

Erasmus web aplikacija

Janeček, Ivor

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Polytechnic of Međimurje in Čakovec / Međimursko veleučilište u Čakovcu**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:110:773774>

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

Download date / Datum preuzimanja: **2024-08-27**



Repository / Repozitorij:

[Polytechnic of Međimurje in Čakovec Repository -
Polytechnic of Međimurje Undergraduate and
Graduate Theses Repository](#)





MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU
STRUČNI PRIJEDIPLOMSKI STUDIJ RAČUNARSTVO

IVOR JANEČEK, 0313019147

WEB APLIKACIJA ERASMUS

Završni rad

Čakovec, siječanj 2024.



MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU
STRUČNI PRIJEDIPLOMSKI STUDIJ RAČUNARSTVO

Ivor Janeček, 0313019147

Web aplikacija Erasmus
Erasmus web application

Završni rad

Mentor:

dr. sc. Bruno Trstenjak, prof. struč. stud.

Čakovec, siječanj 2024.

MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU
ODBOR ZA ZAVRŠNI RAD

Čakovec, 10. rujna 2021.

država: **Republika Hrvatska**
Predmet: **Programiranje - izvanredni**

ZAVRŠNI ZADATAK br. 2020-RAČ-I-162

Pristupnik: **Ivor Janeček (0313019147)**
Studij: **Računarstvo**
Smjer: **Programsko inženjerstvo**

Zadatak: **Web aplikacija Erasmus**

Opis zadatka:

Izrada web jednostranične aplikacije kojom će se omogućiti prikaz podataka vezanih za Erasmus razmjenu studenata na Veleučilištu. Aplikacija će se izvršavati u oblak okruženju. Aplikacija će omogućavati studentima prijavu u program mobilnosti, predaju potrebnih dokumenata te uvid u dinamiku rješavanja njihovog zahtjeva. U aplikaciji će biti implementiran mehanizam generiranja potrebnih dokumenata te izvješća o obavljenoj mobilnosti od strane studenata. Aplikacija će biti izrađena u Spring Boot tehnologiji bazirano na Java objektnom programskom jeziku. Za ostvarivanje dinamičnosti web aplikacije koristi će se AngularJS biblioteka. U aplikaciji će biti implementirani administrativni dio koji će omogućiti uređivanje svih podataka koji se zapisuju u MySQL bazi podataka. Sadržaj podataka u bazi i razrada tablica bit će dogovoreno naknadno s mentorom.

Zadatak uručen pristupniku: 10. rujna 2021.
Rok za predaju rada: 20. rujna 2021.

Mentor:



dr. sc. Bruno Trstenjak, v. pred.

Predsjednik povjerenstva za
završni ispit:

MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU

Bana Josipa Jelačića 22/a, Čakovec

IZJAVA O AUTORSTVU

Završni/diplomski rad isključivo je autorsko djelo studenta te student odgovara za istinitost, izvornost i ispravnost teksta rada. U radu se ne smiju koristiti dijelovi tuđih radova (knjiga, članaka, doktorskih disertacija, magistarskih radova, internetskih i drugih izvora) bez pravilnog citiranja. Dijelovi tuđih radova koji nisu pravilno citirani, smatraju se plagijatom i nezakonitim prisvajanjem tuđeg znanstvenog ili stručnoga rada. Sukladno navedenom studenti su dužni potpisati izjavu o autorstvu rada.

Ja, Ivor Janeček (ime i

prezime studenta) pod punom moralnom, materijalnom i kaznenom odgovornošću,

izjavljujem da sam isključivi autor/ica završnog/diplomskog rada pod naslovom

Web aplikacija Erasmus

te da u navedenom radu nisu na nedozvoljeni način (bez pravilnog citiranja) korišteni dijelovi tuđih radova.

Student/ica:

Ivor Janeček
(vlastoručni potpis)

ZAHVALA

Želim se zahvaliti dragom mentoru na usmjeravanju i vođenju prilikom pisanja, čime je omogućio da temu završnog rada kvalitetno razložim i sročim u strukturiranu cjelinu.

Ivor Janeček

Sažetak

Erasmus+ je program EU koji promiče *mobilnost*, suradnju i cjeloživotno učenje. Olakšava razmjenu učenika, nastavnika i osoblja, potičući kulturno razumijevanje i stjecanje vještina. Projekt podržava suradnju u visokom obrazovanju, zajedničke diplome i istraživačke projekte. Erasmus+ unapređuje strukovno obrazovanje i osposobljavanje kroz praktično iskustvo i suradnju u industriji.

Kako bi se pojednostavio proces prijave za *Erasmus+* projekt, razvijena je namjenska web aplikacija. S obzirom na široku dostupnost Interneta, ovu web aplikaciju sada može koristiti gotovo svatko. Služi kao osnovni alat za *Erasmus+*, olakšava administrativne zadatke, poboljšava komunikaciju i optimizira zadovoljstvo korisnika.

Svi studenti i profesori povezani s veleučilištem imaju priliku prijaviti se za *Erasmus+* razmjenu. Web aplikacija *Erasmus* nudi razne ključne značajke, uključujući korisničko sučelje za registraciju i podnošenje prijava, što studentima olakšava prijavu za programe studiranja ili prakse.

Dodatno, aplikacija omogućuje pristup svim natjecajima veleučilišta s prethodno sklopljenim ugovorima. Zainteresirani studenti i profesori mogu se prijaviti na ove natječaje, ali je važno da su priloženi svi potrebni dokumenti i podaci kako bi prijava bila prihvaćena. Aplikacija osim studenata i profesora ima još jedan tip korisnika a to je administrator.

Administrator ima značajnu ulogu u aplikaciji jer raspisuje natječaje. Štoviše, mogu upravljati korisnicima i odrediti koje će prijave biti prihvaćene ili odbijene.

Tehnologije korištene u razvoju aplikacije uključuju *Spring Boot*, *MySQL*, HTML i CSS. Ove tehnologije su široko poznate i većina programera je upoznata s njima. Kako bismo poboljšali jednostavnost korištenja ovih tehnologija, koristimo alate kao što su IntelliJ i Visual Studio Code. Ovi nam alati pomažu u pisanju i analizi koda kako bismo smanjili pogreške.

Za sigurnost korišten je *Spring Security* koji pomoću svog mehanizma pretvaranja lozinke pri autentifikaciji i autorizaciji osigurava poboljšanje sigurnosti aplikacije.

Heroku je korišten za implementaciju aplikacije u oblak okruženje. Odabran je iz razloga što je platforma koja omogućuje jednostavno upravljanje aplikacijama. Heroku je platforma u oblaku koja tvrtkama omogućuje izradu, isporuku, nadzor i skaliranje aplikacija. Povezivanjem s GitHub projektom, sva ažuriranja na GitHubu automatski će se odraziti na novu verziju aplikacije.

Ključne riječi: *Erasmus, mobilnost, oblak aplikacija, Spring Boot, MySQL.*

Sadržaj

1. Uvod	3
2. Cilj rada	4
3. Korištene tehnologije	6
3.1. Opisni jezik HyperText Markup Language (HTML)	6
3.2. Stilski jezik Cascading Style Sheets (CSS)	7
3.3. Spring Boot	8
3.4. MySQL	9
4. Alati korišteni za izradu aplikacije	11
4.1. IntelliJ	11
4.2. Maven	12
4.3. Heroku	13
4.4. Visual Studio Code	14
4.5. GitHub	15
4.6. WampServer	15
5. Struktura aplikacije	17
6. Baza podataka	25
7. Testiranje rada aplikacije	27
8. Sigurnost aplikacije	36
9. Zaključak	40
10. Literatura	41
11. Popis slika	42
12. Popis koda	44

1. Uvod

ERASMUS+ je vodeći program mobilnosti Europske unije u obrazovanju i osposobljavanju i jedna je od najpoznatijih akcija na razini EU-a. Osnovan je 1987. i od tada je omogućio više od 2 milijuna studenata iz cijele Europe da steknu obogaćujuća iskustva učenja u drugim zemljama.

Iako su ERASMUS razmjene sada popularan i uspostavljen dio visokog obrazovanja diljem Europe, put do uspostave programa nije uvijek bio gladak.

Stari logo Europska komisija podržavala je pilot razmjenu studenata šest godina prije nego što je ERASMUS nastao. Predložio je izvorni program ERASMUS početkom 1986., ali su reakcije zemalja članica EU-a bile različite: one koje su imale značajne vlastite programe razmjene bile su uglavnom neprijateljski nastrojene, dok su preostale zemlje općenito bile za [1].

Razvijanje digitalnih vještina i kompetencija u sektorima koji gledaju u budućnost kao što su borba protiv klimatskih promjena, čista energija, umjetna inteligencija, robotika i analiza velikih podataka ključno je za održivi gospodarski rast i budućnost kohezije u Europi. Program može dati značajan doprinos poticanjem inovacija i pružanjem znanja, vještina i kompetencija potrebnih u Europi.

Cilj završnog rada je izraditi web aplikaciju za potrebe vođenja poslova mobilnosti u sklopu programa Erasmus+. U aplikaciji su implementirane razne funkcionalnosti koje doprinose digitalizaciji cijelog procesa mobilnosti. Pomoću aplikacije studentima i profesorima se daje mogućnost jednostavne prijave na natječaje mobilnosti, predaja raznih dokumenata te transparentno praćenje cijelog procesa mobilnosti.

2. Cilj rada

Erasmus+ je najveći program Europske unije za obrazovanje, osposobljavanje, mlade i sport i pokriva razdoblje od 2014. do 2020. godine. Obuhvaća sve europske i međunarodne programe i inicijative Europske unije u područjima obrazovanja (opće obrazovanje, visoko obrazovanje, obrazovanje odraslih), osposobljavanja (strukovno obrazovanje), mladih i sporta. Erasmus+ ima za cilj jačanje znanja, vještina i zapošljivosti europskih građana te poboljšanje obrazovanja, osposobljavanja i rada u područjima mladih i sporta. Posebice ima za cilj umrežavanje sektora obrazovanja, osposobljavanja i mladih s gospodarstvom te je otvoren za njihove zajedničke projekte [2].



Slika 1. ERASMUS+

(izvor: <https://www.schengenvisainfo.com/news/10-things-you-probably-didnt-know-about-erasmus-according-to-eu-commission/>)

Jedna od visokoškolskih ustanova koja sudjeluje u ovom programu je Međimursko veleučilište u Čakovcu (MEV) [3]. Cilj ovog diplomskog rada je izraditi online aplikaciju koja studentima i profesorima omogućuje laku prijavu za Erasmus+ program koji provodi ovo veleučilište.

Za razvoj korištene su sljedeće tehnologije: Spring Boot tehnologije za pozadinski dio (eng. *Backend*) te Angular programski okvir za vidljivi dio (eng. *Frontend*). Na taj način nastoji se omogućiti korisnicima da se prijave na natječaje raspisane od strane veleučilišta vezane za Erasmus+ program.

