $shell_command_args = "";
                        if(!isset($_POST["task_selection"]))
                            echo "<p>Shell naredba koja će se izvršiti</p>";
                        $shell_command = "g++ " . $shell_command_args . " -o "
. $filenames . ".exe " . $filenames . ".cpp && " . $filenames . ".exe < " .
$filenames . ".in 2>&1";
                        if(!isset($_POST["task_selection"]))
                            echo "<div
id='shell_command'><pre>$shell_command</pre></div>";
                        if(file_exists($filenames . ".cpp"))
                            $output = cmd_exec($shell_command, $stdout,
$stderr);// or die("shell command execution fail");
                        if(isset($output)){
                            echo '<div id="stderr"><pre>';
                            /*for($i = 0; $i < count($stdout); $i++)
                                echo $stdout[$i];*/
                            for($i = 0; $i < count($stderr); $i++)
                                echo $stderr[$i];
                            echo "</pre></div>";
                            //echo "<pre>" . $stdout . $stderr . "</pre>";
                        }
                        $points = 0;
                        if(isset($_POST["task_selection"])){
                            if(isset($code_to_run) &&
!(strpos(implode($stderr), 'error') !== false)){//assumes that there are
compile-time errors in the code only if 'error' is present in compiler
return, $stderr
                                echo '<table id="run_results">';
                                echo "<tr><td>broj test-primjera</td>";
                                echo "<td>izlaz autorovog
programa</td></tr>";
                                for($i = 1; $i <= 10; $i++){
                                    $shell_command = $filenames . ".exe
< ./tasks/".$_POST["task_selection"]."/in_" . $i . ".in 2>&1";
                                    cmd_exec($shell_command, $stdout,
$stderr);// or die("shell command execution fail");
                                    if(Implode($stdout) ==
file_get_contents("./tasks/".$_POST["task_selection"]."/out_" . $i . ".out"))
                                        $points++;
                                    if(count($stdout) == 0){
```

```
                                                if(isset($stdout[$i]))
                                                      echo
"<tr><td>".$i."</td><td>".$stdout[$j]."</td></tr>";
                                                else
                                                      echo
"<tr><td>".$i."</td><td></td></tr>";
                                                }
                                                else{
                                                      echo
"<tr><td>".$i."</td><td>";
                                                for($j = 0; $j < count($stdout);
$j++)
                                                if(isset($stdout[$j]))
                                                      echo $stdout[$j]."<br
/>";
                                                echo "</td></tr>";
                                                }
                                          }
                                          echo "</table>";
                                    }
                              }
                              else{
                                    echo '<div id="no_task_output"><pre>';
                                    if(isset($stdout))
                                          for($i = 0; $i < count($stdout); $i++)
                                                echo $stdout[$i];
                                    echo "</pre></div>";
                              }
                              if(file_exists($filenames . ".cpp"))
                                    unlink($filenames . ".cpp"); //delete .cpp
                              if(file_exists($filenames . ".in"))
                                    unlink($filenames . ".in"); //delete .in
                              if(file_exists($filenames . ".exe"))
                                    unlink($filenames . ".exe"); //delete .exe
                        ?>
                        <?php
                              if(isset($code_to_run) &&
isset($_POST["task_selection"]))
                                    if($points == 0)
                                          echo '<span id="you_scored_nothing">Ovaj
programski kod točno je izračunao '.$points.' test-primjera.</span>';
                                    else
                                          echo '<span id="you_scored_something">Ovaj
programski kod točno je izračunao '.$points.' test-primjera.</span>';
                              if(isset($tester_name))
                                    if($points && $tester_name){

      file_put_contents("./tasks/".$_POST["task_selection"]."/scoreboard",
"\n".$tester_name, FILE_APPEND);

      file_put_contents("./tasks/".$_POST["task_selection"]."/scoreboard",
"\n".$points, FILE_APPEND);
                                    }
                        ?>
            </div>
      </body>
</html>
<script>
      function task_selected(){
            document.getElementById("task_selection_form").submit();
      }
```