Aplikacija je jednostavna za korištenje, namjena studentima i nastavnicima. Studenti i nastavnici imaju mogućnost prijaviti se na natječaje za mobilnosti te uređivati svoje prijave.

Osim navedenih korisnika aplikacije, pristup aplikaciji ima i administrator aplikacije. Uloga administratora je raspisivanje natječaja ,odobrenje ili odbijanje prijava, upravljanje podacima korisnika te dokumentima pojedinačnih prijava.

3. Korištene tehnologije

3.1. Opisni jezik HyperText Markup Language (HTML)

HyperText Markup Language ili HTML standardni je označni jezik za izradu web stranica. 1993. Tim Berners-Lee napisao je prvu verziju HTML-a. Bio je fizičar i informatičar koji je implementirao prvu uspješnu komunikaciju između HTTP klijenta. On je zapravo izumitelj interneta kakvog danas poznajemo i volimo. HTML je imao mnogo verzija tijekom sljedećih nekoliko godina. HTML v2 objavljen je 1995., nakon čega su uslijedili v3 i v4 1997. Na kraju, to nas dovodi do HTML5, koji je objavljen 2011. [4].

HTML se ne smatra programskim jezikom, već skriptnim jezikom. To znači da se koristi za opisivanje izgleda web-mjesta, web-aplikacija ili mobilnih aplikacija



Slika 2. HTML

(Izvor: <https://www.digitalocean.com/community/tutorials/how-to-create-tables-in-html/>)

Kada web-preglednik komunicira s web-stranicom, koristi HTML za primanje blok rasporeda elemenata web-stranice. Iako je moguće opisati estetski izgled stranice pomoću HTML-a, to se više ne preporučuje. CSS se sada koristi za ovu svrhu, kao što je objašnjeno u sljedećem poglavlju. U primjeru koda 1. prikazan je jednostavni primjer HTML-a. Ovaj kod predstavlja blok raspored HTML dokumenata. Osnovni elementi uključuju zaglavlje, naziv dokumenta, tijelo dokumenta, podnaslove (na engleskom "h1") i odlomke (na engleskom "p"). Početak elementa označava se uglatim zagradama unutar kojih se nalazi oznaka elementa. Kraj elementa označen je istom oznakom, kojoj prethodi

kosa crta. Sadržaj elementa nalazi se između početne i završne oznake, a unutar jednog elementa mogu biti i drugi elementi.

```
<div class="col-md-3"></div>

<div class="col-md-9">

  <h1 class="display-4"><span>Dobrodošli na</span> ERASMUS</h1>

  <p class="lead">Početna stranica</p>

  <div [ngSwitch]="account !== null">

    <div class="alert alert-success" *ngSwitchCase="true">

      <span id="home-logged-message" *ngIf="account">Ulogiran si kao
"{{ account.login }}"</span>

    </div>

    <div class="alert alert-warning" *ngSwitchCase="false">

      <span>Ulogiraj se ovdje</span>&nbsp;

      <a class="alert-link" routerLink="/login">Logiranje</a>

    </div>

    <div class="alert alert-warning" *ngSwitchCase="false">

      <span>Nemaš račun , registriraj se</span>&nbsp;

      <a class="alert-link"
routerLink="account/register">Registracija</a>

    </div>

  </div>

</div>
```

Kod 1. HTML kod za prikaz početne stranice

(izvor: Autor)

3.2. Stilski jezik Cascading Style Sheets (CSS)

CSS je kratica za Cascading Style Sheets jezik i koristi se za stiliziranje elemenata napisanih u označnom jeziku kao što je HTML. Odvaja sadržaj od vizualnog prikaza stranice. Odnos između HTML-a i CSS-a snažno je povezan jer je HTML sam temelj web-mjesta, a CSS je estetika cijele web-lokacije [5].

```
.error {
```

```
    color: white;

    background-color: red;
}

.form-control {

    background-color: #fff;
}

.readonly {

    background-color: #eee;

    opacity: 1;
}
```

Kod 2. Primjer korištenja CSS-a za forme, error-e i readonly
(izvor: Autor)

Iz primjera koda 2. može se vidjeti uporaba CSS-a u web aplikaciji na način da se promjeni boja pozadine u formi, za errore te za read only. Njegova uporaba je jednostavna i nudi brojna stilska rješenja. Pomoću CSS-a možemo promijeniti svojstva standardnih elemenata, promijeniti njihove dimenzije, pa čak i prilagoditi njihove obrube. Osim toga, omogućuje izmjene fontova teksta, boja pozadine i boja teksta. Štoviše, koristi se za dizajniranje popisa, gumba, tablica obrazaca i slika. Nadalje, omogućuje dodavanje sjena, kao i 2D i 3D transformacije. Može se utvrditi da nam CSS omogućuje da sve oblikujemo po želji.

3.3. Spring Boot



Slika 3. Spring boot logo
(izvor: <https://medium.com/swlh/customizing-the-spring-boot-banner-52f955a85e86>)

Spring Boot je okvir otvorenog koda koji olakšava stvaranje i implementaciju web aplikacija. Izgrađen je na temelju popularnog Java okvira, Spring, i pruža razvojnim programerima snažan skup alata za brzo stvaranje i implementaciju web aplikacija.

Uključuje snažan generator koda koji može brzo generirati kod za web aplikacije i niz razvojnih alata koji pomažu programerima da brzo kreiraju i implementiraju svoje aplikacije. Dodatno, uključuje značajke kao što su lokalni razvojni poslužitelj, generator koda i niz alata naredbenog retka koji programerima olakšavaju brzu izgradnju, testiranje i implementaciju njihovih aplikacija.

Jedna od glavnih prednosti Spring Boot-a je njegova sposobnost da podržava više web tehnologija. Prikladan je za rad s popularnim web tehnologijama kao što su Java, JavaScript, HTML i CSS, a također uključuje generatora koda i razne alate naredbenog retka koji čine programerima da lako, brzo izgrade, testiraju i implementiraju svoje aplikacije.

3.4. MySQL



Slika 4. MySQL logo

(izvor: <https://aety.io/mysql-logo-2/?lang=en>)

MySQL je sustav za upravljanje relacijskim bazama podataka (RDBMS) koji je stvorila Oracle Corporation. To je jedna od najpopularnijih i najkorištenijih baza podataka otvorenog koda u svijetu [6].

MySQL je relacijska baza podataka, što znači da pohranjuje podatke u tablice i koristi Structured Query Language (SQL) za pristup i rukovanje tim podacima. MySQL je brz, pouzdan i jednostavan za korištenje. Podržava širok izbor značajki, uključujući okidače, pohranjene procedure i prikaze. MySQL je važan dio LAMP skupa, koji označava Linux, Apache, MySQL i PHP. Ovaj skup se koristi za izradu dinamičkih web stranica i popularan je izbor za web razvoj. MySQL je izvrstan izbor za mnoge aplikacije jer je

pouzdan i lak za naučiti. Sa svojom prirodom otvorenog koda, MySQL se može lako prilagoditi i proširiti. Iz tih razloga, MySQL je izvrstan izbor za bilo koju aplikaciju koja se temelji na podacima. Iz primjera koda 3. može se vidjeti kako se pomoću MySQL koda mogu filtrirati natječaji tako da se prikažu samo oni koji imaju status otvoren.

```
SELECT * FROM natjecaj WHERE status = 'OTVOREN';
```

Kod 3. MySQL kod koji dohvaća sve natječaje koji imaju status otvoren

(Izvor: Autor)

4. Alati korišteni za izradu aplikacije

4.1. IntelliJ

IntelliJ IDEA, integrirano razvojno okruženje (IDE) za JVM jezike, ima za cilj povećati produktivnost programera. JVM jezici su: Java, Kotlin, Scala, Groovy [7].

Ono nudi razne mogućnosti koje mogu ubrzati rad ili ga čak i popuniti. Ima u sebi ugrađen i git pa je lako implementirati svoj kod na Github repozitoriju. Git je distribuirani sustav kontrole verzija koji se koristi u razvoju softvera za praćenje promjena u kodu, omogućavajući suradnju, praćenje povijesti i upravljanje kodom. Programerima omogućuje simultani rad na projektima, spajanje promjena i održavanje pouzdane povijesti revizija, poboljšavajući kvalitetu koda i upravljanje projektima.

IntelliJ IDEA služio je kao glavni uređivač koda za backend tijekom izrade web aplikacije. Ovaj sofisticirani softver nudi širok raspon značajki i alata posebno dizajniranih za poboljšanje Java razvoja.

Sa svojom naprednom analizom koda, besprijekornom navigacijom koda i učinkovitim mogućnostima refaktoriranja, programeri su ovlaštteni proizvesti elegantan Java kod koji se može održavati. Štoviše, ovaj izvanredni softver može se pohvaliti opsežnim ekosustavom dodataka i proširenja koje su razvili JetBrains i zajednica. Ova proširenja omogućuju korisnicima da prilagode funkcionalnost IDE-a za podršku različitim jezicima, okvirima i alatima, nudeći neusporedive mogućnosti prilagodbe. Sjaj IntelliJ IDEA također leži u njegovom inteligentnom dovršavanju koda, sveobuhvatnoj inspekciji koda i funkcijama označavanja grešaka.

Razumijevajući strukturu vašeg koda, ovaj izvanredni alat ne samo da daje vrijedne prijedloge, već također automatski ispravlja brojne uobičajene probleme kodiranja, čime značajno povećava produktivnost.

Besprijekorna integracija IntelliJ IDEA s popularnim alatima za izgradnju kao što su Maven i Gradle pojednostavljuje upravljanje ovisnostima, izgradnju projekta i izvođenje testova, osiguravajući besprijekorno razvojno iskustvo.

JetBrains, cijenjena tvrtka koja stoji iza IntelliJ IDEA, poznata je po svom aktivnom održavanju i redovitim ažuriranjima svojih proizvoda. Ova predanost osigurava da se

korisnici mogu osloniti na dosljedne ispravke grešaka, ažuriranja i poboljšanja, jamčeći iznimno korisničko iskustvo. Nadalje, ovaj softver obuhvaća ugrađene provjere koda i alate za analizu kvalitete koji mogu identificirati potencijalne nedostatke i predložiti poboljšanja.



Slika 5. IntelliJ IDEA logo

(izvor: https://en.m.wikipedia.org/wiki/File:IntelliJ_IDEA_Icon.svg)

4.2. Maven

Maven je alat koji se sada može koristiti za izgradnju i upravljanjem bilo kojim projektom temeljenim na Javi [8]. Jedna od njegovih ključnih prednosti leži u njegovoj podršci za više web tehnologija. Pogodan je za rad s popularnim web tehnologijama kao što su Java, JavaScript, HTML i CSS. Štoviše, uključuje snažan generator koda posebno dizajniran za web aplikacije. Sve u svemu, Maven se pokazao neprocjenjivim alatom za programere koji zahtijevaju brzu izradu i implementaciju projekata temeljenih na Javi. Jednostavan je za korištenje i nudi opsežnu biblioteku alata koji olakšavaju brzi razvoj i implementaciju aplikacija.

Osim toga, neprimjetno se integrira s raznim web tehnologijama i pruža programerima generator koda za web aplikacije. Maven, koji je razvijen od strane Apache je alat za automatizaciju izgradnje koji za programere ima za cilj pojednostaviti stvaranje, izgradnju i upravljanje projektima temeljenim na Javi. U web aplikaciji Erasmus, Maven je korišten za automatizaciju izrade i kao alat za upravljanjem projektom. Maven je pomogao učinkovito upravljati ovisnostima projekta definirajući ih u POM datoteci projekta, slika 6. Također je automatizirao proces izrade, osiguravajući da je projekt sastavljen, testiran i pakiran u JAR datoteku što olakšava distribuciju i implementaciju.

Osim toga Maven je upotrijebljen za konfiguriranje raznih dodataka i određivanje verzija za različite biblioteke, osiguravajući dosljedne i predvidljive rezultate.

```
<properties>
  <!-- Build properties -->
  <maven.version>3.2.5</maven.version>
  <java.version>11</java.version>
  <node.version>v16.17.0</node.version>
  <npm.version>8.19.1</npm.version>

  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
  <maven.build.timestamp.format>yyyyMMddHHmmss</maven.build.timestamp.format>
  <maven.compiler.source>${java.version}</maven.compiler.source>
  <maven.compiler.target>${java.version}</maven.compiler.target>
  <start-class>zavrsni.erasmus.ErasmusApp</start-class>
  <argLine>-Djava.security.egd=file:/dev/./urandom -Xmx16</argLine>
  <m2e.apt.activation>jdt_apt</m2e.apt.activation>
  <run.addResources>false</run.addResources>
</properties>
```

Slika 6. Konfiguracija verzija unutar POM datoteke

(izvor: Autor)

4.3. Heroku

Heroku je cloud platforma kao usluga koja se temelji na spremniku. Herokuom se u potpunosti upravlja, što programerima daje slobodu da se usredotoče na svoj temeljni proizvod bez ometanja održavanja poslužitelja, hardvera ili infrastrukture. Heroku iskustvo pruža usluge, alate, tijekove rada i poliglotsku podršku, sve osmišljeno za povećanje produktivnosti programera [9].

Heroku je platforma koja je jednostavna i nudi puno mogućnosti. Lako se postavi i koristi online bazu podataka. Platforma ima metriku koja se može koristiti za provjeru vremena odziva aplikacije. Uz to, sadrži i prozor pregled zapisnika koji zapisuje sve što aplikacija odrađuje i jako je bitan u slučaju greške u aplikaciji. Najistaknutija mogućnost koja nam nudi aplikacije je ta da se može povezati sa Github-om te sve promjene može vršiti automatski.



Slika 7. Heroku logo

(izvor: <https://logowik.com/heroku-vector-logo-5276.html>)

4.4. Visual Studio Code

Visual Studio Code, razvojno okruženje, može se pokrenuti na vašem stolnom računalu i kompatibilan je sa sustavima Windows, macOS i Linux. Uključuje ugrađenu podršku za JavaScript, TypeScript i Node.js te nudi širok raspon proširenja za različite jezike i okruženja, uključujući C++, C#, Java, Python, PHP, Go i .NET [10].

Dodatno, Visual Studio Code podržava sustave kontrole verzija, omogućujući razvojnim programerima da predaju svoj kod u udaljena spremišta. Visual Studio Code Marketplace služi kao online repozitorij s brojnim proširenjima i dodacima, omogućujući razvojnim programerima da dodatno prilagode uređivač.

Ovi dodaci služe raznim zadacima poput isticanja sintakse, oblikovanja koda i lintinga, kao i popularnih okvira i biblioteka kao što su React i Vue. Visual Studio Code vrlo je prilagodljiv uređivač koda koji programerima omogućuje da ga prilagode svojim specifičnim zahtjevima.

Stekao je popularnost među programerima zahvaljujući širokom rasponu značajki, fleksibilnosti i performansi. Uređivač također nudi mogućnosti otklanjanja pogrešaka, omogućujući programerima da koračaju kroz svoj kod, postavljaju prijelomne točke i pregledavaju varijable.

Zaključno, Visual Studio Code je uređivač koda otvorenog koda koji je postao preferirani izbor za mnoge programere. Njegovo intuitivno sučelje, zajedno s otklanjanjem pogrešaka, kontrolom verzija i podrškom za dodatke, čine ga visoko prilagodljivim individualnim preferencijama i potrebama. Štoviše, Visual Studio Code pruža integrirani terminal koji programerima omogućuje pristup naredbenom retku izravno iz uređivača.

Njegovo korisničko sučelje osigurava da programeri mogu brzo postati produktivni, potpomognuti značajkama kao što su navigacija kodom i ploče za pretraživanje za brzi pristup određenim redcima ili tekstualnim nizovima.



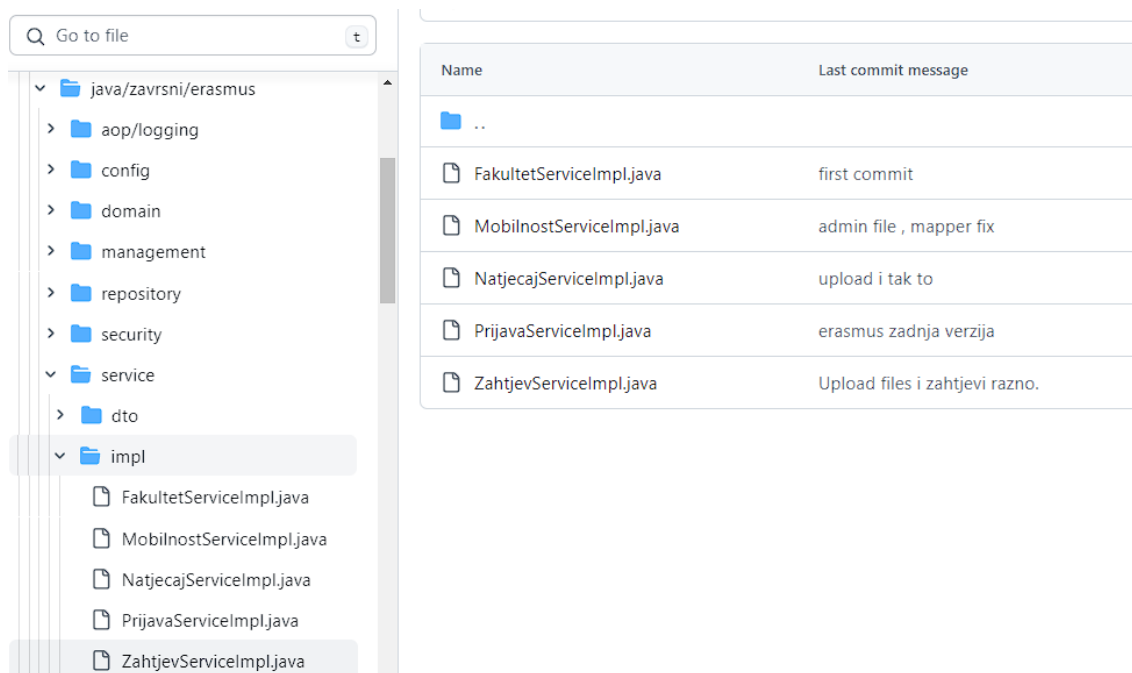
Slika 8. VSCode logo

(Izvor: https://en.m.wikipedia.org/wiki/File:Visual_Studio_Code_1.35_icon.svg)

4.5. GitHub

GitHub je web-bazirana platforma za kontrolu verzija i suradnju za programere softvera. Microsoft, najveći pojedinačni doprinos GitHubu, kupio je platformu za 7,5 milijardi dolara 2018. GitHub, koji se isporučuje kroz poslovni model softvera kao usluge (SaaS), pokrenut je 2008. Utemeljen je na Gitu, otvorenom kodu sustav upravljanja koji je stvorio Linus Torvalds za bržu izradu softvera [10].

Github nudi opciju skupnog uređivanja i spremanja koda, što ga čini idealnim za zajednički rad. Jednostavan je za korištenje i nudi brojne značajke. Na primjer, omogućuje korisnicima da ili sakriju kod od javnosti ili ga učine vidljivim svima. Iako svatko može pregledavati i preuzimati javne repozitorije, samo registrirani korisnici imaju mogućnost dodavanja sadržaja u repozitorije. Sa slike 9. može se vidjeti web aplikacija u javnom Github repozitoriju.



Slika 9. Implementacija servisa web aplikacije Erasmus u Github repozitoriju
(Izvor: Autor)

4.6. WampServer

WampServer je softverski paket otvorenog koda samo za Windows koji pruža potpuno rješenje jednostavno za korištenje za postavljanje i upravljanje lokalnim web poslužiteljem.

Dizajniran je za rad na Windows platformi, a uključuje Apache web poslužitelj, MySQL poslužitelj baze podataka i PHP skriptni jezik. Korisničko sučelje WampServera jednostavno je za korištenje i intuitivno, što korisnicima omogućuje brzo i jednostavno postavljanje, konfiguraciju i upravljanje njihovim lokalnim web poslužiteljem.

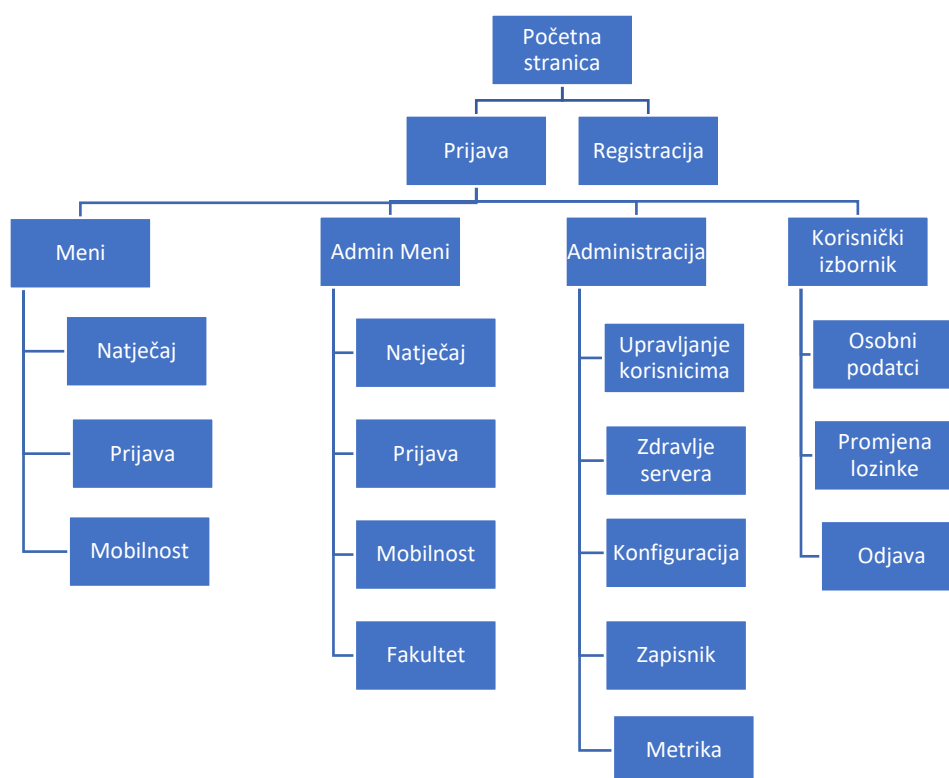
WampServer je odličan izbor za one koji žele brzo i jednostavno postaviti web poslužitelj na svoje računalo. Besplatan je za preuzimanje i korištenje, a pruža dobar skup značajki i alata koji olakšavaju instaliranje i upravljanje web aplikacijama.