```
</script>
```

5.2. PRILOG cpp_programming_competition.css

```
body{
	background-color: black;
	background-image:
url("http://www.publicdomainpictures.net/pictures/130000/velka/green-smooth-
seamless-background.jpg");
}
#main_container{
	width: 90%;
	margin: 0 auto;
	padding: 1%;
	background-color: rgba(255, 255, 255, 0.6);
	background-blend-mode: multiply;
	position: relative;
}
#scoreboard{
	width: 20%;
	text-align: center;
	position: absolute;
	right: 1%;
	border: 1px solid black;
}
#scoreboard_table{
	width: 100%;
	text-align: center;
	border: 1px solid black;
}
#scoreboard_table td{
	border: 1px solid black;
}
#run_results{
	text-align: center;
	border: 1px solid black;
	padding: 0.1em;
}
#run_results td{
	border: 1px solid black;
	font-family: monospace;
}
#you_scored_nothing{
	font-weight: bold;
	font-size: 24;
	color: red;
	text-align: right;
	position: relative;
}
#you_scored_something{
	font-weight: bold;
	font-size: 24;
	color: #44FF44;
	text-align: right;
	position: relative;
}
#form_without_source_code_textarea > div{
	text-align: center;
}
#problem_text, #scoreboard, #run_results{
	background-color: rgba(255, 255, 255, 0.6);
```

```
}
#stderr, #shell_command, #no_task_output{
	background-color: rgba(255, 255, 255, 0.6);
}
a:link{
	color: #003300;
}
a:hover{
	color: #00BB00;
}
a:visited{
	color: #007700;
}
a:active{
	color: #00DD00;

}
```

5.3. PRILOG index.php

```
<html>
	<head>
		<link rel="stylesheet" type="text/css" href="index.css">
		<title>Natjecanje u programiranju - home</title>
	</head>
	<body>
		<div id="main_container">
			<div id="choose_action">
				<div class="asd" id="asd_1">
				<a href="cpp_programming_competition.php">
					<div id="select_programming">
						<span>Programiraj!</span>
					</div>
				</a>
				</div>
				<div class="asd" id="asd_2">
				<a href="design_own_task.php">
					<div id="select_design_own_task">
						<span>Dizajniraj vlastiti zadatak!</span>
					</div>
				</a>
				</div>
			</div>
		</div>
	</body>

</html>
```

5.4. PRILOG index.css

```
body{
	background-color: black;
	background-image:
url("http://www.publicdomainpictures.net/pictures/130000/velka/green-smooth-
seamless-background.jpg");
}
#main_container{
	width: 40%;
	height: 50%;
	margin: 10% auto;
	background-color: rgba(255, 255, 255, 0.6);
	background-blend-mode: multiply;
}
#choose_action{
	height: 100%;
	width: 100%;
	text-align: center;
	font-size: 40;
}
.asd{
	height: 50%;
	width: 100%;
	margin: 0% auto;
}
#select_programming, #select_design_own_task{
	height: 90%;
	width: 90%;
	position: relative; /*stupidest workaround ever*/
	top: 6.66%;
	margin: 0 auto;
	display: table;
	background-color: rgba(255, 255, 255, 0.6);
	background-blend-mode: multiply;
}
#select_design_own_task{
	top: 3.33%;
}
#select_programming > span, #select_design_own_task > span{
	display: table-cell;
	vertical-align: middle;
}
#choose_action > div > a:link{
	color: #003300;
}
#choose_action > div > a:hover{
	color: #00BB00;
}
#choose_action > div > a:visited{
	color: #007700;
}
#choose_action > div > a:active{
	color: #00DD00;

}
```

5.5. PRILOG design_own_task.php

```
<script>
	function check_for_blank_field(){
		if(
			document.getElementById("task_name").value.length < 1 ||
			document.getElementById("task_text").value.length < 1 ||
			document.getElementById("in_1").value.length < 1 ||
			document.getElementById("out_1").value.length < 1 ||
			document.getElementById("in_2").value.length < 1 ||
			document.getElementById("out_2").value.length < 1 ||
			document.getElementById("in_3").value.length < 1 ||
			document.getElementById("out_3").value.length < 1 ||
			document.getElementById("in_4").value.length < 1 ||
			document.getElementById("out_4").value.length < 1 ||
			document.getElementById("in_5").value.length < 1 ||
			document.getElementById("out_5").value.length < 1 ||
			document.getElementById("in_6").value.length < 1 ||
			document.getElementById("out_6").value.length < 1 ||
			document.getElementById("in_7").value.length < 1 ||
			document.getElementById("out_7").value.length < 1 ||
			document.getElementById("in_8").value.length < 1 ||
			document.getElementById("out_8").value.length < 1 ||
			document.getElementById("in_9").value.length < 1 ||
			document.getElementById("out_9").value.length < 1 ||
			document.getElementById("in_10").value.length < 1 ||
			document.getElementById("out_10").value.length < 1
		)
			document.getElementById("submit_task").style.visibility =
"hidden";
		else
			document.getElementById("submit_task").style.visibility =
"visible";
	}
</script>
<html>
	<head>
		<link rel="stylesheet" type="text/css" href="design_own_task.css">
		<title>Natjecanje u programiranju - izradi vlastiti
zadatak</title>
	</head>
	<body>
		<div id="main_container">
			<a href=".">HOME</a>
			<p>Cijeli sustav podrazumijeva točnost test-primjera
definiranih od strane autora zadatka</p>
			<p>Tek kada sva polja budu popunjena će biti moguće spremiti
zadatak</p>
			<?php
				if(
					isset($_POST["task_name"]) &&
					isset($_POST["task_text"]) &&
					isset($_POST["in_1"]) &&
					isset($_POST["in_2"]) &&
					isset($_POST["in_3"]) &&
					isset($_POST["in_4"]) &&
					isset($_POST["in_5"]) &&
					isset($_POST["in_6"]) &&
					isset($_POST["in_7"]) &&
					isset($_POST["in_8"]) &&
```

```
                        isset($_POST["in_9"]) &&
                        isset($_POST["in_10"]) &&
                        isset($_POST["out_1"]) &&
                        isset($_POST["out_2"]) &&
                        isset($_POST["out_3"]) &&
                        isset($_POST["out_4"]) &&
                        isset($_POST["out_5"]) &&
                        isset($_POST["out_6"]) &&
                        isset($_POST["out_7"]) &&
                        isset($_POST["out_8"]) &&
                        isset($_POST["out_9"]) &&
                        isset($_POST["out_10"])
                        ){
                        $task_name = $_POST["task_name"];
                        $task_text = $_POST["task_text"];
                        $in_1 = $_POST["in_1"];
                        $in_2 = $_POST["in_2"];
                        $in_3 = $_POST["in_3"];
                        $in_4 = $_POST["in_4"];
                        $in_5 = $_POST["in_5"];
                        $in_6 = $_POST["in_6"];
                        $in_7 = $_POST["in_7"];
                        $in_8 = $_POST["in_8"];
                        $in_9 = $_POST["in_9"];
                        $in_10 = $_POST["in_10"];
                        $out_1 = $_POST["out_1"];
                        $out_2 = $_POST["out_2"];
                        $out_3 = $_POST["out_3"];
                        $out_4 = $_POST["out_4"];
                        $out_5 = $_POST["out_5"];
                        $out_6 = $_POST["out_6"];
                        $out_7 = $_POST["out_7"];
                        $out_8 = $_POST["out_8"];
                        $out_9 = $_POST["out_9"];
                        $out_10 = $_POST["out_10"];
                        if(!in_array($task_name, scandir("./tasks"))){
                              mkdir("./tasks/".$task_name);

      file_put_contents("./tasks/".$task_name."/task_text" , $task_text);