Također, ima niz sigurnosnih značajki, kao što je mogućnost zaštite web aplikacija lozinkom i mogućnost pokretanja više web aplikacija istovremeno.

Wamp server je korišten u razvoju web aplikacije Erasmus radi brže i lakše interakcije s bazom podataka. Osim toga, korišten je i u svrhe testiranja interakcije aplikacije s bazom podataka prije nego što se aplikacija prebaci u oblak okruženje.

5. Struktura aplikacije

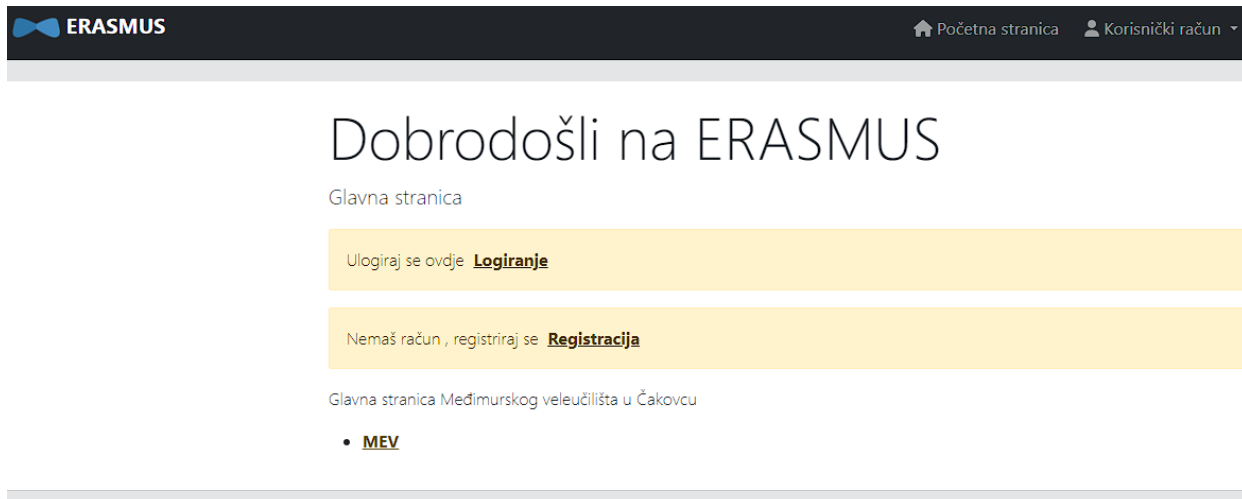
Aplikacija se sastoji od više modula koje imaju različite funkcionalnosti. U aplikaciji razlikujemo 3 vrste korisnika : administrator, profesor i student. Aplikacija započinje s početnom stranicom za prijavu i registraciju korisnika. Nakon prijave korisnici imaju uvid module za natječaj, prijave za natječaj i mobilnosti. Ti moduli sadrže sve potrebne informacije za mobilizaciju unutar Erasmus projekta. Svi korisnici mogu koristiti i korisnički izbornik gdje mogu promijeniti osobne podatke, lozinku te se odjaviti sa svog računa. Administrator ima svoj administratorski meni gdje može vidjeti i upravljati svim natječajima, prijavama, mobilnostima i fakultetima. Osim toga administrator ima i dodatni meni pod nazivom administracija. Tu su sve bitne informacije o korisnicima, serveru, konfiguraciji i zapisniku. Administrator može upravljati tim informacijama da bi postigao optimalni rad aplikacije. Slika 10. prikazuje hijerarhijsku strukturu web aplikacije Erasmus.



Slika 10. Hijerarhijska struktura aplikacije

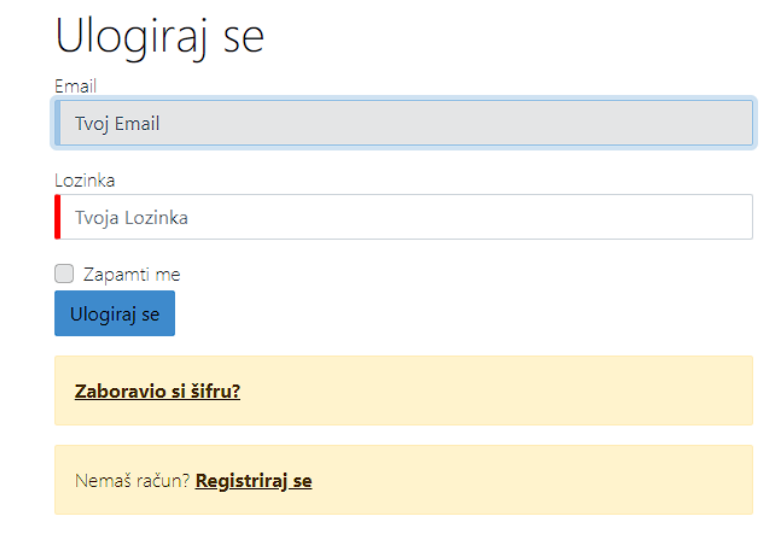
(izvor: Autor)

Početna stranica, Slika 11, koju mogu vidjeti svi korisnici. Za pristup aplikaciji potrebna je prijava korisnika. Ona se vrši putem prijavnice koja sadrži bitne informacije za prijavu, a to su email i lozinka.



Slika 11. Početna stranica web aplikacije Erasmus
(izvor: Autor)

Na stranici prijave, Slika 12, ponuđena je i opcija „zaboravio sam lozinku“ kojom se pristupa stvaranju nove lozinke u slučaju ako je stara izgubljena ili zaboravljena.



Slika 12. Stranica prijave
(izvor: Autor)

Za registraciju novog korisnika aplikacija nudi korisniku registracijsku formu kao što je vidljivo na slici 13. U toj formi obavezna polja su korisničko ime, email, lozinka i potvrda lozinke. Aplikacija provjerava email adrese u slučaju korištenja iste adrese ili nevažeće. Za lozinku postoji redak koji pokazuje koliko je lozinka snažna te što više polja svijetli lozinka je sigurnija.

Ulogiraj se'."/>

Registracija

Korisničko ime

Vaše Korisničko ime

Email

Vaš email

Nova Lozinka

Nova lozinka

Snaga lozinke:

— — — — —

Potvrda nove lozinke

Potvrdi novu lozinku

Register

Već imaš račun [Ulogiraj se](#)

Slika 13. Stranica registracije

(izvor: Autor)

Odabirom prve opcije menija po nazivom natječaj otvori se modul koji pokaže listu svih natječajaja ovisno o rolu prijavljenom korisniku. Administrator može uređivati podatke o natječajima, ostali korisnici mogu pogledati listu svih natječajaja ovisno o vrsti korisnika. Lista natječajaja, slika 14, raspoređena je po datumu raspisivanja natječajaja od najnovijeg do najstarijeg datuma. Administrator može dodavati zahtjeve koje korisnik, koji se prijavljuje, treba popuniti. U listi je moguće pogledati određene detalje natječajaja, te ako je natječaj još u tijeku korisnik je u mogućnosti predati zahtjev za prijavu za taj određeni natječaj. U aplikaciji natječajaji su organizirani u dvije kategorije. Natječajaji za studente

imaju kategorije: nastava i stručna praksa. Natječaji za profesore imaju kategorije: stručno predavanje i stručno usavršavanje.

Ime ↕	Opis ↕	Napravljen ↕	Datum od ↕	Datum do ↕	Status ↕
<u>NATJEČAJ ZA PROFESORE</u> <u>25.08.2023</u>	ERASMUS NATJEČAJ ZA PROFESORE	25.08.2023 09:42:00	23.08.2023	31.08.2023	OTVOREN
<u>NOVI NATJEČAJ ZA STUDENTE</u> <u>25.08.2023</u>	NOVI NATJEČAJ ZA STUDENTE MEĐIMURSKOG VELEUČILIŠTA.	25.08.2023 09:40:00	24.08.2023	31.08.2023	OTVOREN
<u>Drugi natječaj za studente za mjesec kolovoz</u>	Drugi natječaj za studente Natječaj raspisan za mjesec kolovoz Natječaj zahtjeva motivacijsko pismo i ocjene. Testiranje natječaja.	31.07.2023 20:07:00	01.08.2023	24.08.2023	OTVOREN
<u>Novi natjecaj za Studente</u>	Novi natjecaj za Studente natječaj za mjesec kolovoz natječaj raspisan samo za studente	31.07.2023 20:06:00	01.08.2023	31.08.2023	OTVOREN

Slika 14. Lista natječaja
(izvor: Autor)

Korisnik svoje prijave može vidjeti pod opcijom moje prijave iz menija kojeg korisnici mogu vidjeti dok administrator ima uvid u sve prijave. Uređivanje prijava je moguće ali sa ograničenjem te je to dostupno samo korisniku koji je napravio prijavu. Isto tako je i sa brisanjem prijava. Detalji prijave, slika 15, sastoje se od informacija prijave, zahtjeva i dokumenata. U prijavu je potrebno staviti dokumenata ovisno o broju zahtjeva. Korisnik dokumente može unijeti, urediti pogledati i obrisati. Administrator je u mogućnosti pogledati dokument unesen od strane korisnika. Prijava može imati jedno od 3 statusa a to su nova prijava, prihvaćen, odbijen. Administrator može potvrditi ili odbiti prijavu ali samo ako je ta prijava ispravna i popunjena do kraja.