      file_put_contents("./tasks/".$task_name."/in_1.in" , $in_1);

      file_put_contents("./tasks/".$task_name."/in_2.in" , $in_2);

      file_put_contents("./tasks/".$task_name."/in_3.in" , $in_3);

      file_put_contents("./tasks/".$task_name."/in_4.in" , $in_4);

      file_put_contents("./tasks/".$task_name."/in_5.in" , $in_5);

      file_put_contents("./tasks/".$task_name."/in_6.in" , $in_6);

      file_put_contents("./tasks/".$task_name."/in_7.in" , $in_7);

      file_put_contents("./tasks/".$task_name."/in_8.in" , $in_8);

      file_put_contents("./tasks/".$task_name."/in_9.in" , $in_9);

      file_put_contents("./tasks/".$task_name."/in_10.in" , $in_10);

      file_put_contents("./tasks/".$task_name."/out_1.out" , $out_1);
```

```
            file_put_contents("./tasks/".$task_name."/out_2.out" , $out_2);
            file_put_contents("./tasks/".$task_name."/out_3.out" , $out_3);
            file_put_contents("./tasks/".$task_name."/out_4.out" , $out_4);
            file_put_contents("./tasks/".$task_name."/out_5.out" , $out_5);
            file_put_contents("./tasks/".$task_name."/out_6.out" , $out_6);
            file_put_contents("./tasks/".$task_name."/out_7.out" , $out_7);
            file_put_contents("./tasks/".$task_name."/out_8.out" , $out_8);
            file_put_contents("./tasks/".$task_name."/out_9.out" , $out_9);
            file_put_contents("./tasks/".$task_name."/out_10.out" , $out_10);
            file_put_contents("./tasks/".$task_name."/scoreboard" , "");
                                    echo '<p id="saving_outcome">Zadatak
uspješno spremljen!</p>';
                                }
                                else
                                    echo '<p id="saving_outcome">Zadatak istog
imena već postoji</p>';
                            }

                        ?>
                        <div id="form_container">
                            <form action="design_own_task.php" method="post">
                                <span>Ime zadatka</span><br />
                                <input type="text" id="task_name"
name="task_name" value="<?php if(isset($_POST["task_name"])) echo
$_POST["task_name"];?>"/><br />
                                <span>Tekst zadatka</span><br />
                                <textarea id="task_text" name="task_text"
rows="10" cols="40"><?php if(isset($_POST["task_text"])) echo
$_POST["task_text"];?></textarea><br />
                                <span>Ulaz 1. test-primjera</span><br />
                                <textarea id="in_1" name="in_1" class="inputs"
rows="1" cols="40"><?php if(isset($_POST["in_1"])) echo
$_POST["in_1"];?></textarea><br />
                                <span>Izlaz 1. test-primjera</span><br />
                                <textarea id="out_1" name="out_1"
class="outputs" rows="1" cols="40"><?php if(isset($_POST["out_1"])) echo
$_POST["out_1"];?></textarea><br />
                                <span>Ulaz 2. test-primjera</span><br />
                                <textarea id="in_2" name="in_2" class="inputs"
rows="1" cols="40"><?php if(isset($_POST["in_2"])) echo
$_POST["in_2"];?></textarea><br />
                                <span>Izlaz 2. test-primjera</span><br />
                                <textarea id="out_2" name="out_2"
class="outputs" rows="1" cols="40"><?php if(isset($_POST["out_2"])) echo
$_POST["out_2"];?></textarea><br />
                                <span>Ulaz 3. test-primjera</span><br />
                                <textarea id="in_3" name="in_3" class="inputs"
rows="1" cols="40"><?php if(isset($_POST["in_3"])) echo
$_POST["in_3"];?></textarea><br />
                                <span>Izlaz 3. test-primjera</span><br />
```

```
                                <textarea id="out_3" name="out_3"
class="outputs" rows="1" cols="40"><?php if(isset($_POST["out_3"])) echo
$_POST["out_3"];?></textarea><br />
                                <span>Ulaz 4. test-primjera</span><br />
                                <textarea id="in_4" name="in_4" class="inputs"
rows="1" cols="40"><?php if(isset($_POST["in_4"])) echo
$_POST["in_4"];?></textarea><br />
                                <span>Izlaz 4. test-primjera</span><br />
                                <textarea id="out_4" name="out_4"
class="outputs" rows="1" cols="40"><?php if(isset($_POST["out_4"])) echo
$_POST["out_4"];?></textarea><br />
                                <span>Ulaz 5. test-primjera</span><br />
                                <textarea id="in_5" name="in_5" class="inputs"
rows="1" cols="40"><?php if(isset($_POST["in_5"])) echo
$_POST["in_5"];?></textarea><br />
                                <span>Izlaz 5. test-primjera</span><br />
                                <textarea id="out_5" name="out_5"
class="outputs" rows="1" cols="40"><?php if(isset($_POST["out_5"])) echo
$_POST["out_5"];?></textarea><br />
                                <span>Ulaz 6. test-primjera</span><br />
                                <textarea id="in_6" name="in_6" class="inputs"
rows="1" cols="40"><?php if(isset($_POST["in_6"])) echo
$_POST["in_6"];?></textarea><br />
                                <span>Izlaz 6. test-primjera</span><br />
                                <textarea id="out_6" name="out_6"
class="outputs" rows="1" cols="40"><?php if(isset($_POST["out_6"])) echo
$_POST["out_6"];?></textarea><br />
                                <span>Ulaz 7. test-primjera</span><br />
                                <textarea id="in_7" name="in_7" class="inputs"
rows="1" cols="40"><?php if(isset($_POST["in_7"])) echo
$_POST["in_7"];?></textarea><br />
                                <span>Izlaz 7. test-primjera</span><br />
                                <textarea id="out_7" name="out_7"
class="outputs" rows="1" cols="40"><?php if(isset($_POST["out_7"])) echo
$_POST["out_7"];?></textarea><br />
                                <span>Ulaz 8. test-primjera</span><br />
                                <textarea id="in_8" name="in_8" class="inputs"
rows="1" cols="40"><?php if(isset($_POST["in_8"])) echo
$_POST["in_8"];?></textarea><br />
                                <span>Izlaz 8. test-primjera</span><br />
                                <textarea id="out_8" name="out_8"
class="outputs" rows="1" cols="40"><?php if(isset($_POST["out_8"])) echo
$_POST["out_8"];?></textarea><br />
                                <span>Ulaz 9. test-primjera</span><br />
                                <textarea id="in_9" name="in_9" class="inputs"
rows="1" cols="40"><?php if(isset($_POST["in_9"])) echo
$_POST["in_9"];?></textarea><br />
                                <span>Izlaz 9. test-primjera</span><br />
                                <textarea id="out_9" name="out_9"
class="outputs" rows="1" cols="40"><?php if(isset($_POST["out_9"])) echo
$_POST["out_9"];?></textarea><br />
                                <span>Ulaz 10. test-primjera</span><br />
                                <textarea id="in_10" name="in_10" class="inputs"
rows="1" cols="40"><?php if(isset($_POST["in_10"])) echo
$_POST["in_10"];?></textarea><br />
                                <span>Izlaz 10. test-primjera</span><br />
                                <textarea id="out_10" name="out_10"
class="outputs" rows="1" cols="40"><?php if(isset($_POST["out_10"])) echo
$_POST["out_10"];?></textarea><br />
                                <input type="submit" id="submit_task"
value="Spremi zadatak" />
```