Prijava		Zahtjevi	Dokumenti
Ime prijave	Prijava za NATJEČAJ ZA PROFESORE 25.08.2023	ZAHTJEV 1	Otvori Upute za prijavitelje (1).docx
Napravljen	25.08.2023 09:43:00		
Status	PRIHVACEN		
Trajanje od	19.10.2023		
Trajanje do	16.11.2023		
Kategorija	STRUCNO_USAVRSAVANJE		
Korisnik	bruno		
Fakultet	Fakultet A		
Natječaj	<u>NATJEČAJ ZA PROFESORE 25.08.2023</u>		

[← Nazad](#)

Slika 15. Detalji prijave

(izvor: Autor)

Nakon što administrator potvrdi prijavu aplikacija automatski popunjava podatke te kreira novu mobilnost te prosljeđuje administratora na listu mobilnosti kao što prikazuje slika 16. Kao i sa prijavama studenti i profesori mogu vidjeti samo svoje mobilnosti dok administrator ima uvid u sve mobilnosti.

Kada se klikne na pogledaj pokraj određene mobilnosti gumb nas proslijedi na detalje te određene mobilnosti. U tim detaljima su općenite stvari kao od kad i do kad je trajala mobilnost itd. Pokraj toga se nalaze dvije tablice dokumenata. Jedna tablica je za učesnike mobilizacije, a druga za administratora. U te tablice korisnici i administrator unose bitne dokumente koje su potrebne da bi se mobilnost završila.

Administrator ima i dodatne opcije kao što su zatvori mobilnost i neispravna mobilnost. Ako je mobilnost neispravna, to znači da korisnik nije unesao dobre dokumente ili da ih je manjak. Da bi administrator mogao kliknuti zatvori mobilnost, dokumenti moraju biti ispravni te datum završetka mobilnosti ne smije biti manje od trenutnog datuma.

Mobilnosti

Ime ↕	Status Mobilnosti ↕	Natječaj ↕	Prijava ↕
Mobilnost za Novi natjecaj za Studente	ZATVORENA	Novi natjecaj za Studente	Prijava za Novi natjecaj za Studente
Mobilnost za NATJEČAJ ZA PROFESORE 25.08.2023	OTVORENA	NATJEČAJ ZA PROFESORE 25.08.2023	Prijava za NATJEČAJ ZA PROFESORE 25.08.2023

Slika 16. Lista mobilnosti

(izvor: Autor)

Profil korisnika je stranica koja sadrži javno dostupne podatke nekog korisnika i vidljiva je svim prijavljenim korisnicima. Sadrži osobne podatke poput imena i prezimena. Korisnik može uređivati vlastite podatke na profilu korisnika.

Administrator ima posebni meni pod imenom admin meni. Taj meni sadrži sve isto kao i korisnički ali ima i dodatnu opciju pod nazivom fakultet. Kada se klikne na taj odabir dobijemo u uvid popis fakulteta kao što je prikazano na slici 17. Administrator je jedini koji ga može vidjeti, a također i jedini koji može dodavati nove fakultete, uređivati ih ili brisati.

Ime ↕	Zemlja ↕	Grad ↕	Adresa ↕
Fakultet A	Hrvatska	Čakovec	Park
Fakultet B	Slovenija	Maribor	Koroška cesta 46

Slika 17. Lista fakulteta

(izvor: Autor)

Administrator je jedini u mogućnosti u odabiru menija Administracija na kojem može obavljati jedno od važnijih opcija administracije aplikacije. Prvi izbor u meniju su svi računi korisnika, a administrator je jedini u mogućnosti aktivirati ih ili obrisati. Slika 18 prikazuje popis korisnika web aplikacije.

ID ^	Ime ⇅	Email ⇅		Rolovi	Napravljen ⇅
1	admin	admin@mev.hr	Aktivan	ROLE_ADMIN	
2	user	user@student.mev.hr	Aktivan	ROLE_USER	
3	profesor	profesor@mev.hr	Aktivan	ROLE_PROFESOR	
4	ivor123	ivor.janecek@student.mev.hr	Aktivan	ROLE_USER	21/09/23 17:37

Slika 18. Popis korisnika aplikacije

(izvor: Autor)

Jedan od izbora je zdravlje servisa aplikacije, slika 19. Tu se nalazi par servisa te njihov status. Po statusima možemo provjeriti da li aplikacija radi kako bi trebala.

Service name	Status	Details
Db	UP	👁
DiskSpace	UP	👁
LivenessState	UP	
Ping	UP	
ReadinessState	UP	

Slika 19. Zdravlje servisa aplikacije

(izvor: Autor)

Jedan od izbora koji nam nudi administratorski izvornik je konfiguracija. Kao što možemo vidjeti sa slike 20 tu je moguće pretražiti sve konfiguracije koje su namještene i u programu te ih je moguće pretražiti i pomoću filtera.

Filter (by prefix)

spring.sed

Spring configuration

Prefix ^	Postavke
spring.security	filter <pre>{ "order": -100, "dispatcherTypes": ["ASYNC", "ERROR", "REQUEST"] }</pre>
	user <pre>{ "name": "user", "password": "*****", "roles": [] }</pre>

server.ports

Property	Value
local.server.port	8080

Slika 20. Konfiguracija aplikacije

(izvor: Autor)

Zadnji izbor je zapisnik. U zapisniku su i svi ostali zapisnici te ih ima cca 1300 te ih je moguće sve i jedan podesiti da se upale / ugase ili da samo određene stvari zapisuju. Slika 21 pokazuje pretraživanje po zapisniku aplikacije te neke od postavki zapisnika.

Logs

There are 1386 loggers.

Filter

javax

Name ^	Level ⇅
javax	TRACE DEBUG INFO WARN ERROR OFF
javax.activation	TRACE DEBUG INFO WARN ERROR OFF
javax.mail	TRACE DEBUG INFO WARN ERROR OFF
javax.management	TRACE DEBUG INFO WARN ERROR OFF
javax.management.remote	TRACE DEBUG INFO WARN ERROR OFF
javax.xml	TRACE DEBUG INFO WARN ERROR OFF
javax.xml.bind	TRACE DEBUG INFO WARN ERROR OFF

Slika 21. Zapisnik aplikacije

(izvor: Autor)

6. Baza podataka

Aplikacija koristi MySQL bazu podataka. Baza se sastoji se od sedam tablica i tablice su međusobno povezane relacijama radi lakšeg i bržeg pretraživanja te očuvanja integriteta baze podataka.

Table	Action	Rows	Type	Collation	Size	Overhead
1.user	Browse Structure Search Insert Empty Drop	3	MyISAM	latin2_croatian_ci	4.4 KiB	-
2.authority	Browse Structure Search Insert Empty Drop	3	MyISAM	latin2_croatian_ci	2.1 KiB	-
3.user_authority	Browse Structure Search Insert Empty Drop	3	MyISAM	latin2_croatian_ci	3.1 KiB	-
4.natjecaj	Browse Structure Search Insert Empty Drop	5	MyISAM	latin2_croatian_ci	2.5 KiB	-
5.prijava	Browse Structure Search Insert Empty Drop	5	MyISAM	latin2_croatian_ci	5.5 KiB	-
6.mobilnost	Browse Structure Search Insert Empty Drop	3	MyISAM	latin2_croatian_ci	4.3 KiB	-
7.zahtjev	Browse Structure Search Insert Empty Drop	22	MyISAM	latin2_croatian_ci	2.7 KiB	-
8.fakultet	Browse Structure Search Insert Empty Drop	10	MyISAM	latin2_croatian_ci	2.5 KiB	-
9.upload_file	Browse Structure Search Insert Empty Drop	20	MyISAM	latin2_croatian_ci	5.7 MiB	-
10.upload_file_admin	Browse Structure Search Insert Empty Drop	0	MyISAM	latin2_croatian_ci	1.0 KiB	-
10 tables	Sum	74	MyISAM	latin2_croatian_ci	5.7 MiB	0 B

Slika 22. Struktura baze podataka
(izvor: Autor)

Popis tablica:

- **user**, služi kao popis svih registriranih korisnika i u njoj se spremaju svi osobni podaci korisnika. Osim osobnih podataka tablica sadrži i druge bitne podatke poput podataka je li je korisnik aktiviran, je li su korisnikovi podatci promijenjeni, od koga te datum mijenjanja. Tablica korisnik je centralna tablica u bazi podataka jer se na nju veže većina drugih tablica.

- **authority**, tablica služi kao popis svih rolava koje mogu biti dodijeljene korisniku.

- **user_authority**, tablica služi kao poveznica između tablice user i authority. Tablica omogućuje da dodjeljujemo korisnicima određene uloge.

- **natječaji**, tablica natječaja sadrži podatke vezane za raspisane natječeaje.

- **prijava**, tablica koja sadrži sve zahtjeve koje su korisnici postavili za određeni natječaj. Sadrži: naziv zahtjeva, datum stvaranja i informaciju o stanju zahtjeva (na čekanju, odobren ili odbijen).

-***mobilnosti***, tablica mobilnost je povezana s tablicom natječaji tako da se zna za koji natječaj je pokrenuta određena mobilnost. Povezana i s tablicom prijava tako da se uzme id od prijave koja je odobrena te se zapiše u tablicu mobilnost.

-***zajtjevi***, tablica zahtjeva je povezana s tablicom natječaji te se koristi kao popis zahtjeva određenih dokumenata koje korisnik mora umetnuti uz prijavu.

-fakultet, tablica fakulteta sadrži sav sadržaj vezan za fakultete na koje korisnik želi ići na mobilnost.

-***upload_file***, tablica upload_file je tablica dokumenata i sadrži sve važne dokumente koje korisnik treba uploadati vezano za prijavu ili mobilnost.

-***upload_file_admin***, tablica upload_file_admin je tablica dokumenata koja sadrži sve važne dokumente koji su potrebni korisniku za određenu mobilnost. U toj tablici su dokumenti uneseni od strane administratora.

7. Testiranje rada aplikacije

Svrha testiranja aplikacije je potvrditi njezinu funkcionalnost. Kako bismo osigurali da program radi ispravno, svaki program mora biti testiran kako bi se postigao naš zamišljeni, programirani i željeni ishod.

Aplikacija je prošla ručno testiranje svakog modula pojedinačno. Prilikom testiranja izvodilo se testiranje funkcionalnosti, testiranje sigurnosti, testiranje baze i testiranje izgleda kao što će biti prikazano u ovom poglavlju.

Testiranje je započeto s registracijskom formom. Aplikacija zahtijeva da su svi podaci upisani. Kod unosa emaila, mogu se koristiti samo mail adrese koje su registrirane na veleučilištu. Točnije, adrese koje završavaju sa @mev.hr ili @student.mev.hr, ako mail ne sadrži “student“ dodaje mu se role profesor te ako sadrži onda mu se dodaje role student. Vršiti se i dodatna provjera prilikom unosa emaila koja onemogućuje korisniku da se registrira s adresom koja se već koristi. Prilikom unosa lozinke i potvrde nove lozinke, one se moraju podudarati. U slučaju da lozinke nisu jednake, korisniku se prikaže poruka o pogrešci.

Registracija

Lozinka i potvrda lozinke se ne podudaraju

Korisničko ime
ivor123

Email
ivor.janecek@student.mev.hr

Nova Lozinka
.....