```
                        </form>
                    </div>
                </div>
        </body>
</html>
<script>
        check_for_blank_field();
        document.getElementById("task_name").addEventListener("keyup",
check_for_blank_field);
        document.getElementById("task_text").addEventListener("keyup",
check_for_blank_field);
        document.getElementById("in_1").addEventListener("keyup",
check_for_blank_field);
        document.getElementById("out_1").addEventListener("keyup",
check_for_blank_field);
        document.getElementById("in_2").addEventListener("keyup",
check_for_blank_field);
        document.getElementById("out_2").addEventListener("keyup",
check_for_blank_field);
        document.getElementById("in_3").addEventListener("keyup",
check_for_blank_field);
        document.getElementById("out_3").addEventListener("keyup",
check_for_blank_field);
        document.getElementById("in_4").addEventListener("keyup",
check_for_blank_field);
        document.getElementById("out_4").addEventListener("keyup",
check_for_blank_field);
        document.getElementById("in_5").addEventListener("keyup",
check_for_blank_field);
        document.getElementById("out_5").addEventListener("keyup",
check_for_blank_field);
        document.getElementById("in_6").addEventListener("keyup",
check_for_blank_field);
        document.getElementById("out_6").addEventListener("keyup",
check_for_blank_field);
        document.getElementById("in_7").addEventListener("keyup",
check_for_blank_field);
        document.getElementById("out_7").addEventListener("keyup",
check_for_blank_field);
        document.getElementById("in_8").addEventListener("keyup",
check_for_blank_field);
        document.getElementById("out_8").addEventListener("keyup",
check_for_blank_field);
        document.getElementById("in_9").addEventListener("keyup",
check_for_blank_field);
        document.getElementById("out_9").addEventListener("keyup",
check_for_blank_field);
        document.getElementById("in_10").addEventListener("keyup",
check_for_blank_field);
        document.getElementById("out_10").addEventListener("keyup",
check_for_blank_field);
</script>
```

5.6. PRILOG design_own_task.css