Snaga lozinke:
■■■■■■■■■■

Potvrda nove lozinke
.....

Napravi novi račun.

Već imaš račun [Ulogiraj.se](#)

Slika 23. Netočna potvrda lozinke pri registraciji
(izvor: Autor)

Nakon uspješne registracije, korisniku se šalje email s linkom za potvrdu računa na unesenu adresu. Klikom na aktivacijski link poslan u mailu, račun se potvrđuje i spreman je za korištenje.

Aktivacija

Vaš račun je napravljen. Molimo vas [Ulogiraj se](#).

Slika 24. Aktivacija novog računa

(izvor: Autor)

Nakon aktivacije, korisnik se sa novo kreiranim računom može prijaviti. Korisnik unosi svoje pristupne podatke u prijavnu formu koja se sastoji od emaila i lozinke. Unosom pogrešnog maila ili lozinke na ekranu se prikaže poruka „Neuspješno logiranje ! <https://www.youtube.com/watch?v=gjXXPha1tGk>“

Poslije uspješne prijave, korisniku se prikaže početna s njegovim osobnim podacima. Na početnoj stranici je korištena ngIf funkcija koja provjerava da li je račun prijavljen ili nije te po tome odlučuje što će korisnik vidjeti na početnoj stranici.

Prilikom odabirom iz menija na Natječajе korisnici se susreću s tablicom natječaj. Pri učitavanju liste natječaja koristi se filter koji filtrira korisnicima natječaje ovisno o njihovom rolu.

```
@GetMapping("/natjecajs")
public ResponseEntity<List<NatjecajDTO>> getAllNatjecajs(
    @org.springframework.api.annotations.ParameterObject Pageable
    pageable,
    @RequestParam(required = false) String filter
) {
    if ("mobilnost-is-null".equals(filter)) {
```

```

        log.debug("REST request to get all Natjecajs where mobilnost
is null");

        return new
ResponseEntity<>(natjecajService.findAllWhereMobilnostIsNull(),
HttpStatus.OK);

    }

    log.debug("REST request to get a page of Natjecajs");

    Page<NatjecajDTO> page = natjecajService.findAll(pageable);
    List<NatjecajDTO> natjecajDTOS = page.getContent();

    if (SecurityUtils.hasCurrentUserAdminRole()) {

        return new ResponseEntity<>(natjecajDTOS, HttpStatus.OK);

    } else {

        natjecajDTOS =

            page

                .getContent()

                .stream()

                .filter(x ->
SecurityUtils.hasCurrentUserThisAuthority("ROLE_" + x.getKorisnik()))

                .collect(Collectors.toList());

        return new ResponseEntity<>(natjecajDTOS, HttpStatus.OK);

    }

}

```

Kod 4. Korištenje filtera natječaja u servisima

(izvor: Autor)

Iz priloženog koda 4. može se vidjeti da aplikacija prvo provjerava ako korisnik ima admin rolu s komandom `if (SecurityUtils.hasCurrentUserAdminRole())` i u tom slučaju ispiše sve natječaje. Ako korisnik nema rolu admin, onda aplikacija filtrira natječaje i ispiše mu samo natječaje koje bi smio vidjeti.

Administrator tu ima opcije pravljenje novog natječaja, uređivanje i brisanje postojećeg te opciju pogledaj koji i korisnik može koristiti koja odvede na detalje nekog određenog natječaja. Prilikom brisanja natječaja provjeravamo da li natječaj u sebi već ima prijave. Prilikom brisanja natječaja s prijavama u sebi narušili bi funkcionalnost stranice pa zbog toga aplikacija sadrži kod koji to provjerava. Kod 5. provjerava da li je polje prijave u objektu natječaj prazan te u tom slučaju daje dopuštenje korisniku da obriše prijavu.

```
@PreRemove

private void preventDeleteIfHasPrijave() {

    if (!prijavas.isEmpty()) {

        throw new NatjecajHasPrijaveException("Nije moguće
izbrisati natjecaj jer postoje prijave.");

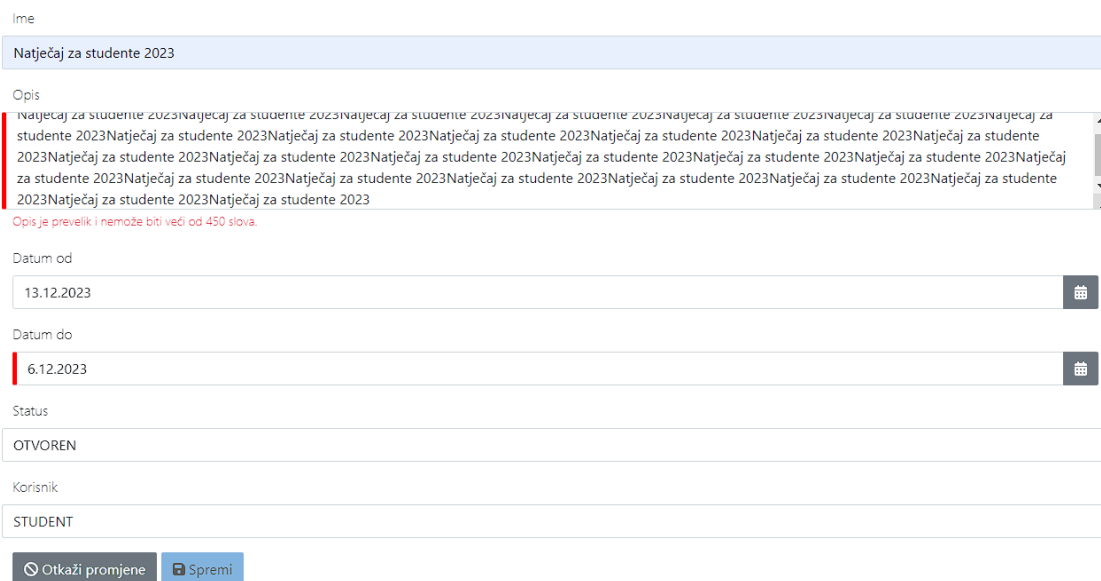
    }

}
```

Kod 5. Kod koji spriječi korisnika da obriše natječaj ako sadrži prijave

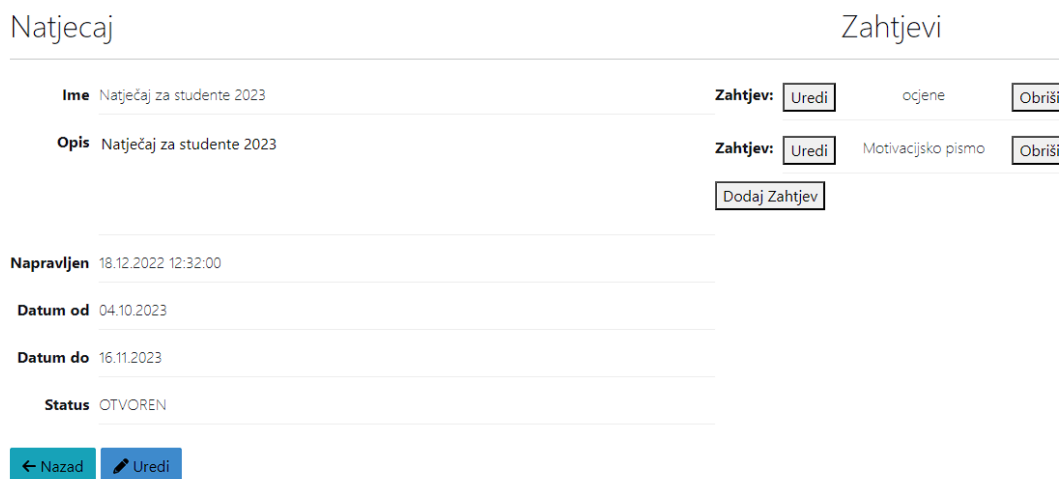
(izvor: Autor)

Prilikom pravljenja novog natječaja ili uređivanjem postojećeg, aplikacija provjerava ispravnost upisanih podataka. Za opis prilikom testiranja polje se čudno ponašalo na način da je tekst prešao preko drugih redaka u istom redu ili se slomio ali i na opis natječaja ispod pa je zato ograničen na 450 slova. Za polje datuma aplikacija provjerava da datum završetka mobilnosti ne može biti veći od datuma početka mobilnosti.



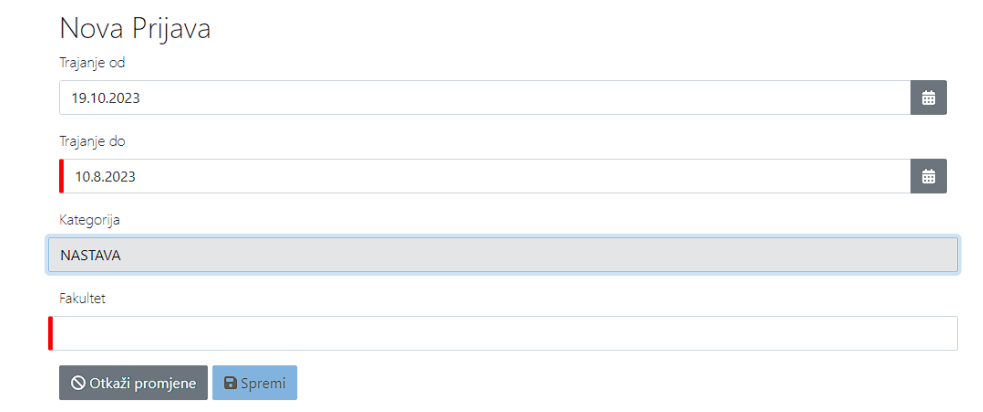
Slika 25. Uređivanje već postojećeg natječaja
(izvor: Autor)

Kao što možemo vidjeti, iz primjera sa slike natječaj se ne može spremiti ako sva polja nisu popunjena ili ako je polje tehnički neispravno. Odabirom opcije pogledaj, aplikacija nas proslijedi na detalje određenog natječaja gdje se nalaze zahtjevi kojima može upravljati samo administrator. Tu se također nalazi i gumb za prijavu na taj natječaj.



Slika 26. Prikaz detalja natječaja sa administratorske strane
(izvor: Autor)

Zahtjevi služe kao indikacija koje dokumente korisnik mora unijeti kako bi njegova prijava na natječaj bila potpuna.



Nova Prijava

Trajanje od
19.10.2023

Trajanje do
10.8.2023

Kategorija
NASTAVA

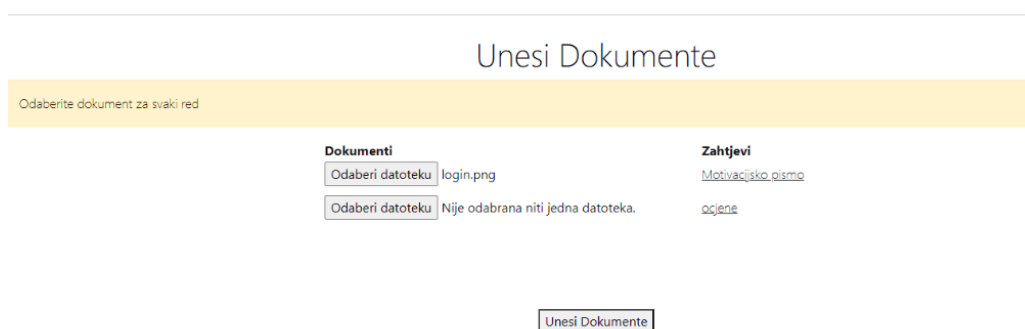
Fakultet

Otkaži promjene Spremi

Slika 27. Prijava za natječaj sa studentske strane
(izvor: Autor)

Prilikom klika na gumb prijavi se na natječaj, aplikacija proslijedi korisnika na formu koju mora popuniti da bi se prijavio na natječaj kao što je prikazano na slici 27. Forma sadrži četiri polja koja moraju biti popunjena te datum završetka mobilnosti mora biti veći od datuma početka mobilnosti da bi se korisnik mogao prijaviti.

Kada korisnik uspješno popuni prijavu, aplikacija će ga proslijediti na stranicu gdje mora unijeti dokumente za prijavu. Unos dokumenata aplikacija provjerava te se mogu unijeti svi tipovi dokumenata. Dokumenti ne smiju biti veći od 1MB radi ograničenog memorijskog prostora koji se koristi u testiranju aplikacije.



Unesi Dokumente

Odaberite dokument za svaki red

Dokumenti	Zahjevi
Odaberi datoteku login.png	Motivacijsko pismo
Odaberi datoteku Nije odabrana niti jedna datoteka.	ocjena

Unesi Dokumente

Slika 28. Unos dokumenata za prijavu
(izvor: Autor)

Ovisno o odabranom natječaju, aplikacija na izbor daje korisniku unos dokumenata ovisno o broju zahtjeva sa tog natječaja. U slučaju da u jednom ili više redova nije unesen dokument aplikacija prikazuje korisniku poruku o pogrešci.

Ako su svi dokumenti uneseni prema pravilima, aplikacija će proslijedi korisnika na listu njegovih prijava.

Prijava ↕	Napravljen ↕	Status Prijave ↕	Trajanje od ↕	Trajanje do ↕	Kategorija ↕	Korisnik ↕	Fakultet ↕	Natječaj ↕
Prijava za NATJEČAJ ZA PROFESORE 25.08.2023	25.08.2023 09:43:00	PRIHVACEN	19.10.2023	16.11.2023	STRUCNO_USAVRSAVANJE	bruno	Fakultet A	NATJEČAJ ZA PROFESORE 25.08.2023
Prijava za Novi natjecaj za Studente	31.07.2023 20:53:00	PRIHVACEN	01.08.2023	22.09.2023	NASTAVA	user	Fakultet A	Novi natjecaj za Studente

Slika 29. Lista prijava iz pogleda studentske strane

(izvor: Autor)

Do podataka o vlastitim prijavama, korisnik može doći i preko menija. Korisnik može mijenjati podatke u vlastitoj prijavi a pod to se misli uređivanje ili brisanje. Pri uređivanju prijave dobije se ista forma i moraju se poštivati ista pravila da bi se promjene mogle spremiti.

Prijava

Ime prijave Prijava za Natječaj za studente 2023

Napravljen 21.09.2023 19:03:00

Status NOVA_PRIJAVA

Trajanje od 19.10.2023

Trajanje do 25.10.2023

Kategorija NASTAVA

Korisnik user

Fakultet monitor Somalia

Natječaj Natječaj za studente 2023

[← Nazad](#) [Uredi](#)

Zahtjevi

Motivacijsko pismo **Otvori** proof.png [Uredi](#)

ocjene **Otvori** login.png [Uredi](#)

Dokumenti

Slika 30. Detalji prijave iz pogleda studentske strane

(izvor: Autor)

U slučaju da korisnik unese pogrešne dokumente, korisnik može urediti dokument i zamijeniti postojeći dokument s novo odabranim. Ako je korisnik izašao prije nego je

unio dokumente poslije popunjena prijave ispod naslova dokumenti će stajati opcija koja će proslijediti korisnika na link da ih ponovno unese, ta stranica je ista kao i na slici 28.

Administrator provjerava dokumente ručno klikom na link otvori. Administrator može odbiti prijavu. Ako je prijava odbijena, korisnik se može ponovo prijaviti na taj isti natječaj. Da bi administrator prihvatio prijavu svi podaci moraju biti ispravni te mora biti jednako dokumenata kao što je i zahtjeva. Promjenom statusa prijave, korisnik više nije u mogućnosti uređivati istu ili je obrisati. Nakon što administrator potvrdi prijavu aplikacija automatski popuni mobilnost sa informacijama iz povezanog natječaja i prijave.

Podaci o mobilnosti se ne mogu mijenjati, samo se može promijeniti status. Kao i kod natječaja i prijavi, mobilnost također ima filter gdje svaki korisnik može vidjeti svoje mobilnosti dok administrator može vidjeti sve.

Ime ↕	Status Mobilnosti ↕	Natječaj ↕	Prijava ↕	
Mobilnost za Novi natjecaj za Studente	ZATVORENA	Novi natjecaj za Studente	Prijava za Novi natjecaj za Studente	Pogledaj
Mobilnost za NATJEČAJ ZA PROFESORE 25.08.2023	OTVORENA	NATJEČAJ ZA PROFESORE 25.08.2023	Prijava za NATJEČAJ ZA PROFESORE 25.08.2023	Pogledaj

Slika 31. Lista mobilnosti

(izvor: Autor)

Odabirom određene mobilnosti, aplikacija prikaže detalje o odabranoj mobilnosti. Tako je prikazano u slici 32. Osim tih detalja imaju i dvije tablice koja jedna služi za dokumente od strane studenta ili profesora, a druga za dokumente od strane administratora.

Mobilnost

Mobilnost za Natječaj za studente 2023

Mobilnost Od 19.10.2023

Mobilnost Do 25.10.2023

Status OTVORENA

Natječaj [Natječaj za studente 2023](#)

Prijava [Prijava za Natječaj za studente 2023](#)

[← Nazad](#) [Zatvori Mobilnost](#) [Neispravna Mobilnost](#)

Dokumenti Korisnik

Otvori [zrcocf.png](#)

Otvori [sscode.promjene.txt](#)

Dokumenti Administrator

Otvori [Uredi erasmus useri.png](#) [Obrisi](#)

Otvori [Uredi aws key.txt](#) [Obrisi](#)

[Uploadaj Dokumente](#)

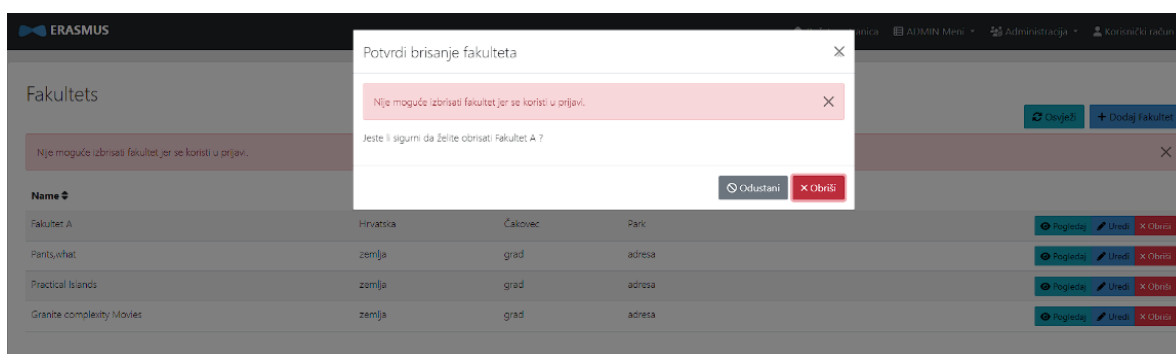
Slika 32. Detalji mobilnosti iz pogleda administratorske strane

(izvor: Autor)

Svi korisnici mogu otvarati ili preuzimati sve dokumente. Administrator može određivati status svake mobilnosti. U slučaju da administrator odluči da su priloženi dokumenti nevažeći ili ih nije dovoljno može promijeniti status na neispravna mobilnost.

Kada administrator odluči da su svi dokumenti popunjeni te da je sve što je potrebno da se zatvori mobilnost uneseno može kliknuti na gumb zatvori mobilnost.

Jedna od opcija koje ima samo administrator je lista fakulteta. Administrator može uređivati podatke o fakultetima. Fakultet isto kao i natječaj ne može biti obrisano ako se koristi u jednoj od prijava ili natječaja. Prilikom dodavanja novog fakulteta ili uređivanja postojećeg.



Slika 33. Brisanje fakulteta koji se već koristi
(izvor: Autor)

8. Sigurnost aplikacije

Sigurnost aplikacije oslanja se na Spring Security. Mogućnosti koje pruža Spring Security su prvenstveno autentifikacija te autorizacija no i dosta drugih sigurnosnih značajki koje mogu dodatno osigurati aplikaciju.

Kako bi se implementirao dodatni sloj zaštite, korisnička se lozinka pretvara pomoću mehanizma sha-256. Osim toga, aplikacija koristi JSON web tokene (JWT), koji su standardizirana metoda za siguran prijenos informacija između strana kao JSON objekt.

Ovi se tokeni mogu provjeriti i vjerovati im zahvaljujući svojim digitalnim potpisima. JWT-ovi se mogu potpisati korištenjem tajnog ključa s HMAC algoritmom ili para javnih/privatnih ključeva pomoću RSA ili ECDSA.



Slika 34. Tehnologije korištene za sigurnost aplikacije

(Izvor: <https://www.javatpoint.com/spring-security-tutorial>)

Iako JWT-ovi imaju mogućnost šifriranja za osiguranje tajnosti, prvenstveno se daje prioritet potpisanim tokenima.

Potpisani tokeni služe u svrhu provjere integriteta priloženih zahtjeva, dok šifrirani tokeni skrivaju te zahtjeve od drugih entiteta. Kada se za potpisivanje tokena koriste parovi javni/privatni ključevi, potpis dodatno jamči da je samo subjekt koji posjeduje privatni ključ odgovoran za potpisivanje.