```
body{
	background-color: black;
	background-image:
url("http://www.publicdomainpictures.net/pictures/130000/velka/green-smooth-
seamless-background.jpg");
}
#main_container{
	width: 60%;
	margin: 10% auto;
	background-color: rgba(255, 255, 255, 0.6);
	background-blend-mode: multiply;
}
#form_container{
	text-align: center;
	margin: 5%;
}
#main_container > p{
	text-align: center;
	color: red;
	font-size: 1.1em;
	margin: 0 auto;
}
#saving_outcome{
	color: red;
	font-size: 1.4em;
	font-weight: bold
}
a:link{
	color: #003300;
}
a:hover{
	color: #00BB00;
}
a:visited{
	color: #007700;
}
a:active{
	color: #00DD00;
}
#submit_task{
	width: 20%;
}
```

6. LITERATURA

[1] „Neki od učestalih programskih stogova“ https://en.wikipedia.org/wiki/Solution_stack#Some_common_named_stacks lipanj 2016.

[2] Goran Martinović, Radni materijali kolegija Operacijski Sustavi, ETFOS, OS-P-001.pdf, str 4,

[3] „Vrste operativnih sustava“ https://en.wikipedia.org/wiki/Operating_system#Types_of_operating_systems lipanj 2016.

[4] „Microsoft Windows“ https://en.wikipedia.org/wiki/Microsoft_Windows lipanj 2016.

[5] „Rane verzije Microsoft Windows-a“ https://en.wikipedia.org/wiki/Microsoft_Windows#Early_versions lipanj 2016.

[6] „Povijest Microsoft Windows operativnih sustava“ https://upload.wikimedia.org/wikipedia/commons/thumb/6/6d/Windows_Updated_Family_Tree.png/700px-Windows_Updated_Family_Tree.png lipanj 2016.

[7] „Raspodjela korisnika po različitim operativnim sustavima na tržištu“ https://en.wikipedia.org/wiki/Windows_10#Market_share_and_sales lipanj 2016.

[8] „Tekstualno sučelje MS-DOS operativnog sustava“ http://winsource.com/wp-content/uploads/2012/10/Win8DOS.png lipanj 2016.

[9] „Grafičko sučelje Windows 3.0 operativnog sustava“ https://upload.wikimedia.org/wikipedia/en/1/15/Windows_3.0_workspace.png lipanj 2016.

[10] „Grafičko sučelje Windows 10 operativnog sustava“ https://cdn0.vox-cdn.com/thumbor/xw68yMxmzret66x1Eb2h4YpnvRw=/cdn0.vox-cdn.com/uploads/chorus_asset/file/3913062/Screenshot__40_.0.png lipanj 2016.

[11] „Web poslužitelj“ https://en.wikipedia.org/wiki/Web_server lipanj 2016.

[12] „Raspodjela korisnika po različitim web poslužiteljima na tržištu“ http://news.netcraft.com/archives/2016/02/22/february-2016-web-server-survey.html lipanj 2016.

[13] „Prevođenje URL reference u putanju datoteka razumljivu operativnom sustavu“ https://en.wikipedia.org/wiki/Web_server#Path_translation lipanj 2016.

[14] „XAMPP homepage“ https://www.apachefriends.org/index.html lipanj 2016.

[15] „XAMPP download“ https://www.apachefriends.org/index.html lipanj 2016.

[16] „Uvod u HTML“ http://www.w3schools.com/html/html_intro.asp lipanj 2016.

[17] „Wikipedija članak o jeziku HTML“ https://en.wikipedia.org/wiki/HTML#Development lipanj 2016.

[18] „Primjer jednostavne HTML stranice“ http://help.websiteos.com/websiteos/example_of_a_simple_html_page.htm lipanj 2016.

[19] „GCC homepage“ https://gcc.gnu.org/ lipanj 2016.

[20] „MinGW“ http://www.mingw.org/ lipanj 2016.

[21] „Preuzimanje MinGW-a“ https://sourceforge.net/downloads/mingw lipanj 2016.

[22] „Dynu DDNS usluga“ https://www.dynu.com/ lipanj 2016.

[23] „Depicus online magic packet sender“ https://www.depicus.com/wake-on-lan/woli lipanj 2016.

[24] „Web aplikacija za natjecanje u programiranju“ http://tgudelj.dynu.net/ lipanj 2016.

7. SAŽETAK

Naslov: Web aplikacija za natjecanje u programiranju

Web aplikacija za natjecanje u programiranju je web stranica koja korisniku omogućava izbor zadatka čije je rješenje C++ programski kod posebno napisan da izračunima rješava problem zadan u zadatku. Korisniku se omogućava unos vlastitog programskog koda kao rješenja zadatka, a rezultati generirani od strane tog programskog koda se tada automatski uspoređuju s rezultatima očekivanima od strane same web aplikacije. Na temelju valjanosti rezultata korisniku se dodjeljuje broj bodova razmjeran točnosti koda, a korisnik može izabrati želi li da se njegov rezultat upiše na javnu rang listu.

Ključne riječi: web aplikacija, programiranje, natjecanje, online prevoditelj, C++ jezik

8. ABSTRACT

Title: Programming competition web application

The Programming competition web application is a web page allowing its user to attempt to solve a particular programming problem by designing a C++ program code such that, utilizing well ruminated computations and calculations, produces output results which are then compared to expected results discovered *a priori* by the web application in question. Based on validity of results produced by the user's program code, the user is awarded a proportionate number of score points, while allowing the user the option to have their name and score listed in a public scoreboard.

Keywords: web application, programming, competition, online compiler, C++ language

9. ŽIVOTOPIS

Tomislav Gudelj rođen je 18. kolovoza 1994. godine u Osijeku. Po završetku osnovne škole Bilje u Bilju, upisuje Treću gimnaziju u Osijeku i maturira 2013. godine, a zatim upisuje preddiplomski studij računarstva na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija. Uz položene ispite treće i nižih godina preddiplomskog studija krajem godine upisat će diplomski studij. 2015. godine sudjeluje na europskom natjecanju u programiranju CERC 2015, predvođenom od strane udruge ACM ICPC.