HMAC igra ključnu ulogu u jamčenju integriteta podataka i autentifikacije. Oslanja se na kriptografska načela kako bi osigurala da podaci ostanu nepromijenjeni tijekom prijenosa. HMAC služi kao pouzdana metoda za autentifikaciju podataka i korisnika bez otkrivanja

tajnih ključeva. Slučajevi korištenja iz stvarnog svijeta pokazuju kako se HMAC koristi za osiguranje prijenosa podataka i API autentifikaciju.

```
public TokenProvider(Properties properties, SecurityMetersService
securityMetersService) {
    byte[] keyBytes;
    String secret =
properties.getSecurity().getAuthentication().getJwt().getBase64S
ecret();
    if (!ObjectUtils.isEmpty(secret)) {
        log.debug("Using a Base64-encoded JWT secret key");
        keyBytes = Decoders.BASE64.decode(secret);
    } else {
        log.warn(
            "Warning: the JWT key used is not Base64-encoded. "
        );
        secret =
properties.getSecurity().getAuthentication().getJwt().getSecret(
);
        keyBytes = secret.getBytes(StandardCharsets.UTF_8);
    }
    key = Keys.hmacShaKeyFor(keyBytes);
    jwtParser =
JwtParser.builder().setSigningKey(key).build();
    this.tokenValidityInMilliseconds = 1000 *
properties.getSecurity().getAuthentication().getJwt().getTokenVa
lidityInSeconds();
    this.tokenValidityInMillisecondsForRememberMe =
1000 *
properties.getSecurity().getAuthentication().getJwt().getTokenVa
lidityInSecondsForRememberMe();
    this.securityMetersService = securityMetersService;
}
```

Kod 6. Korištenje HMAC-a za kodiranje ključa

(izvor: Autor)

Priloženi kod 6. prvo provjerava je li dostavljen JWT tajni ključ kodiran Base64. Ako je tajna kodirana Base64 dostupna, dekodira je pomoću `Decoders.BASE64.decode(secret)` i pohranjuje dobiveni niz bajtova u `keyBytes`. Ovo je uobičajena praksa za rukovanje tajnim ključevima za JWT (JSON web tokene) kada su pohranjeni kao nizovi kodirani Base64.

Ako tajni ključ kodiran Base64 nije dostavljen ili je prazan, vraća se na drugi blok koda. U ovom slučaju, dohvaća JWT tajnu kao uobičajeni niz pomoću `properties.getSecurity().getAuthentication().getJwt().getSecret()`. Zatim pretvara ovaj niz u bajtove koristeći UTF-8 kodiranje i pohranjuje dobiveni niz bajtova u `keyBytes`.

Konačno, koristi `Keys.hmacShaKeyFor(keyBytes)` za stvaranje HMAC ključa iz `keyBytes`. Ovaj je ključ prikladan za upotrebu u procesima potpisivanja i verifikacije temeljenim na HMAC-u. Budući da je token potpisan, korisnik ga ne može mijenjati te je povećana sigurnost aplikacije.

Sigurnost je povećana i sa takozvanim Rolovima. Korisnici mogu imati jedno od dva moguća ovlaštenja za korisnika koja su `ROLE_USER` i `ROLE_PROFESOR`. U teoriji oba dva ovlaštenja su veoma slična. Administrator ima posebna ovlaštenja za `ROLE_ADMIN` te ima puno veću mogućnost mijenjanja različitih podataka ili čak podešavanja nekih od raznih opcija u izvođenju aplikacije.

```
public static boolean hasCurrentUserAnyOfAuthorities(String...
authorities) {
    Authentication authentication =
SecurityContextHolder.getContext().getAuthentication();
    return (
        authentication != null &&
getAuthorities(authentication).anyMatch(authority ->
Arrays.asList(authorities).contains(authority))
    );
}
```

Kod 7. Provjera rolova korisnika

(izvor: Autor)

Ovaj dio koda 7. iz web aplikacije dohvaća trenutni korisnikov Authentication objekt iz SecurityContextHolder-a.

Provjerava je li Authentication objekt null, a zatim poziva metodu getAuthorities za dobivanje toka dodijeljenih ovlaštenja za trenutnog korisnika.

Koristi se metodom anyMatch za provjeru podudara li se bilo koje od navedenih ovlaštenja (dostavljenih kao niz nizova) s bilo kojim od ovlaštenja dodijeljenih korisniku. Ako postoji podudaranje, vraća se istina, što pokazuje da korisnik ima barem jedno od navedenih ovlaštenja.

Dakle, metoda hasCurrentUserAnyOfAuthorities koristi se za provjeru ima li trenutni korisnik neku od navedenih uloga/ovlasti.

9. Zaključak

Erasmus web aplikacija jednostavna je za korištenje te je pristupačna za sve studente veleučilišta. Aplikacija omogućava studentima i nastavnicima da se prijave za razmjenu studenata i nastavnika unutar veleučilišta. Aplikacija sve podatke sprema u bazu podataka te korisničke šifre pretvara u random hash kod pomoću Bcrypt algoritma radi veće sigurnosti.

Cilj rada je ostvaren. Korisnicima web stranice omogućit će se objava natječaja za mobilnost, a korisnicima mobilnosti omogućit će se trajna pohrana podataka vezana za mobilnost.

Testiranje aplikacije pokazalo je da je aplikacija sigurna, funkcionalna, stabilna, pouzdana, optimizirana i spremna za rad. Detaljno testiranje aplikacije rezultiralo je ispravljanjem potencijalnih pogrešaka u radu aplikacije te da je u mogućnosti podržati unos velike količine podataka bez grešaka pri unosu.

Sigurnost podataka u aplikaciji je ostvarena na visokoj razini. Prvenstveno zbog korištenja Spring Security-a. Spring Security nam omogućuje korištenje rolova, kriptiranje šifre korisnika, dekriptiranje šifre korisnika. Spring Security je također poznat po podršci za autorizaciju i autentifikaciju te za protekciju protiv uobičajenih grešaka sistema.

Budući razvoj aplikacije započinje s poboljšanjem korisničkog profila. Tu će se dodati dodatne informacije koje su bitne za svakog korisnika kao što su: datum rođenja, spol, i tako dalje. Nakon toga, umjesto obične login forme će se uvesti OAuth 2.0 autentifikacija koja bi poboljšala sigurnost i smanjila rizik od neovlaštenog pristupa web aplikaciji.

Sigurnost aplikacije moguće je poboljšati implementiranjem ograničene brzine za zahtijeva tokena kako bi se spriječila zlouporaba i postavio nadzor i upozoravanje za sumnjive ili neobične aktivnosti. Izgled aplikacije moguće je poboljšati tako da bi se unijelo više slika te dodale ikone za svaki izbornik u meniju.

Bilo bi dobro još i implementirati da aplikacija se može povezati sa stranicom fakulteta da unese sve podatke studenata i profesora unutar tog fakulteta.

10. Literatura

- [1] Wayback Machine. Erasmus: [Online]. 2023. Dostupno na:
https://web.archive.org/web/20130404063516/http://ec.europa.eu/education/erasmus/history_en.html (30.9.2023.)
- [2] Mobilnost. Erasmus+: [Online]. 2023. Dostupno na:
<https://www.mobilnost.hr/hr/sadrzaj/programi/erasmus/> (13.9.2023)
- [3] Međimursko veleučilište u Čakovcu. Erasmus: [Online]. 2023. Dostupno na:
<https://www.mev.hr/erasmus/> (13.9.2023.)
- [4] Washington. HTML history. [Online] 2023. Dostupno na:
https://www.washington.edu/accesscomputing/webd2/student/unit1/module3/html_history.html (8.10.2023)
- [5] Hostinger. What is CSS?: [Online]. 2023. Dostupno na:
<https://www.hostinger.com/tutorials/what-is-css> (14.9.2023.)
- [6] Bybase. Postgres vs. MySQL: a Complete Compararison in 2023: [Online]. 2023.
Dostupno na: <https://www.bytebase.com/blog/postgres-vs-mysql/>(14.9.2023.)
- [7] JetBrains. IntelliJ IDEA overview: [Online]. 2023. Dostupno na:
<https://www.jetbrains.com/help/idea/discover-intellij-idea.html#IntelliJ-IDEA-supported-languages> (14.9.2023.)
- [8] Apache Maven Project. Maven: [Online]. 2023. Dostupno na:
<https://maven.apache.org/what-is-maven.html> (30.9.2023.)
- [9] Heroku. What is Heroku?: [Online]. 2023. Dostupno na:
<https://www.heroku.com/about#:~:text=Heroku%20is%20a%20container%2Dbased,getting%20their%20apps%20to%20market.> (30.9.2023.)
- [10] Visual Studio Code. Getting Started: [Online]. 2023. Dostupno na:
<https://code.visualstudio.com/docs> (14.9.2023.)
- [11] TechTarget. GitHub: [Online]. 2023. Dostupno na:
<https://www.techtarget.com/searchitoperations/definition/GitHub> (30.9.2023.)

11. Popis slika

Slika 1. Erasmus+ program	6
Slika 2. HTML.....	8
Slika 3. Spring boot logo	10
Slika 4. MySQL logo.....	11
Slika 5. IntelliJ IDEA logo	13
Slika 6. Konfiguracija verzija unutar POM datoteke	14
Slika 7. Heroku logo.....	15
Slika 8. VSCode logo	16
Slika 9. Implementacija servisa web aplikacije Erasmus u Github repozitoriju	17
Slika 10. Hijerarhijska struktura aplikacije	19
Slika 11. Početna stranica web aplikacije erasmus.....	20
Slika 12. Stranica Prijave	20
Slika 13. Stranica registracije	21
Slika 14. Lista natječaja.....	22
Slika 15. Detalji prijave.....	23
Slika 16. Lista mobilnosti.....	24
Slika 17. Lista Fakulteta	24
Slika 18. Popis korisnika aplikacije.....	25
Slika 19. Zdravlje servisa aplikacije	25
Slika 20. Konfiguracija aplikacije	26
Slika 21. Zapisnik aplikacije	26
Slika 22. Struktura baze podataka	27
Slika 23. Netočna potvrda lozinke pri registraciji	29
Slika 24. Aktivacija novog računa.....	30
Slika 25. Uređivanje već postojećeg natječaja	33
Slika 26. Prikaz detalja natječaja sa administratorske strane.	33
Slika 27. Prijava za natječaj sa studentske strane.....	34
Slika 28. Unos dokumenata za prijavu	34
Slika 29. Lista prijava iz pogleda studentske strane	35
Slika 30. Detalji prijave iz pogleda studentske strane.....	35
Slika 31. Lista mobilnosti	36
Međimursko veleučilište u Čakovcu	42

Slika 32. Detalji mobilnosti iz pogleda administratorske strane.	36
Slika 33. Brisanje fakulteta koji se već koristi.	37
Slika 34. Tehnologije korištene za sigurnost aplikacije	38

12. Popis koda

Kod 1. HTML kod za prikaz početne stranice.....	9
Kod 2. Primjer korištenja CSS-a za forme, error-e i readonly..	10
Kod 3. MySQL kod koji dohvaća sve natječeaje koji imaju status otvoren..	10
Kod 4. Korištenje filtera natječaja u servisima	30
Kod 5. Kod koji spriječi korisnika da obriše natječaj ako sadrži prijave..	31
Kod 6. Korištenje HMAC-a za kodiranje ključa..	39
Kod 7. Provjera rolova korisnika.....	41